CSE 390B, Autumn 2022

Building Academic Success Through Bottom-Up Computing

Building a Computer & Midterm Practice

Building a Computer, Hack CPU Interface, Midterm Topics Brainstorm and Practice Problems, Project 6 Overview

Lecture Outline

- **Building a Computer**
  - **Architecture, Fetch and Execute Cycle**
- Hack CPU Interface
  - Implementation and Operations
- CSE 390B Midterm Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly
- Project 6 Overview
  - Project Tips and Workflow

Building a Computer

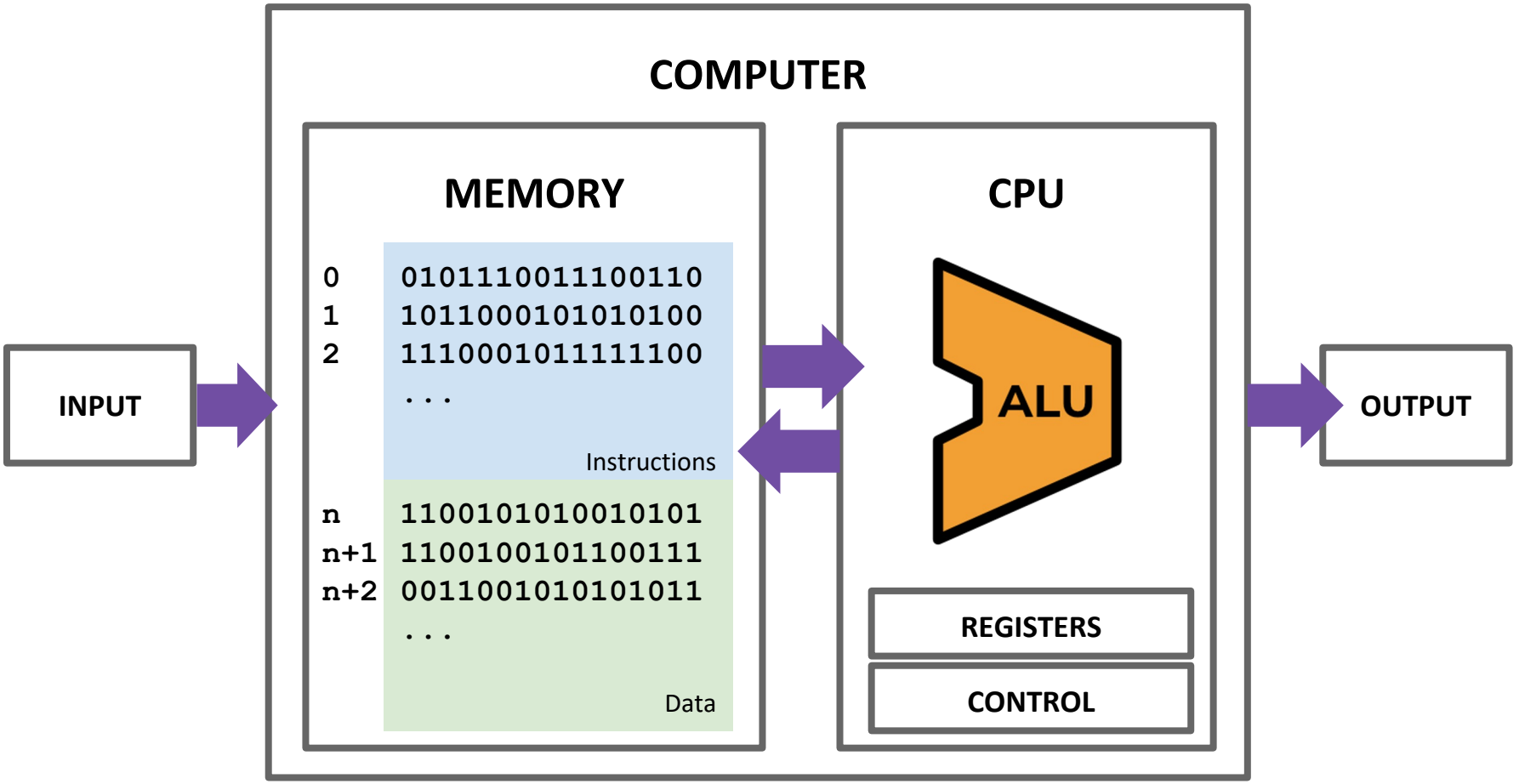
- All your hardware efforts are about to pay off!

