Final Exam Study Guide

The final is comprehensive, but most coverage is on material since midterm: (we will provide copies of the green card again – there could be MIPS programming questions; no calculators allowed)

- Style is similar to Midterm: some shorter questions, some longer ones
- Reading is on the lectures page.
- Study the slides, review the reading, exercises in lecture, HW4.

Topics covered since the midterm:

Memory hierarchy – how a logical address results in a physical memory access
    Caching
    VM/Paging
Performance
Interrupts/Exceptions
I/O
Parallelism (basic concepts)

Here is a list of a few questions/topics that may help in your studying.
*__Note this is not meant to be a comprehensive list__ of what you will be asked on the exam, but should help you identify some areas you need to study further.

Design of Pipelined MIPS:

- Know all components and their operation
- Know flow of logic - which components are active when implementing a given operation
- Be able to specify control signals needed to accomplish specific instructions
- Be able to compare with 1-cycle, multi-cycle
- Know the issues in pipelined performance:
  - Why do we want to move branch decision logic earlier?
  - Give examples of problems from instructions being started before preceding instructions are complete

Caching

- Know the various forms of caching: direct mapped, fully associative, k-way set associative
  - How they work (be able to find or place items in the cache
  - Policies: write-through, write back, write around, allocate on write, LRU, etc.
- Describe importance of blocks, associativity, size, etc. on performance
- Know terms: index, tag, valid bit, dirty bit, etc.
- Explain what the “three Cs” are

Virtual Memory & TLBs

- Explain why RAM is considered a “fully associative cache for the disk”
  - (Full associativity in an L1 cache requires a “parallel compare” of tags to find a line …don’t pages also require it to find the page????) (in other words, how does VM work?)
- Pages
  - For 16K pages, and 32 bit addresses, how large is the virtual page number?
  - Give advantages/disadvantages of large vs. small page sizes
- TLB
  - Why do we bother with a TLB?
  - What stuff goes into a TLB entry?
  - How does the TLB work?

Skills

- In a pipelined datapath, show which wires are active for forwarding for a given code fragment.
- In schematic diagram of pipelined instructions show bubbles, stalls and forwarding
- Compute a physical address given a virtual address
- Determine if an address is in a cache (given cache contents)
- Determine if a page is in memory (given page table/TLB contents)
- Decide, given CPI and other data, which of two machines is faster
- Revise assembly code to be hazard-free