CSE 390B, Spring 2022

Building Academic Success Through Bottom-Up Computing

Midterm Review & Project 6 Overview

Midterm Topics Brainstorm, Midterm Practice Problems, Project 6 Overview, Hack CPU Logic Exercise

Lecture Outline

- **Midterm Topics Brainstorm**
  - **Starting a Study Plan for Exams**
- CSE 390B Midterm Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly
- Project 6 Overview
  - Project Tips and Workflow
- Hack CPU Logic Example: writeM
  - Project 6 CPU Logic Exercise

CSE 390B Midterm Topics Brainstorm

- Based on what we have covered thus far in class, what are topics, concepts, questions that you might expect to show up on next week's CSE 390B midterm?

Lecture Outline

- Midterm Topics Brainstorm
  - Starting a Study Plan for Exams
- **CSE 390B Midterm Practice Problems**
  - **Circuit Design, Writing Assembly, Tracing Assembly**
- Project 6 Overview
  - Project Tips and Workflow
- Hack CPU Logic Example: writeM
  - Project 6 CPU Logic Exercise

CSE 390B Review Session

- Step 1: Work on the problem individually for ten minutes
- Step 2: Discuss in groups your problem-solving approach
- Step 3: Discuss tips for approaching this exam problem and write it on the whiteboard
- Step 4: Identify two people in your group to walk through the problem and present tips for solving the problem

Review Session Debrief

- Based on your experience with this exercise, how does it inform how you approach your studying?

- What resources can you utilize to help you deepen your understanding?

Previous CSE 390B Midterms

- Four midterms from previous quarters
  - 20sp midterm likely more difficult than midterm this quarter
  - Midterm from 21wi, 21sp, and 22wi are more similar to what this quarter's midterm will look like
- 20sp midterm recommended to become familiar with problem types
- 21wi, 21sp, and 22wi midterms recommended for practicing a timed exam
  - Set a timer for 50 minutes and take the exam in its entirety
  - Helps practice time management and simulate exam environment

Lecture Outline

- Midterm Topics Brainstorm
  - Starting a Study Plan for Exams
- CSE 390B Midterm Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly
- **Project 6 Overview**
  - **Project Tips and Workflow**
- Hack CPU Logic Example: writeM
  - Project 6 CPU Logic Exercise

Project 6: Overview

- Part I: Mock Exam Problem

- Part II: Building a Computer
  - **`LoadAReg.hdl`**, **`LoadDReg.hdl`** (Easier)
  - **`JumpLogic.hdl`** (Medium)
  - **`CPU.hdl`** (Harder)
  - **`Computer.hdl`** (Easier)

- Part III: Project 6 Reflection

Project 6, Part I: Mock Exam Problem

- Your group will meet for a 30-minute session to do one mock exam problem
  - Your group's mock exam problem will be emailed right before your session

- Your 30-minute session will include:
  - Set up: 5 minutes
  - Mock Exam Problem: 10 minutes
  - Debrief & Reflection: 15 minutes

- Part I Task: Submit the completed mock exam problem and complete the reflection questions

Project 6 Tips

- **`CPU.hdl`**: We provide an overview diagram, but there are details to fill in, especially control
  - Draw your own detailed diagram first
  - Handling jumps will require a lot of logic—sketch out the cases
  - Textbook chapter 4 and 5 helpful for Project 6

- Multi-Bit Buses: MSB to the left, LSB to the right
  - Important to keep in mind when taking apart the instruction

- Debugging: Consult .out and .cmp files to debug, then look at internal wires in simulator
  - See also the "Debugging tips" section of the specification

Hack CPU Logic

- How do we determine the unimplemented logic for the CPU (all of the c's in the diagrams)?
- Refer to the assembly specification
- Project 6 will requires understanding of textbook chapters to determine how to use the instruction bits to implement the control logic
  - Helpful textbook sections: 4.2.2, 4.2.3, and 5.3.1

Hack CPU Logic Workflow

- Step 1: What do we pay attention to?
  - Usually, will be some combination of instruction bits or intermediate outputs
  - These are the "inputs" to your sub-problem

- Step 2: Determine logic for the part you are working on
  - Uses the "inputs" from Step 1
  - Usually requires reading a relevant section of the textbook or assembly specification

Lecture Outline

- Midterm Topics Brainstorm
  - Starting a Study Plan for Exams
- CSE 390B Midterm Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly
- Project 6 Overview
  - Project Tips and Workflow
- **Hack CPU Logic Example: writeM**
  - **Project 6 CPU Logic Exercise**

Hack CPU Logic Example: writeM

- Example: Determine when **`writeM`** should be set to 1

- Step 1: What do we pay attention to?
  - **`writeM`** is related to whether we write to memory or not
  - We need to look up the destination bits specification from Chapter 4

d1	d2	d3	*Mnemonic*	*Destination (where to store the computed value)*
0	0	0	null	The value is not stored anywhere
0	0	1	M	Memory[A] (memory register addressed by A)
0	1	0	D	D register
0	1	1	MD	Memory[A] and D register
1	0	0	A	A register
1	0	1	AM	A register and Memory[A]
1	1	0	AD	A register and D register
1	1	1	AMD	A register, Memory[A], and D register

Figure 4.4 The *dest* field of the *C*-instruction.

Hack CPU Logic Example: writeM

- Example: Determine when **`writeM`** should be set to 1

- Step 2: Determine logic for specification
  - Read the "Destination Specification" section of Chapter 4
  - Instruction bits:

```
1 1 1 a c1 c2 c3 c4 c5 c6 d1 d2 d3 j1 j2 j3
```

d1	d2	d3	*Mnemonic*	*Destination (where to store the computed value)*
0	0	0	null	The value is not stored anywhere
0	0	1	M	Memory[A] (memory register addressed by A)
0	1	0	D	D register
0	1	1	MD	Memory[A] and D register
1	0	0	A	A register
1	0	1	AM	A register and Memory[A]
1	1	0	AD	A register and D register
1	1	1	AMD	A register, Memory[A], and D register

Figure 4.4 The *dest* field of the *C*-instruction.

Hack CPU Implementation: Logic Sub Chips

- We provide three sub chips and tests that implement the control logic for the A Register, D Register, and PC
  - **`LoadAReg`** contains logic for loading the A Register
  - **`LoadDReg`** contains logic for loading the D Register
  - **`JumpLogic`** contains logic for determining if the PC should load, jump, or increment

- Implement and test these first, then use them in your CPU implementation
  - Intended to help you narrow the scope of bugs

Lecture 10 Wrap-up

- Crossing the mid-quarter bridge to Week 6!
  - Metacognitive Subject: Exam-Taking Strategies
  - Technical Subject: Review of Technical Subjects

- CSE 390B Midterm next Thursday (5/5) during lecture

- Project Reminders
  - Project 4 grades and feedback released on Gradescope
  - **Project 5 due tonight (4/28) at 11:59pm PDT**
  - Project 6 (Mock Exam Problem & Building a Computer) released, due **in two weeks on Thursday (5/12)** at 11:59pm PDT