- Perspective: **BUILDING A COMPUTER**

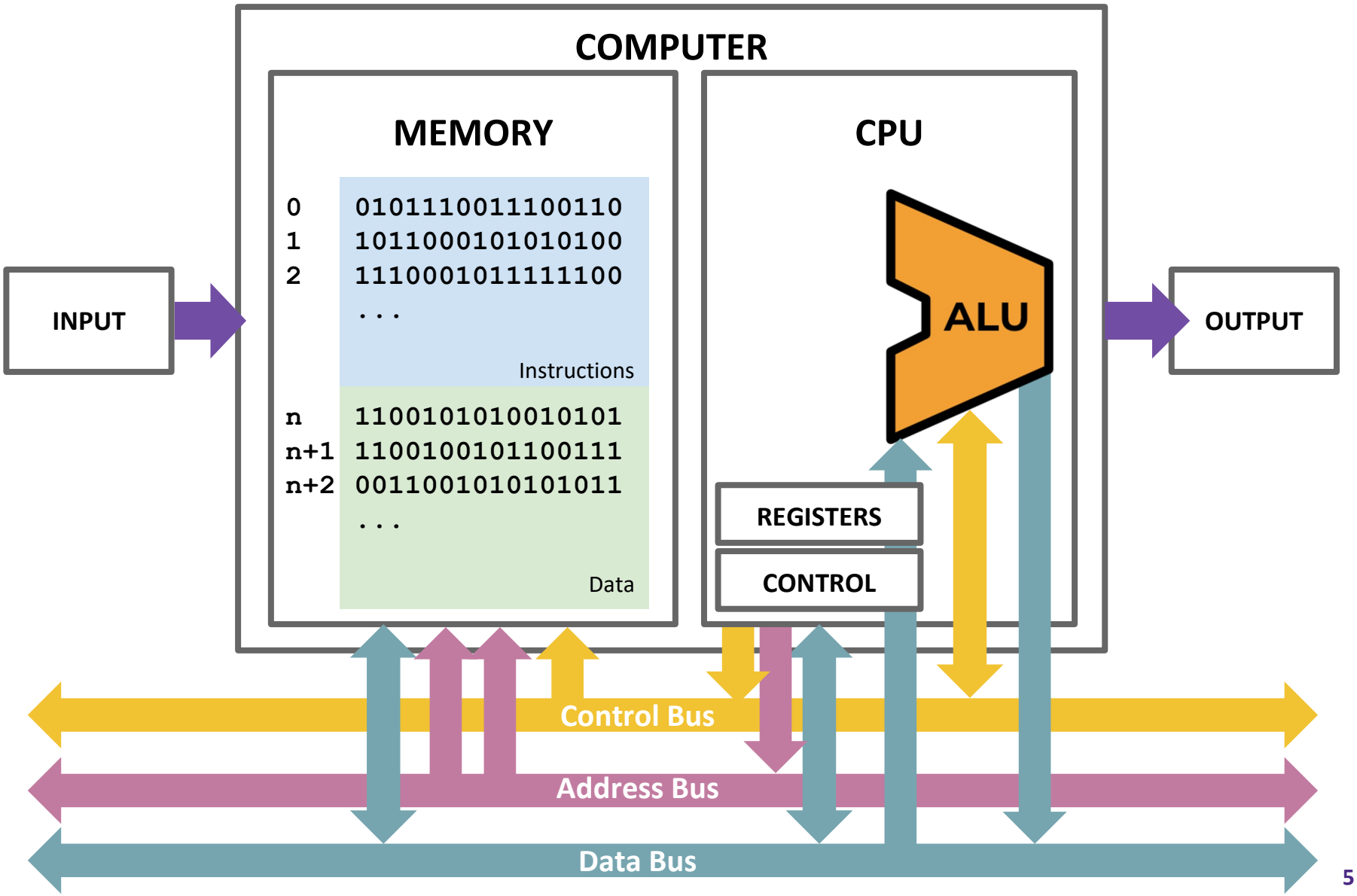
- In Project 6, you will build **`Computer.hdl`**, the final, top-level chip in this course
  - For all intents and purposes, a real computer
  - Simplified, but organization very similar to your laptop

