Final Exam Review

1 Time and Location

The final exam will be given from 10:30a.m.-12:30p.m., on Thursday December 16th, in the normal meeting place, CHE 2108.

2 Format

The final exam will have an identical format to the midterm. See Handout #13 for information about exam format.

3 Scope

The final exam is comprehensive, and covers all material from the first day of class up and including the last lecture, given on December 9th. Essentially, this includes the introductory material in Chapter 1 of H&P, instruction set architectures, basic pipelining, instruction-level parallelism, memory systems, and multiprocessors. However, an emphasis will be placed on instruction-level parallelism, memory systems, and multiprocessors.

4 Course Content

Since the final exam is comprehensive, you should refer to Handout #13 for an outline of the course content relevant to the first half of the semester. Below, you will find a fairly comprehensive outline of the topics covered since the midterm. Disclaimer: this is not meant to be an absolutely water-tight complete list of topics. In other words, if there is a topic not present in this list, it may still show up on the final exam. However, it is a pretty good first-cut at what we have covered.

IV. Instruction-Level Parallelism
   A. ILP basics
      1. Dependences
         a. Data
         b. Name
         c. Control
      2. Code scheduling
      3. Memory disambiguation
4. Register renaming
5. Loop unrolling
B. Multi-Issue
1. VLIW
   a. Static Issue
   b. Two-dimensional code scheduling
   c. Code expansion, extra NOPs
   d. Software pipelining
2. In-Order Superscalar
   a. Dynamic issue
   b. Hazards (more structural hazards possible)
   c. Issue rules
C. Dynamic instruction scheduling
1. General concepts
   a. Out-of-order execution (/issue)
   b. Dynamic loop unrolling
   c. Instruction window size
2. Scoreboarding
   a. Busy bits
   b. Scoreboard structure
   c. Checking for structural hazards
   d. Checking for WAW hazards
   e. Checking for WAR hazards
   f. Limitations
3. Tomasulo’s algorithm
   a. Ready bits
   b. Reservation stations
   c. Register renaming (register dataflow)
   d. Result bus
   e. Conflict queue (memory dataflow)
4. Code scheduling examples using scoreboarding or Tomasulo
5. Modern superscalar case study
   a. Register renaming logic
   b. Wakeup and select
   c. Reorder buffer
D. Compiler-based instruction scheduling
1. Trace Scheduling
   a. Trace selection
   b. Instruction scheduling
   c. Compensation code
2. Software speculation techniques
   a. Memory dependence speculation
   b. Ignore terminating exceptions
   c. Poison bits
E. Dynamic branch prediction
1. Branch history table
2. Multi-bit predictors
3. Correlating predictors
4. Branch target buffers

V. Memory Systems
A. Caches
1. Principle of locality
2. Cache organization
   a. Direct-mapped
   b. Set associative
   c. Fully associative
   d. Addressing the cache
3. Cache Management
   a. Replacement policies
   b. Write-hit policies
   c. Write-miss policies
   d. Instructions vs. data
4. Cache performance
   a. Average memory access time
   b. Impact on CPU time
5. 3 C’s
   a. Compulsory
   b. Capacity
   c. Conflict
6. Design tradeoffs / performance optimizations
   a. Increase block size
   b. Increase associativity
   c. Victim cache
   d. Prefetching (hardware and software)
   e. Multi-level caches
B. Naming and Protection
1. How programs use names
2. Relocation: base-register addressing
3. Adding protection: base-length register addressing
4. Adding sharing: segmented addressing
   a. Segment tables
   b. Translation lookaside buffers
   c. Privilege bits
C. Resource management
1. Virtual memory
2. Fragmentation problem
3. Paged addressing
   a. Page tables
   b. Translation lookaside buffers
4. Virtual memory and caching
   a. Synonym problem
   b. Homonym problem
   c. Solutions to these problems
D. Main memory
1. Memory sub-system organization
2. Building wider banks
3. Interleaved memory
   a. Low-order interleaving
   b. Increasing number of banks
   c. Prime number of banks
4. DRAM architecture
   a. Row and column access
   b. multiple internal banks
c. CDRAMs
d. Cache-line interleaving

VI. Multiprocessors
A. Parallel programming models
1. Model examples
   a. Shared memory
   b. Message passing
   c. Others
2. Explicit parallelism
   a. Partitioning
   b. Communication
   c. Synchronization
B. Machine organization
   1. Symmetric multiprocessor (SMP)
   2. Cache-coherence problem
   3. Memory consistency problem
   4. Bus-based cache-coherence protocols
      a. 3-state protocol
      b. MESI protocol
   5. Distributed shared memory

5 Practice Problems

Again, like the midterm, the best way to study for the final exam is to do lots of practice problems. Here are a list of H&P problems that I found to be good candidates (some of them overlap with problems given in Handout #13).

- Chapter 3: 3.1, 3.2, 3.3, 3.5, 3.6, 3.7, 3.9, 3.10, 3.14, 3.15
- Chapter 4: 4.2, 4.3, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.23
- Chapter 5: 5.1, 5.3, 5.4, 5.6, 5.8, 5.17, 5.18
- Chapter 6: 6.4