CSE 390 B Spring 2021

Exam Problem Review & Project 5

Exam Review Session, Project 5 demo, CPU Logic

CSE 390B Midterm Brainstorm

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

Agenda

- 390B Exam Review Session
- Project 5 Overview
- CPU Logic Review and Practice

Agenda

- **390B Exam Review Session**
- Project 5 Overview
- CPU Logic Review and Practice

CSE 390B Review Session

Review Session Activity

- In breakouts work through three potential exam problems, spending ~12-15min per problem
  - Problem #1: Circuit Design
  - Problem #2: Writing Hack Assembly Programs
  - Problem #3: Tracing Hack Assembly Programs

- For each problem:
  - You will spend **5 min** working on the problem individually
  - You will then check in with each other and see if anyone has questions or is having difficulty getting started
  - You will spend **3-4 min** more working on the problem, either together or individually
  - You will then create a list of strategies/tips for that problem type (edit the slide corresponding to the current question)

Problem #1: Circuit Design

- Tips for working through these types of problems (how to approach them, things to remember, ways to get partial credit, etc.)
  -

Problem #2: Writing Hack Assembly

- Tips for working through these types of problems (how to approach them, things to remember, ways to get partial credit, etc.)
  -

Problem #3: Tracing Hack Assembly

- Tips for working through these types of problems (how to approach them, things to remember, ways to get partial credit, etc.)
  -

What now?

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

- Identify **one** exam problem type/concept that you want to prioritize in your studying
  - What resources can you utilize to help you deepen your understanding?

Previous CSE 390B Midterms

- We have two old midterms from previous quarters
  - Spring 2020 is likely more difficult than the midterm this quarter
  - Winter 2021 is more similar to what this quarter's midterm will look like

- We recommend you use Spring 2020's midterm to become familiar with problem types and gauge what you need to study and practice more

- We recommend you use Winter 2021's midterm as a chance to practice taking a timed exam
  - Set a timer for 50 min and fully take the exam
  - Will help you practice time management

Agenda

- 390B Exam Review Session
- **Project 5 Overview**
- CPU Logic Review and Practice

Project 5: Overview

- Timed Mock Exam Problem
- Build a Computer!
  - LoadAReg.hdl, LoadDReg.hdl -- Easier
  - JumpLogic.hdl -- Mediumish
  - CPU.hdl -- Quite Hard
  - Computer.hdl -- Easier
- Social Computing Reflection II and Annotation
  - New prompts, plus using at least three annotation strategies while reading your article
- Project 5 is due in two weeks. Don't delay in getting started!

Note: Project 3 grades have been released. We encourage you all to review the feedback that was given

Project 5: Timed Mock Exam Problem

- Your group will meet for a 30-min session to do **one** mock exam problem
  - Your group's mock exam problem will be emailed right before your session
- Your 30-min session will include:
  - Set-Up: 5 minutes
  - Mock Exam Problem: 10 minutes
  - Debrief & Reflection: 15 minutes
- Complete and submit the reflection questions