- Project 7 onward, we will write software to make it useful

Von Neumann Architecture



Connecting the Computer: Buses

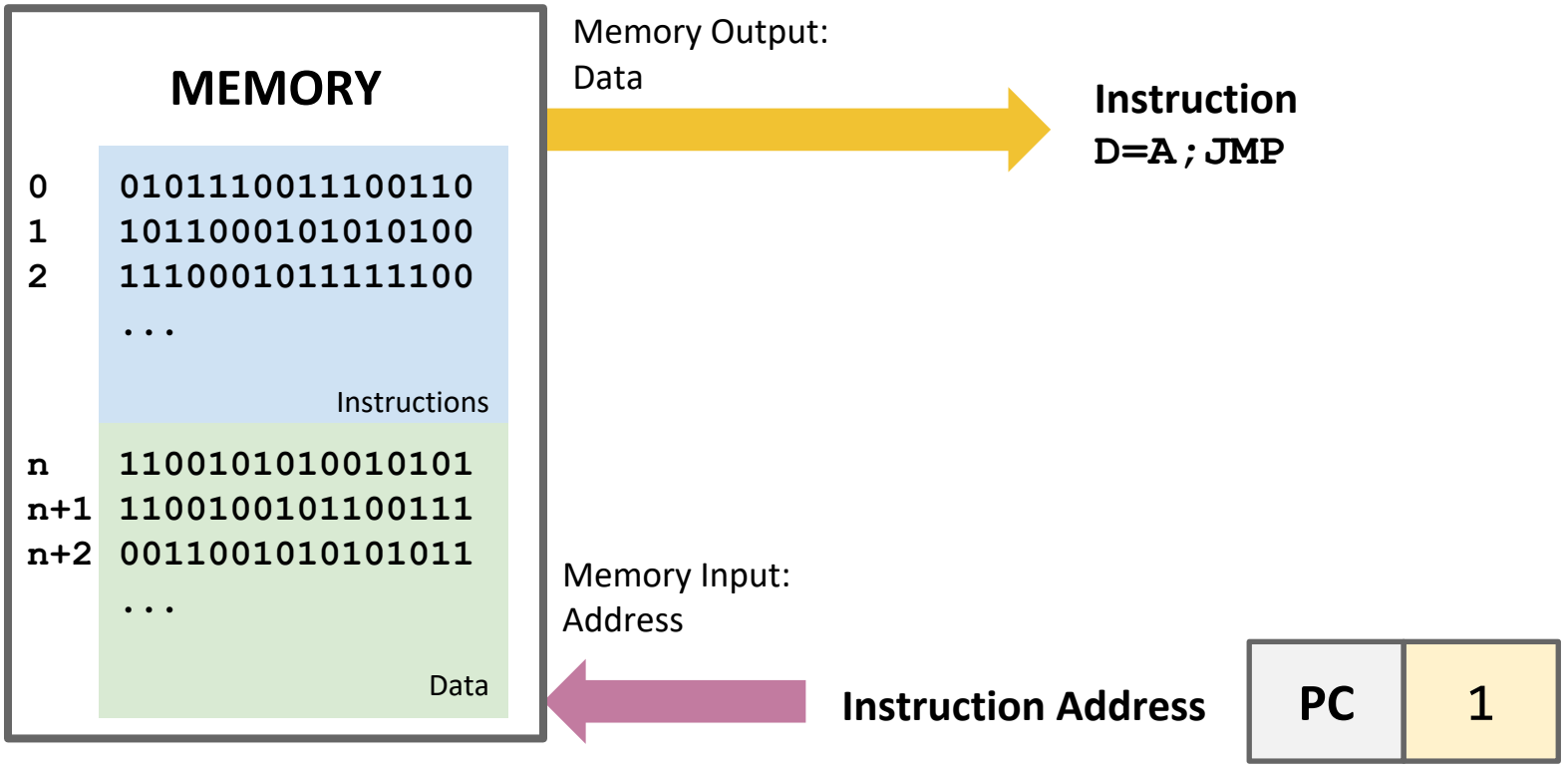


Basic CPU Loop

- Repeat forever:
  - **Fetch** an instruction from the program memory
  - **Execute** that instruction

Fetching

- Specify which instruction to read as the address input to our memory
- Data output: actual bits of the instruction

