CIS 501
Introduction to Computer Architecture

Class Project

Mini-Research Course Project

- What?
  - Investigate a research idea covered in class, in paper, or in head
- Who?
  - You and one or two other people
- When?
  - Proposal: Friday, Nov 18 (submit PDF via blackboard)
  - Final report: Friday, Dec 9 (last day of classes)
- Tools?
  - Simplescalar and Simics
- How much?
  - 20% of your final grade

Proposal and Final Report

- Proposal (< 500 words)
  - Names of group members
  - **Brief** description of idea
  - What you plan to measure and how
- Final report (~4000 words)
  - Brief description of idea
  - Brief description of your implementation
  - Description of your experimental configuration
  - Presentation **and analysis** of results
  - Like a mini conference paper (you've read some)

Simulation Tools

- SimpleScalar
  - User-level programs
  - Functional simulation: sim-func, sim-cache, sim-DLX
    - Simulates insn-by-insn
    - Fast, but impossible to evaluate some ideas
  - Timing simulation: **sim-R10K**
    - Simulates cycle-by-cycle
    - Accurate, only way to measure some things, but slow
- Simics
  - Full-system simulation (program + OS)
  - Can run any program
  - Few timing

Finding Ideas

- Research ideas described in class
  - E.g., runahead execution, dynamic cache resizing, etc.
- Ideas found in recent research papers
  - ACM digital library: http://www.acm.org/dl
  - Recent MICRO conferences: http://www.microarch.org/
  - Others: http://www.cs.wisc.edu/~arch/www/conferences.html
- At most two groups may work on same idea
  - First-come, first-served
  - E-mail us with a general topic as soon as you know it

Ideas: Branch Prediction Algorithms

- Implement new branch predictor in bpred.c
  - Investigate effects of table size, history length, etc.
  - Compare with other predictors
  - Examples:
    - Agree [Sprangle+, ISCA’97]
    - YAGS [Eden+, MICRO’98]
    - Perceptron [Jimenez, HPCA’01]
  - Interesting: predictors for more than one branch at a time
  - Can be done with functional simulation

Ideas: Cache Design

- Implement new cache organization in cache.c
  - Examples
    - Multi-lateral cache [Rivers+, ICPP’96]
    - Dynamically resizing cache [Albonesi, ISCA’98, Yang+, HPCA’02]
    - Cache with dynamically varying block sizes [Johnson+, MICRO’97]
    - Trace cache [Rotenberg+, MICRO’95]
    - Distance associative cache [Chishti+, MICRO’03]
    - Fully-associative software managed L2 [Hallnor+, ISCA’00]
    - Way prediction [Powell, MICRO’01]
  - Measure effect of capacity, associativity, block size
  - Functional simulation: measure $\%_{miss}$, calculate t_{avg}
  - Timing simulation: better measurements
    - Can probably plug cache.c right in

Ideas: Prefetching

- Implement prefetching algorithm in cache.c (or outside)
  - Instructions or data
  - Examples
    - Call-graph prefetching for insns [Annavaram+, HPCA’01]
    - Dependence-based prefetching for pointers [Roth+, ASPLOS’98]
    - Context-based prefetching for pointers [Cooksey+, ASPLOS’02]
    - Jump-pointer prefetching [Roth+, ISCA’99]
    - Dead-block prefetching [Lai+, ISCA’01]
    - Stream-buffers [Jouppi]
    - In-memory prefetching [Solihin, ISCA’02]
  - Functional simulation: study prefetch coverage and accuracy
  - Timing simulation: study prefetch timing
    - Not necessary if coverage/accuracy work thorough enough

Ideas: ILP Limit/Critical Path Study

- Study the effects of real constraints on potential of ILP
  - Write your own functional model or hack sim-R10K.c a little
  - Examples
    - Critical path modeling [Fields+, ISCA'01]
    - Limits of ILP [Wall, ASPLOS'91, Lam+, ISCA'02]

Ideas: Pipeline Techniques

- Implement something new in sim-R10K.c
  - Not for the feint of heart (or hacking novices)
  - Examples
    - Clustering/steering [Farkas+, MICRO'97, Baniasadi+, MICRO'00]
    - Speculation gating [Manne+, ISCA'98]
    - Runahead execution [Dundas]

Ideas: Value Prediction

- Value prediction: rather than wait for load values, predict
  - Still have to execute to verify prediction
  - Functional simulation: prediction accuracy
    - How many values can you guess right?
  - Timing simulation: speculation effectiveness
    - How much gain/loss correct/wrong speculation?
  - Examples
    - Load value locality [Lipasti+, ASPLOS'96]
    - Predictability of data values [Sazeides+, MICRO'97]

Ideas: Instruction Reuse

- Instruction reuse: opposite of value predictions
  - Remember prior computations and reuse, don't repeat
  - Like common sub-expression elimination in hardware
  - Functional simulation: reusability and repitition
  - Timing simulation: effectiveness of reuse (easier than VP)
  - Examples
    - Dynamic instruction reuse [Sodani+, ISCA'97]
    - Analysis of instruction repitition [Sodani+, ASPLOS'98]
    - Register integration [Petric+, MICRO'02]

Other Ideas

- Ask me if interested
  - Speculative scheduling
  - Power modeling
  - Multiprocessing
  - Multithreading
  - Buses and memory hierarchy
- Or if you have your own idea…