# 548

# Lecture 1

# Overview

- No book, just topical literature
- Project: Choose your own adventure
- Homework: slide prep on readings
- Grading: 50% project, 20% homework / participation, 30% Final exam (essay style)

# Homework

- For each class 1-3 papers are assigned
- Read each, and produce 4 slides on each
  - 3 slides: summary
  - 1 slide: questions
- Email them to me before class
- Depending on class size, 2-3 times a quarter you'll present your slides

# Final exam

- Essay style
- 3-4 questions
- Not graded on much grammar or speling.

# Project timeline:

- Pick a partner and a project (~now)
- Talk to me about it (~this week or early/late next)
- Get started (next week)
- Tell me where you are at (~October 25th)
- Hand in a draft of report ( < Thanksgiving)
- Present findings to class (near the end of the quarter)

# Project scope

- In a nutshell: Goal is a paper that could be submitted to a conference
  - i.e., ~ 10 pages small font, well written and argued, experimental or observational or analytical data that justifies the position
  - Talk ~ 20 minutes + 5-10 for questions

# Sample projects

- Experimental work:
  - Trace a complete OS/app stack and compute the parallelization limit
  - Hack VMware from a guest by exploiting a processor bug
  - "Learn" something about another process by observing information leakage in a multicore
  - How much data redundancy is in a cache? How do we exploit it?
- Design work:
  - a < 200xtr microprocessor
  - a system capable of running "forever" from scavenged energy
  - What would a python or ruby processor look like?
- Application work:
  - Parallelize something hard
  - Implement various algorithms on XMT
- Literature work:
  - History of instruction sets
  - Survey of the last 10 years of commercial microprocessors

# Lecture Cancelations

- Oct 5th (grant review in BWI)
- Nov 28th (Thanksgiving)
- TBD: Dec 5th & 7th (MICRO in Brazil)

# Office Hours

- Open-door & by appointment (in other words, whenever you need it).
- My cell is 206-293-9456
- I'm thinking of hanging out at Big Time Monday after class. If you want to join me, I'll buy the first couple of weeks
- But not next Monday (Travel to BWI)

# As you are about discover, class does not move unless you participate...

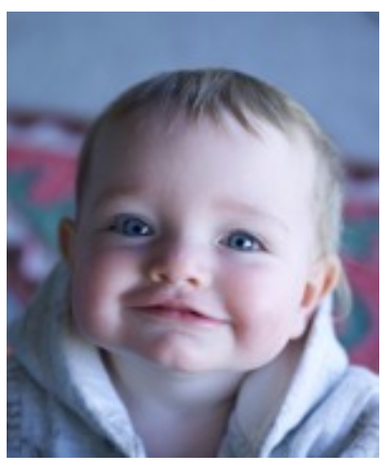
- Start lecture with one of you presenting the current reading
- We then have an interactive discussion about it
- Don't expect canned slides, and don't expect the notes we make in class to be intelligible if you were not there.
- This will all become clear shortly...

# Who are you?

- 4 y1, 4 y2, 1 y3
- 2 systems architecture, 1 ML, 2 networks/systems, 1 OS/systems, 1 theory, 1 QC
- Mostly from the east coast
-

# Who am I?

- Started hacking code ~ 82, burnt my first computer to the ground (crossed wires) ~ 88.
- I can't spelll. I don't know good english. I try and hide these facts by writing illegibly.
- Joined the faculty in 2001, went on leave 2008, just now returning.
- Founded Corensic in 2008; tools for writing parallel programs
- Worked on a variety of topics, including: intelligent memory, quantum microarchitectures, dataflow, graph machines; and dabbled in many more: determinism, FPGA computing, etc..
- Hobbies: motorcycles, woodworking, construction, good food, good beer, photography, fishing, sailing, hiking, radio (KE7SCH)
- BUT, mostly I spend my free time on this:



# What do you want from this class?

- Checking off systems requirement for quals
- better understanding of hardware
- better ideas of current challenges and potential solutions
- problems in architecture that could be "solved" by quantum techniques
- understand the hardware so can write more efficient code
- understand why the HW/SW interface is the way it is
- understand the co-development of systems and hardware
- better grasp of memory consistency
- where HW is going now

# What is computer architecture?

- designing buildings with CAD - NOT!
- physical layout of a computer (micro-architecture)
    - how components fit together to execute programs
- how software goals govern microarchitecture
- the HW/SW interface

# What to computer architects worry about?

- power/energy  (watt)
- performance (MIPS)
- MIPS/watt
- cost
- programming model
- technology

# Some topics...

- Technology (From relays to tubes to xtrs to chips)
- How architectures work (design, measurement, etc)
- Architecture (ISA) vs. Microarchitecture
- Instruction Level Parallelism (ILP)
- Memory
- Parallel processors
- Virtualization
- Dark Silicon
- Dark Corners...

# What are the major architectural innovations?

- ISAs, RISC, CISC
- Superscalar
- Multicore (SMP), SMT
- Pipelining
- Caching
- Out-of-order execution
- Speculation
- Security, etc

# What are the current major challenges?

# How do architects do what they do?

# What should HW provide?

# Why HW?