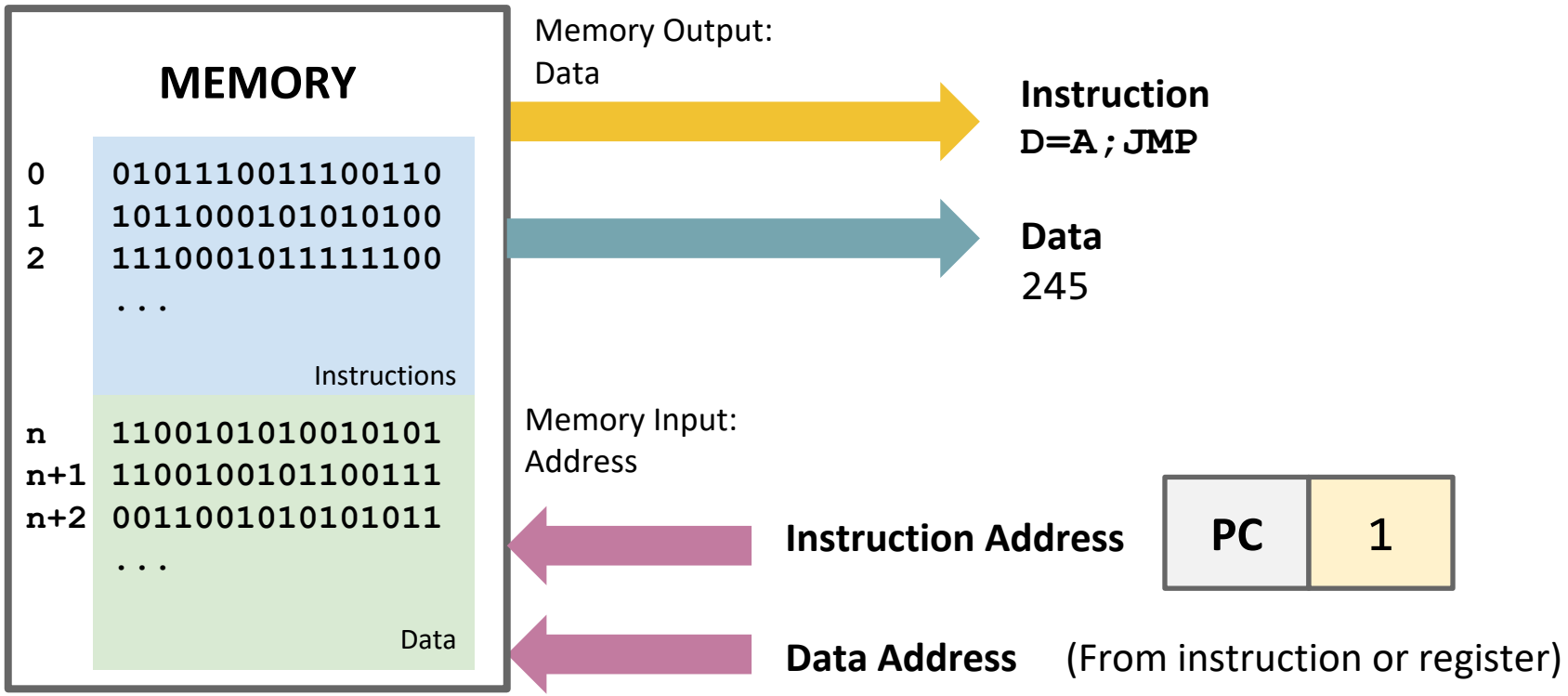


Executing

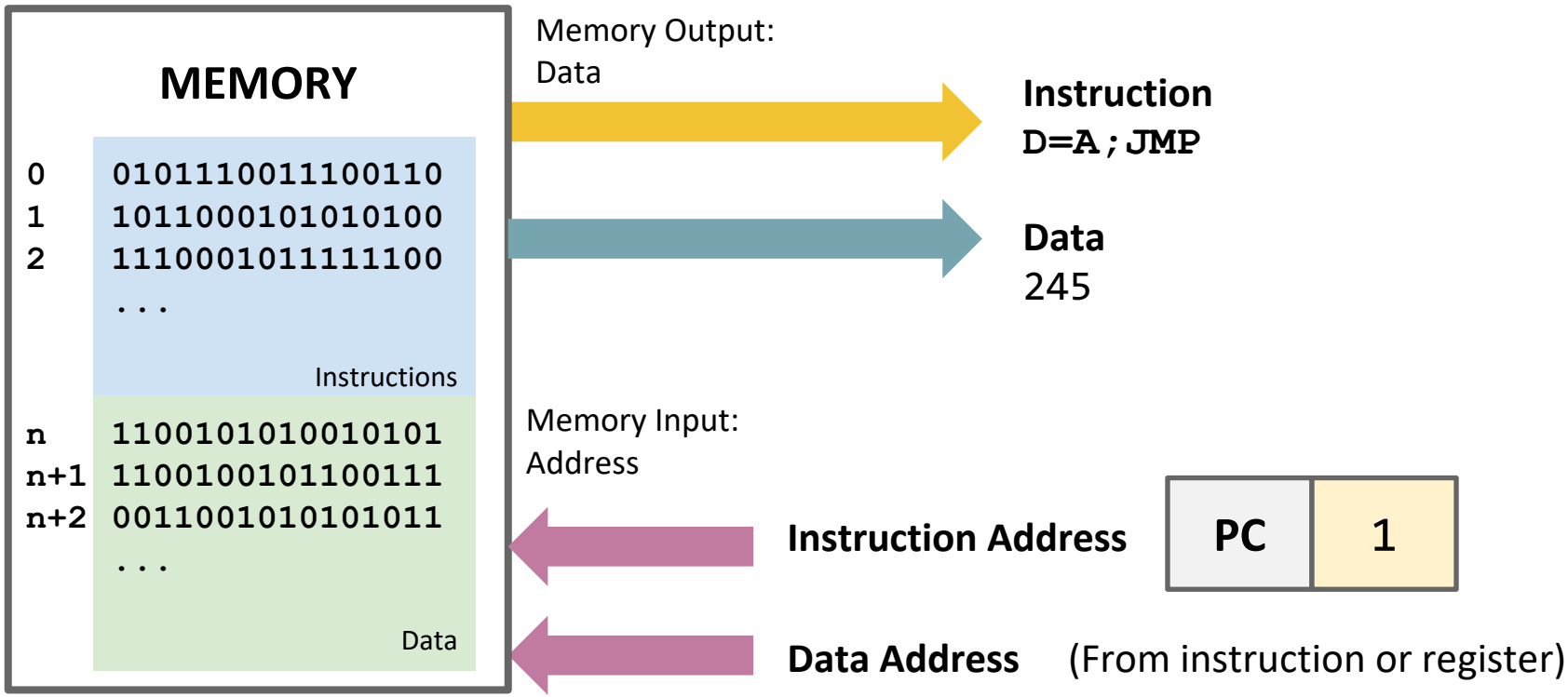
- The instruction bits describe exactly "what to do"
  - A-instruction or C-instruction?
  - Which operation for the ALU?
  - What memory address to read? To write?
  - If I should jump after this instruction, and where?