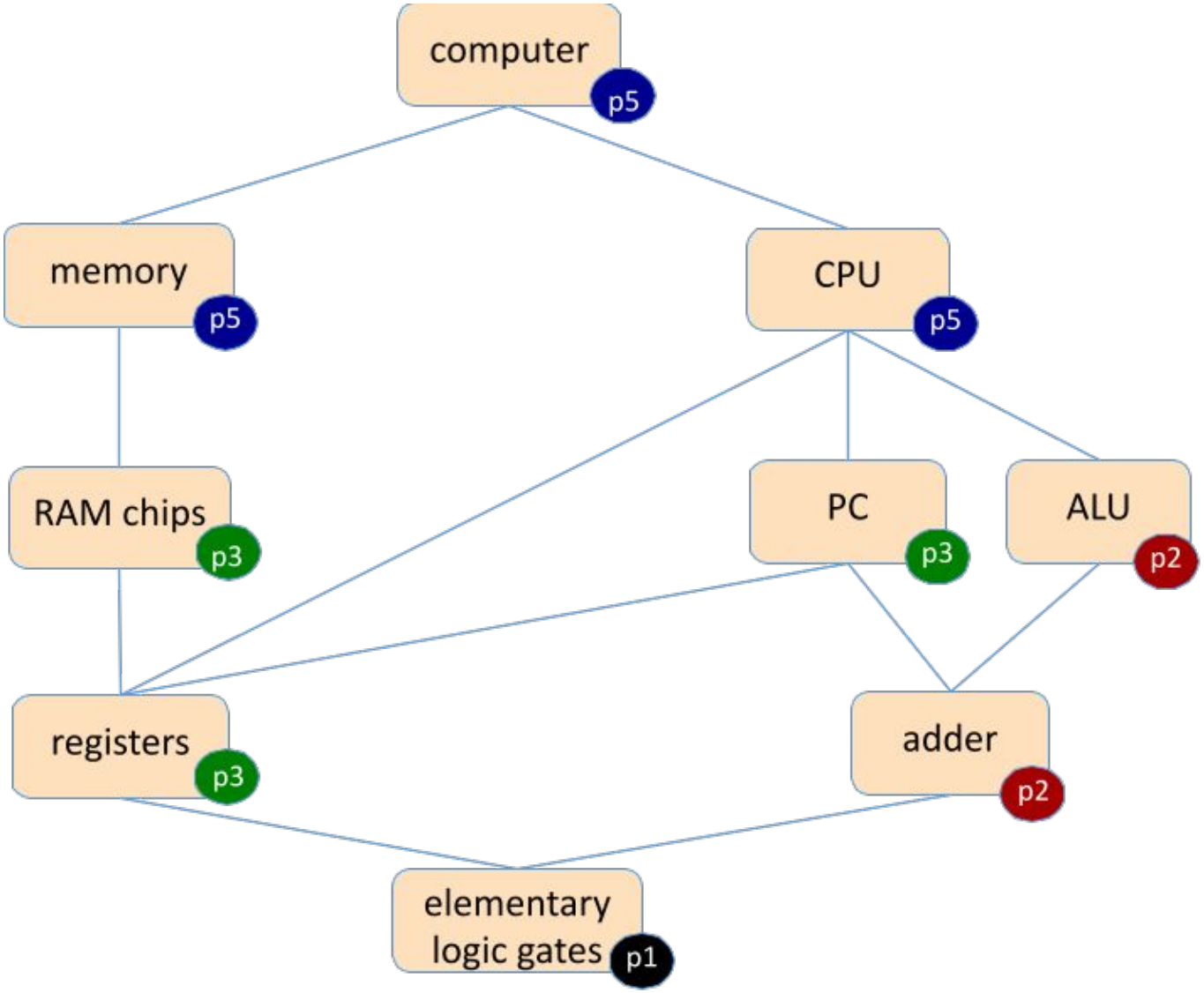
Project 5 Tips

- **CPU.hdl**: we provide an overview diagram, but there are *plenty* of details to fill in (especially for control)!
  - Very interconnected → tough to split up
  - Instead, crucial to draw your own detailed diagram first!
  - Handling jumps will require a lot of logic; make sure to sketch out all possible cases
  - Chapter 4 and 5 are going to be extremely useful!!!

- **Multi-Bit Buses**: we typically write MSB to the left and LSB to the right: 15 14 13 12 11 10 … 3 2 1 0
  - Important to keep in mind when taking apart the instruction!

- **Debugging**: when something goes wrong, consult .out and .cmp files. Then, look at internal wires in simulator
  - See also the "Debugging tips" section of the spec

How Far You've Come!



