Course Summary

(CPEG323: Intro. to Computer System Engineering)

Course Evaluation!!!

- Now through Dec. 4, 2014

Instant replay

- The semester was split into roughly four parts.
  - The first quarter covered the basic knowledge of high level languages (i.e., C) and hardware designs.
  - The 2nd quarter covered instruction set architecture—the connection between software and hardware.
  - In the 3rd quarter of the course we discussed processor design. We focused on pipelining, which is one of the most important ways of improving processor performance.
  - Finally, we discussed large and fast memory systems (via caching), virtual memory.
- We also introduced many performance metrics to estimate the actual benefits of all of these fancy designs.

Some recurring themes

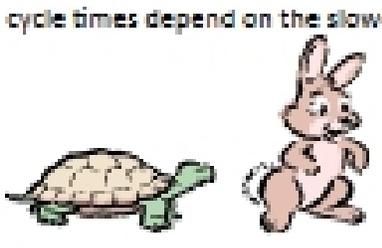
- There were several recurring themes throughout the semester.
  - Instruction set and processor designs are intimately related.
  - Parallel processing can often make systems faster.
  - Hierarchical designs combine different parts of a system.
  - Hardware and software depend on each other.

Instruction sets and processor designs

- The MIPS instruction set was designed for pipelining.
  - All instructions are the same length, to make instruction fetch and jump and branch address calculations simpler.
  - Opcode and operand fields appear in the same place in each of the three instruction formats, making instruction decoding easier.
  - Only relatively simple arithmetic and data transfer instructions are supported.
- These decisions have multiple advantages.
  - They lead to shorter pipeline stages and higher clock rates.
  - They result in simpler hardware, leaving room for other performance enhancements like forwarding, branch prediction, and on-die caches.

Performance

- First Law of Performance: Make the common case fast!
- But, performance is limited by the slowest component of the system.
- We've seen this in regard to cycle times in our CPU implementations.
  - Single-cycle clock times are limited by the slowest instruction.
  - Pipelined cycle times depend on the slowest individual stage.
- .



Hierarchical designs

- Hierarchies separate fast and slow parts of a system, and minimize the interference between them.
  - Caches are fast memories which speed up access to frequently-used data and reduce traffic to slower main memory. [Registers are even faster…]

Five things that I hope you will remember

- **Abstraction**: the separation of interface from implementation.
  - ISA's specify what the processor does, not how it does it.
- **Locality**:
  - Temporal Locality: "if you used it, you'll use it again"
  - Spatial Locality: "if you used it, you'll use something near it"
- **Caching**: buffering a subset of something nearby, for quicker access
  - Typically used to exploit locality.
- **Indirection**: adding a flexible mapping from names to things
  - Virtual memory's page table maps virtual to physical address.
- **Throughput** vs. **Latency**: (# things/time) vs. (time to do one thing)
  - Improving one does not necessitate improving the other.

Good luck on your exams
and have a great break!