- Executing the instruction involves data of some kind
  - Accessing registers
  - Accessing memory

Combining Fetch & Execute

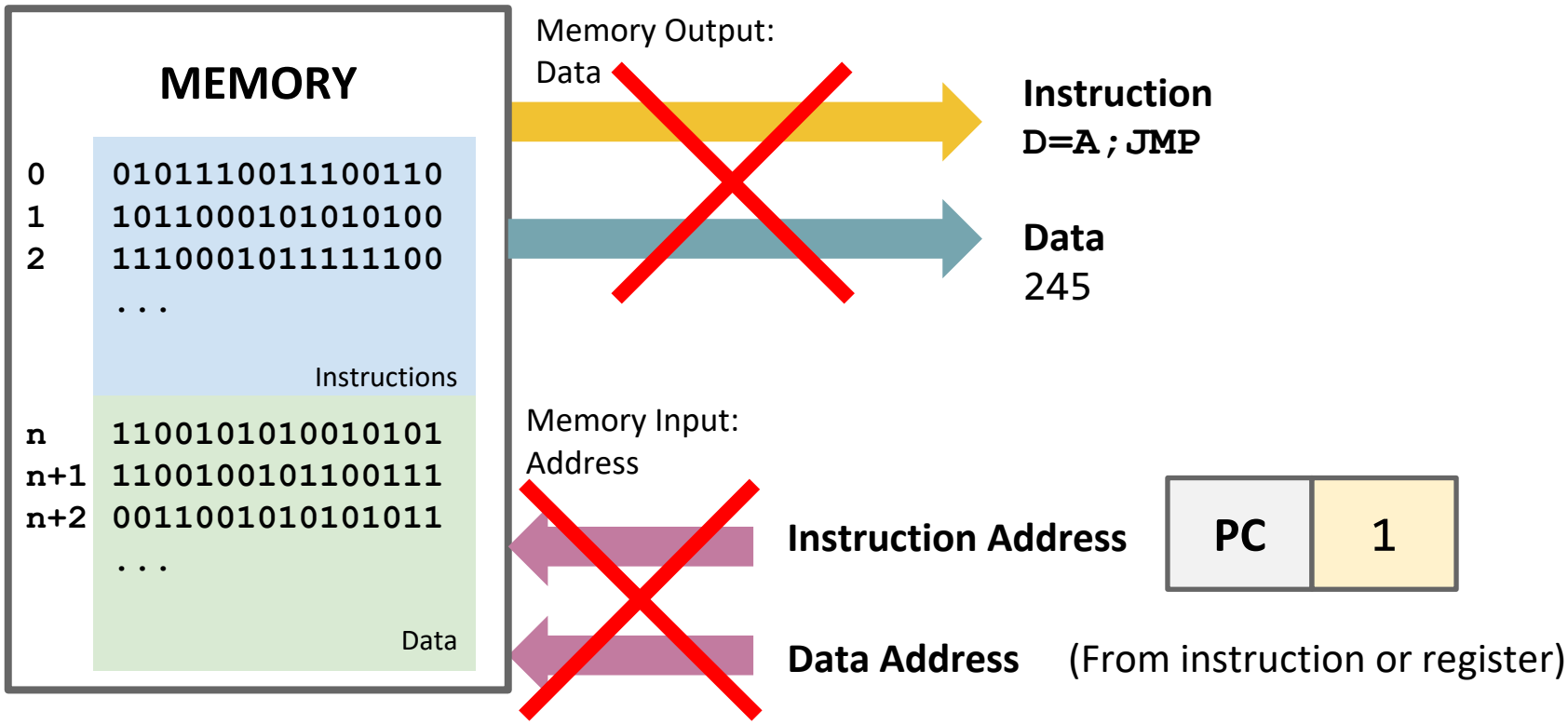