Agenda

- Approaching Exam Review Sessions
- 390B Exam Review Session
- Project 5 Overview
- **CPU Logic Review and Practice**

Hack CPU Logic

- How do we determine the unimplemented logic for the CPU (all of the c's in the diagrams)?
- Need to refer to the assembly specification!
- Project 5 will require a good bit of consulting of Chapter 4 to figure out how to use the instruction bits to implement the control logic

Hack CPU Logic Workflow

- Step 1: Figure out what to pay attention to
  - Usually will be some combination of instruction bits and/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/assembly specification

Instruction Bits: A-instruction

```
16 bits: 0 v v v v v v v v v v v v v v v
```

- Most significant bit is a 0 (indicates an A-instruction)
- Rest of the bits are the value to be loaded
  - Since most significant bit is 0, entire A-instruction is also the value to be loaded

Instruction Bits: C-instruction

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

- Most significant bit is a 1 (indicates a C-instruction)
- Next two most significant bits aren't used (always 1)
- a-bit and c-bits are related to computations
- d-bits are related to destination locations
- j-bits are related to jumping

Hack CPU Logic Example: writeM

- Example: determining when writeM should be set to 1

- Step 1: figure out what to pay attention to
  - writeM is related to where we store the output, or what **destination** we use
  - We need to look up the destination bits specification from Chapter 4!

Hack CPU Logic Example: writeM

- Example: determining when writeM should be set to 1

- Step 2: determine logic for specification
  - Read the "Destination Specification" section of Chapter 4
  - d3 determines if the output should be written to memory
  - Which bit of our instruction is that???
  - Instruction bits:

    1 1 1 a c1 c2 c3 c4 c5 c6 d1 d2 d3 j1 j2 j3
  - So writeM = instruction[3]?

Hack CPU Logic Example: writeM

- Example: determining when writeM should be set to 1

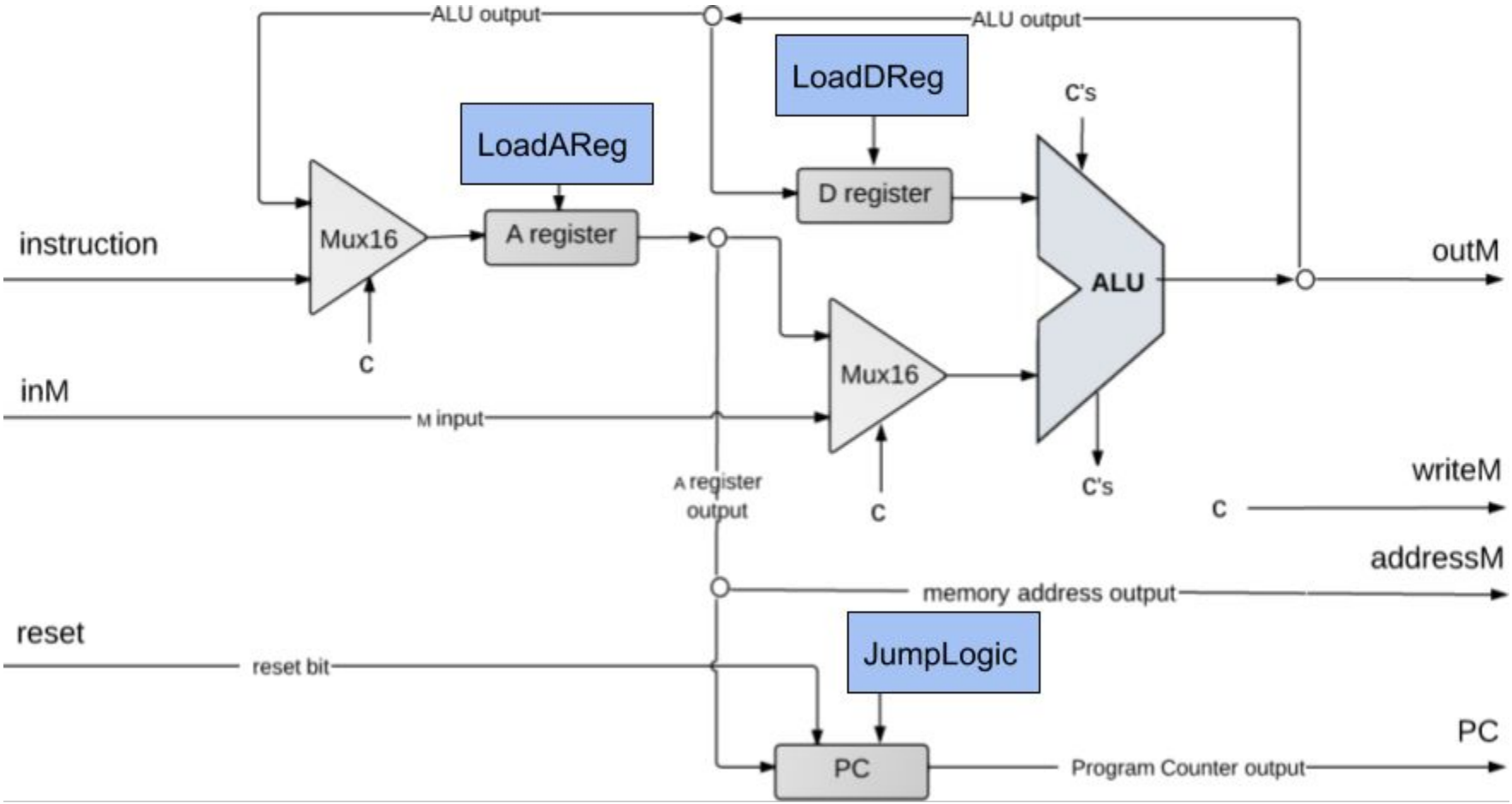
- Not so fast…
  - What happens if it's an A-instruction?
  - We only write to destinations in the case of a C-instruction
  - So writeM = C-instruction & instruction[3]
  - Remember that certain actions only occur on certain instruction types, you may have to include a check for instruction type in your logic depending on the action!

