548 Topic list, Course structure, etc.

Me: Mark Oskin, oskin@cs.washington.edu, 206-293-9456 (cell, txt is preferred).

Daily reading: reading will be given for each class period. Expectation is everyone will read the material and answer two questions: (1) summarize the key ideas; (2) formulate 3-4 discussion questions.

Daily presentation: everyone will rotate (we'll assign an order today) taking turns presenting a short 5-10m introduction to the reading. Expectation is 3-4 slides.

Class structure: This class will rarely have slides or pre-canned material. Instead the expectation is you have read the material and we will spend the class period discussing it and digging deeper into the ideas. I may project stuff on a screen but it is more for taking shared notes than presenting anything.

Format: Class is in person, *unless* you are quarantining at home and notify me in advance, in which case class is still in person but a zoom window will be open at http://washington.zoom.us/my/markoskin There will be times that I am traveling and the class will be all zoom. Watch your email for notification.

Project: Choose a topic you are interested in. Suggested topics are later in this document. Produce a small research paper on that topic and present it in class (10-15 minute presentation, 7-10 page paper).

Grading: 25% participation (including intro talks); 75% project (including presentation). No exam.

Topics this class will cover. Suggestions welcome:

- RISC v CISC
- Modern CPUs
    - branch prediction
    - caching
    - prefetching
    - out-of-order execution
    - vector/SIMD
    - scheduling
    - threading (MTA)
    - (value prediction?)
- Multicores
    - Consistency
    - Cache coherence
    - Directory coherence
    - Software approaches to coherence (page, language, …)
    - NOCs
- Many cores
    - Hardware accelerators
- GPUs
    - Hardware design
    - Software model & tools
- Datacenters
    - Rack & Wearhouse scale

    - Virtual Machines
- Quantum computers/architecture
    - Survey of current systems

Project ideas

- Using BlackParrot, an RTL level RISC-V multicore that can boot Linux, implement some feature from the research literature. Features may be a more sophisticated branch predictor, a cache prefetcher, a SIMD extension unit, etc.
- Integrate BlackParrot with X, where X is some interesting RTL project. Study the features of X and the necessary changes for efficient integration into the system.
- Choose a paper from the last 10 years of ISCA/MICRO/ASPLOS and attempt to reproduce their results. Particularly interesting would be to implement a model for their idea on your own and attempt to reproduce the author(s)' findings.
- Perform a comprehensive literature review of a particular topic. This project involves little coding but extensive reading. Literature review projects are expected to survey at least 50 papers and produce a new way to think about the research area.
- Design and evaluate an accelerator for a new application area.
- Write a paper that argues convincingly what computers will be like in 2035. This is harder than it sounds. I expect at least 25 citations and 10 graphs.

Project Milestones:

End of week 2: Pick a project and have our OK on it. This should be a short paragraph describing the project and how it will be conducted.

End of week 4: Status update. Provide a detailed, but short update on where you are with the project.

End of week 6: Draft of project write up.

Week 7-8: project presentation

End of week 9: Finished project write up.