Combining Fetch & Execute



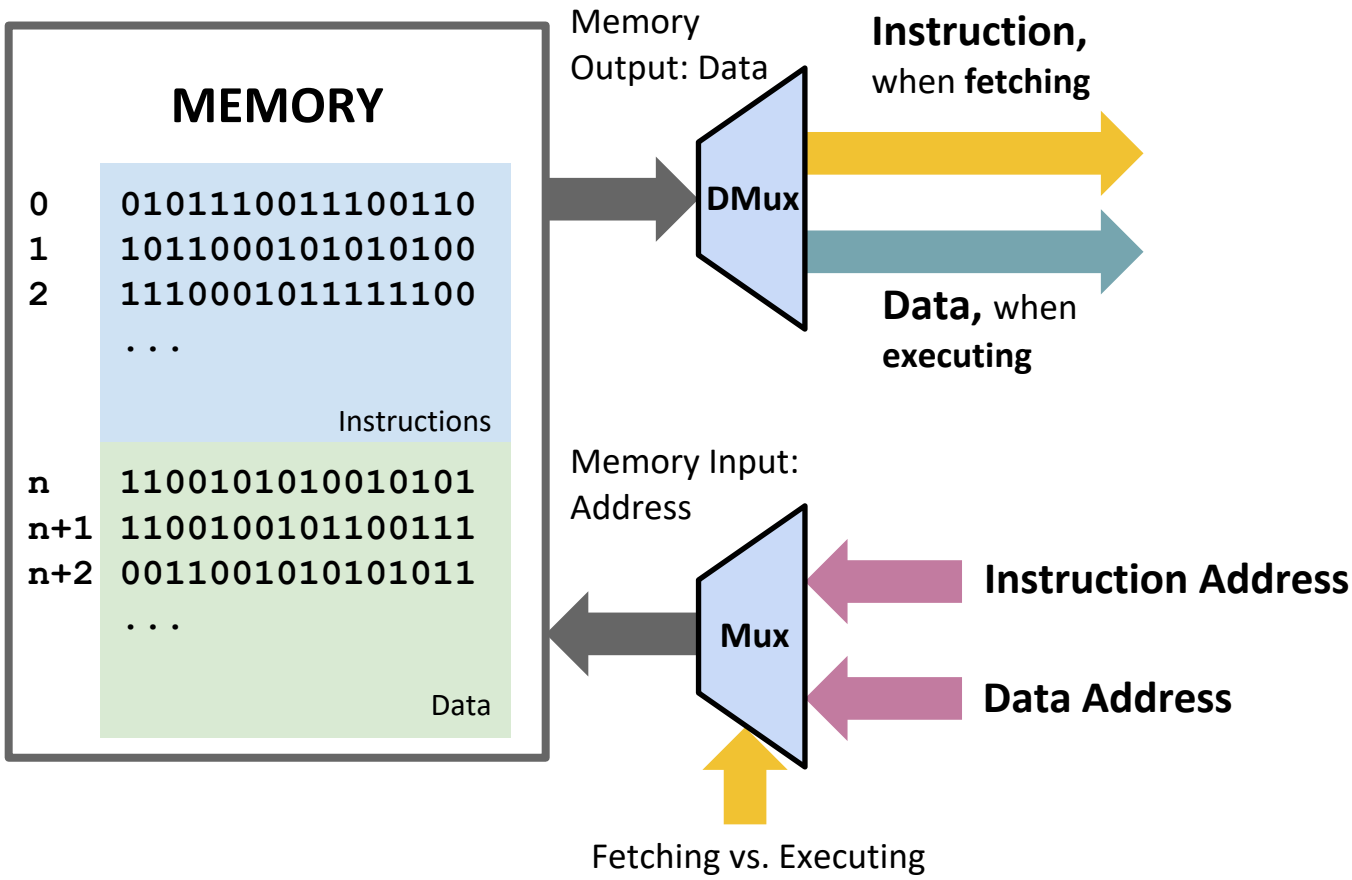
- Could we implement with **`RAM16K.hdl`**?
  - (Hint: Think about the I/O of RAM)