Hack CPU Implementation: Logic sub-chips

- We provide you with 3 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/test these first, then use them in your CPU implementation!
  - Intended to help you narrow the scope of any bugs you may have

Hack CPU Implementation: Logic sub-chips



Hack CPU: LoadAReg and LoadDReg

- Now we will give you some time to work on LoadDReg and LoadAReg in groups
  - Will require you to run "git pull" to retrieve the starter code
- We recommend starting with LoadDReg
- Use lecture examples and ask questions if you don't know where to begin!
  - Working together might be nice because you can all share when you've potentially found relevant information

Wrapping Up

What's in store for Week 6?

- Test-Taking Strategies via a mock exam
- Midterm on Thursday, May 6th

Reminders

- Project 3 Grades Released on Gradescope
- Project 4 Due Tonight 11:59PM PDT

Approaching Review Sessions

PREPARATION

- **Reviewing exam review guides / past exams**
- **Reviewing lecture notes and past homework assignments**
- **Doing sample exam questions**

What remains unclear / confusing that your review session could help clarify?

Approaching Review Sessions

PREPARATION

- **Reviewing exam review guides / past exams**
- **Reviewing lecture notes and past homework assignments**
- **Doing sample exam questions**

What remains unclear / confusing that your review session could help clarify?

GENERATING QUESTIONS

- **Asking questions that help you understand the significance of a particular topic/concept**
- **Asking for examples that help you see a concept applied in a new way**

Writing your questions down ahead of time or as you think of them can be helpful to reference during the review session

Approaching Review Sessions

PREPARATION

- **Reviewing exam review guides / past exams**
- **Reviewing lecture notes and past homework assignments**
- **Doing sample exam questions**

What remains unclear / confusing that your review session could help clarify?

GENERATING QUESTIONS

- **Asking questions that help you understand the significance of a particular topic/concept**
- **Asking for examples that help you see a concept applied in a new way**

Writing your questions down ahead of time or as you think of them can be helpful to reference during the review session

ACTIVE PARTICIPATION

- **Going through the process of writing down notes, example questions and if given the space to solve, solve them!**
- **Mark the questions that you were able to get answered within the review session. Identify what's left**

Bloom's taxonomy! Active participation can engage the higher levels of learning.