

# Who are you?

- 1 UG 1 UN 19 G
- 5-6 figure out while here
- 2 Architecture
- 3 Graphics, 3 AI
- 2 Databases
- 1 Ubiquitous computing
- 2 Systems
- 2 Theory

# A little about me...

- I am not a processor architect
- I am wondering lately if I am an architect
- I used to worry about the memory system
- Now I worry about:
  - simulation
  - statistics
  - quantum computation
  - models of computation
  - limits of computation
  - processors

# What do you want from this class?

- What matters in the hype
- Compiler interactions
  - VLIW, trace scheduling ILP
    - (Intel: EPIC)
- Multiprocessors SMT vs. CMP
- Heterogeneous processors
  - Network processors
- Pipelining, Advanced Processors

# What is computer architecture?

- Just hardware
- Interacting components
- Division of jobs among hardware / software
- Organization of components
- Instructions
- Design

# What is not CA?

- Possibly:
  - (Complexity theory)
    - Algorithms
  - High level programming language
  - Circuits
  - Software Engineering
  - AI
  - Learning chips
- Actually Architecture is a little of all of these...

# What are the major innovations/principles in computer architecture?

- Protected memory / Virtual Memory
- Pipelining
- Caches
- Compilers
- Von-Neumann Machines
- Circuit optimizations (P/G)
- Parallel Processing
- Instruction Level Parallelism
- Locality
- Bus
- Hardware Emulation
- Branch Prediction
- Speculation

# What do computer architects do?

- Make computers faster
- Task Scheduling
- More Robust
  - Fault tolerant
- Distributed
- Better OS interfaces
- Lower Power
- More Secure
- Better compiler interface

# How do CA's do their work?

- Wrote simulation
- Hack the compiler
- Trading off different ideas
- Run some traces
- Write/Read a lot of papers
  - Generate graphs.. Lots of ISCA like graphs
- Teaching
- Design



# Why is this bad?

- No reality check
  - Does it matter?
  - Is it accurate?
- Hard to debug
  - Hard to say its objective
- No standard framework