Combining Fetch & Execute



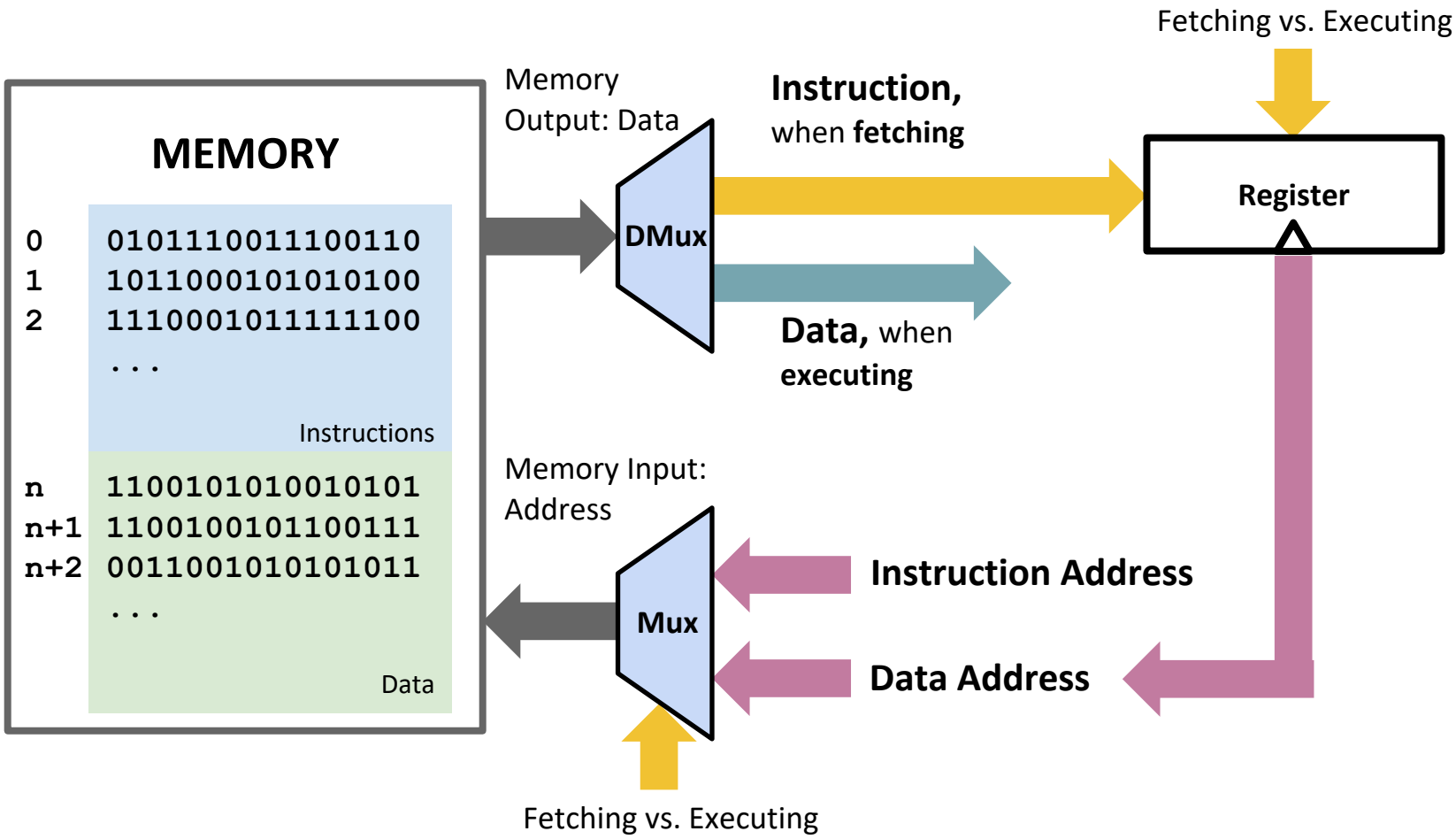
- Could we implement with **`RAM16K.hdl`**?
  - **No!** Our memory chips only have one input and one output

