Today’s Agenda

Course retrospective
- Revisit our bottom-up journey

Format and content of final exam
- Comprehensive, open book, similar format as midterm

Course evaluations
- Tells us what worked; what to improve for next time

Pizza!

Friday: four big concepts in processor design
- Caches, virtual memory, pipelining, multi-threading/-processing

Big Picture

Hardware
- Representing data, transistors, digital logic structures, von Neumann machine model

Assembly language
- Instructions, (structured) programming, input/output, relationship to hardware

C programming
- Syntax, operators, control structures, functions, pointers, data structures, relationship to assembly language

We’ve Come a Long Way...

How do computers work?
- Now you know!

Bottom-up approach
- Bits are bits
- Transistors
- Digital circuits (gates, latches, memories, state machines)
- Processor datapath
- Instructions
- Assembly programming
- C programming
  - Relationship to assembly
Beyond the Book
Transistors as pneumatic valves
Datapath
• “Working” single-cycle datapath
New LC-3 simulator
New I/O devices
• Video output
• Timer
Operating system (BreakOS)
• Unix history
• Input/output
• Trap table
• Memory protection with Memory Protection Register (MPR)
C Programming
• Comparisons to Java
• More function call stack discussion
• Security ramifications of C (smashing the stack example)
• Heap management discussion (malloc/free & garbage collection)

Course Assignments
Hardware
• HW#1: Data types
• HW#2: Digital logic
• HW#3: Hardware state machines
• HW#4: LC-3 instructions
Assembly Programming
• HW#5: Basic assembly
• HW#6: BreakOS
• HW#7: BreakOut
C Programming
• HW#8: LC-3 Disassembler
• HW#9: LC-3 Assembler (pseudo instructions)

This year’s CSE240
(Almost) All-new course (2nd time around)
• New textbook
• New lectures
• New assignments
• etc.

Course statistics
• 101 students
• 39 lectures
• 100+ hours of “office hours” (TAs and instructors)
• 1500+ e-mail messages in my “cse240” folder
  ➢ 15+ e-mails per day!
• 1435+ articles on the forum
Final Exam Format

Similar format as midterm exam
- Questions that span multiple concepts
- Synthesis of knowledge

...but longer
- Relatively shorter: 120 minute final vs 50 minute midterm

Open book/open note
- But study like it is not open book
- Use only as a reference and safety net

No hats!

Preparing for Exam

Review material
- Book (don’t forget Ch. 19!)
- Notes
- Midterm
- Homeworks

Work example questions
- End of chapter exercises (some answers online)
- Last year’s exam (handed out on Monday and online)
- Sample exams online (courtesy of Sanjay Patel)
  - Linked from course web page
  - Warning: some of the exams use LC-2 (which is different slightly from LC-3)

Optional review session
- 6:00pm in Wu and Chen on Mon, Dec 12th
- Chao and Joe

Final Exam Topics

Comprehensive
- With more of a focus on assembly and C
  - Especially relationship between assembly and C
  - Especially pointers, structs, function calls, and runtime stack
- Relationship between assembly/C and hardware

Types of questions
- Write an assembly or C function to do something
- Translating C to LC-3 assembly
  - Much like HW#6 & HW#7, but using C for pseudocode
- Translating LC-3 assembly to C code
- C to C translation
  - Convert a while loop to a for loop
- What does this code/circuit do? (circuit, assembly, C code)
- State machines (saw them in hardware & a C code example)
- Some second-chance questions from midterm exam
  - Example: datapath question from midterm

What’s Next?

Perhaps, nothing...
- An explicit course goal:
  - broad overview for students not continuing
- Touch on big ideas, bottom-up, no magic

CSE371/372: Computer Architecture
- Focus on performance and design of processors
- Caching, pipelining, parallelism, I/O devices, virtual memory
- Lab: design a processor in a hardware description language (HDL)

CSE380/381: Operating Systems
- Concepts of operating systems: threading, protection, file system, etc.
- Lab: write a Unix “shell”, implement parts of an operating system

CSE341: Compilers
- Translating a Java-like language to a real assembly language
- Automating much of what we talked about
- Techniques for parsing, translating, and optimizing code

Many other courses build on this course
- For example: networking, security, databases, embedded systems, etc.
Remainder of Today and Friday

Course evaluations
- Give us your feedback
- New format
  - We spent a lot of effort to try avoid rough edges
  - Tell us what to do better next time!

Pizza!

Friday:
- One lecture on the fundamental issues in processor design
- Caches, virtual memory, pipelining, multi-threading/-processing
- For those continuing and those not continuing in CSE