Midterm Review

1 Time and Location

The midterm will be given during normal class hours, 2:00p.m.-3:15p.m., on Thursday October 14th, in the normal meeting place, CHE 2108.

2 Format

The midterm will be open-book, open-notes, closed everything else (you can bring any material that you have accumulated in class, but nothing else). Expect problems that will require you to analyze designs, compute performance, analyze the execution of code, etc. It is recommended that you bring a calculator.

3 Scope

The midterm will cover all material from the first day of class up and including lecture 13, given on October 12th. Essentially, this includes the introductory material in Chapter 1 of H&P, instruction set architectures, basic pipelining, introduction to ILP, multiple issue machines, Scoreboarding, and Tomasulo’s algorithm.

4 Course Content

Here’s a fairly comprehensive outline of the topics covered thus far. 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 midterm. However, it is a pretty good first-cut at what we have covered.

I. Preliminaries
   A. Architecture basics
      1. Technology
      2. Applications
      3. Cost
      4. Interface design
      5. Performance evaluation
   B. Performance basics
1. Amdahl’s law
2. Performance equation
   \[ T = I \times CPI \times tcycle \]

II. Instruction Set Architectures
   A. Machine state
      1. Memory organization
      2. Register organization
      3. Data types
      4. Interrupts and events
   B. Register organizations
      1. Accumulator
      2. Index Register
      3. General Purpose Registers
      4. Load-store
      5. Stack
   C. Instruction types
      1. Operations
      2. Data movement
      3. Control flow
   D. Data types
   E. Addressing Modes
      1. Big vs. little endian
      2. Register
      3. Immediate
      4. Direct
      5. Register indirect
      6. Displacement
      7. Indexed
   F. Instruction Encoding
      1. Fixed instruction formats
      2. Variable-length instructions
      3. Compromise: a few good formats
   G. Control
      1. Unconditional jumps
      2. Conditional jumps (branches)
      3. Conditions (condition codes, flags, registers)
      4. Support for procedures
      5. Support for exceptions

III. Pipelining
   A. Implementing an ISA
      1. Computer architecture building blocks
      2. Pipeline implementation of MIPs
         a. Stages (F,R,E,M,W)
         b. Pipeline registers
         c. Computing pipeline performance
   B. Structural hazards
   C. Data hazards
      1. Types (RAW,WAW,WAR)
      2. Pipeline stall
      3. Bypass (forwarding)
D. Control hazards
   1. Pipeline stall
   2. Early branch test and target calculation
   3. Static prediction
   4. Delayed branch
E. Exceptions
   1. Types (synchronous, asynchronous)
   2. Restartability
   3. Precise semantics
   4. Problems with synchronous exceptions
      a. Multiple exceptions
      b. Out-of-order exceptions
   5. Single commit point
F. Multi-cycle operations
   1. Bandwidth, latency
   2. Impact on pipeline
      a. New structural hazards (competing for writeback port)
      b. More forwarding paths
      c. New data hazards that must stall
      d. Complications for precise interrupts

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 scoreboard or Tomasulo

5 Practice Problems

Given the problem-solving nature of the midterm, the best way to study for the exam is to do **LOTS** of practice problems. Here are a list of problems from H&P that I found to be good candidates. (Some of these are from your homeworks, but those are good problems as well).

- Chapter 1: 1.2, 1.3, 1.4, 1.6, 1.7, 1.16
- Chapter 2: 2.1, 2.4, 2.5, 2.6, 2.8, 2.9, 2.12
- Appendix A: A.1, A.2, A.3, A.5, A.6, A.8
- Chapter 3: 3.1, 3.2, 3.3, 3.5, 3.6, 3.7
- Chapter 4: 4.2, 4.3, 4.5, 4.6, 4.7, 4.8