Solution 1: Handling Single Input / Output



- Can use multiplexing to share a single input or output

Solution 1: Fetching / Executing Separately



- Need to store fetched instruction so it's available during execution phase

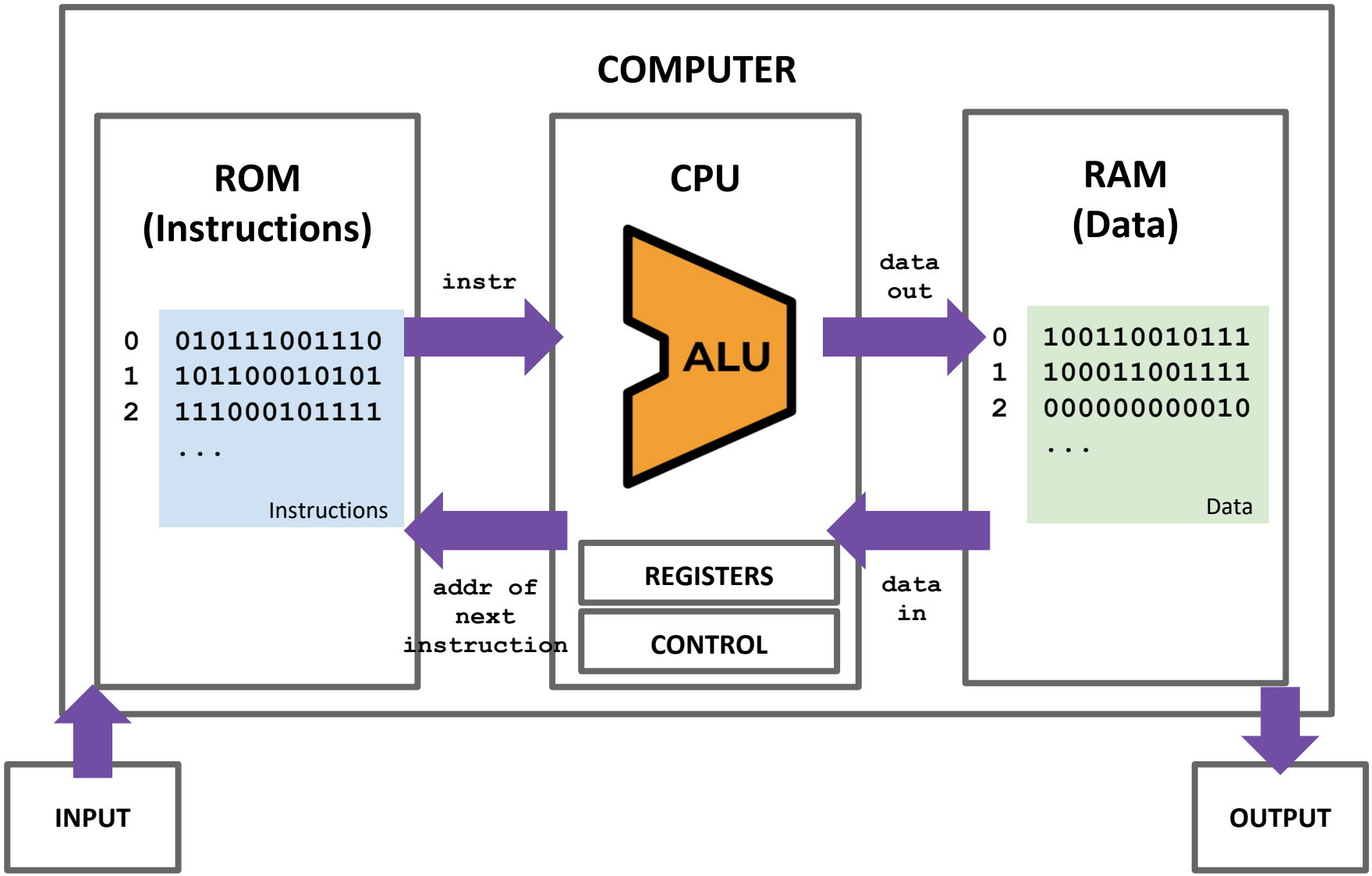
Solution 2: Separate Memory Units

- Separate instruction memory and data memory into two different chips
  - Each can be independently addressed, read from, written to
- Pros:
  - Simpler to implement
- Cons:
  - Fixed size of each partition, rather than flexible storage
  - Two chips → redundant circuitry

Lecture Outline

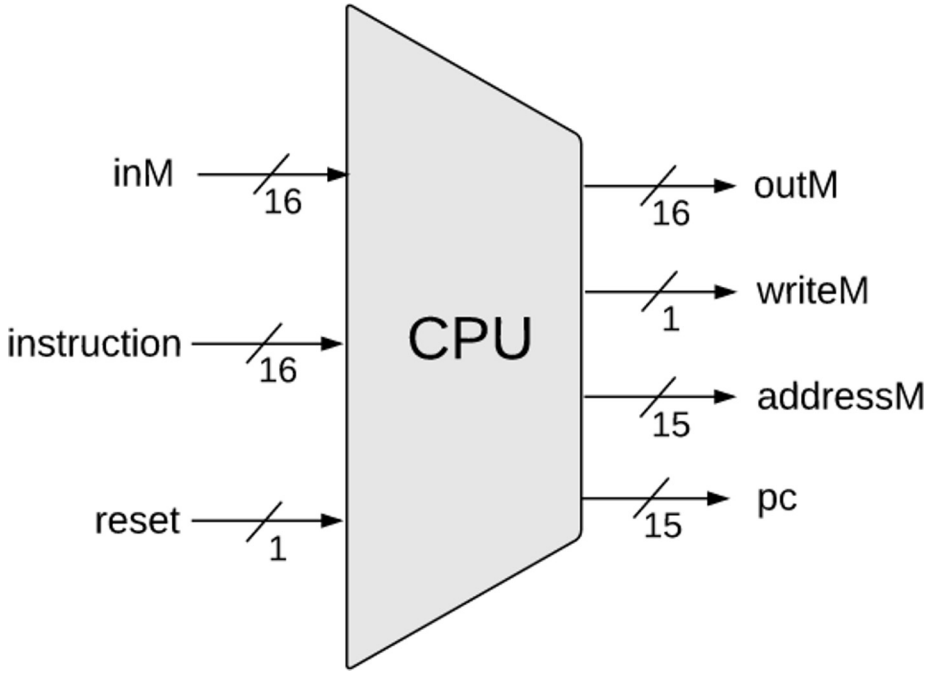
- Building a Computer
  - Architecture, Fetch and Execute Cycle
- **Hack CPU Interface**
  - **Implementation and Operations**
- CSE 390B Midterm Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly
- Project 6 Overview
  - Project Tips and Workflow

Hack CPU


COMPUTER
ROM
(Instructions)
0 010111001110
1 101100010101
2 111000101111
...
Instructions
instr
CPU
ALU
REGISTERS
CONTROL
addr of
next
instruction
data
out
data
in
RAM
(Data)
0 100110010111
1 100011001111
2 000000000010
...
Data
INPUT
OUTPUT

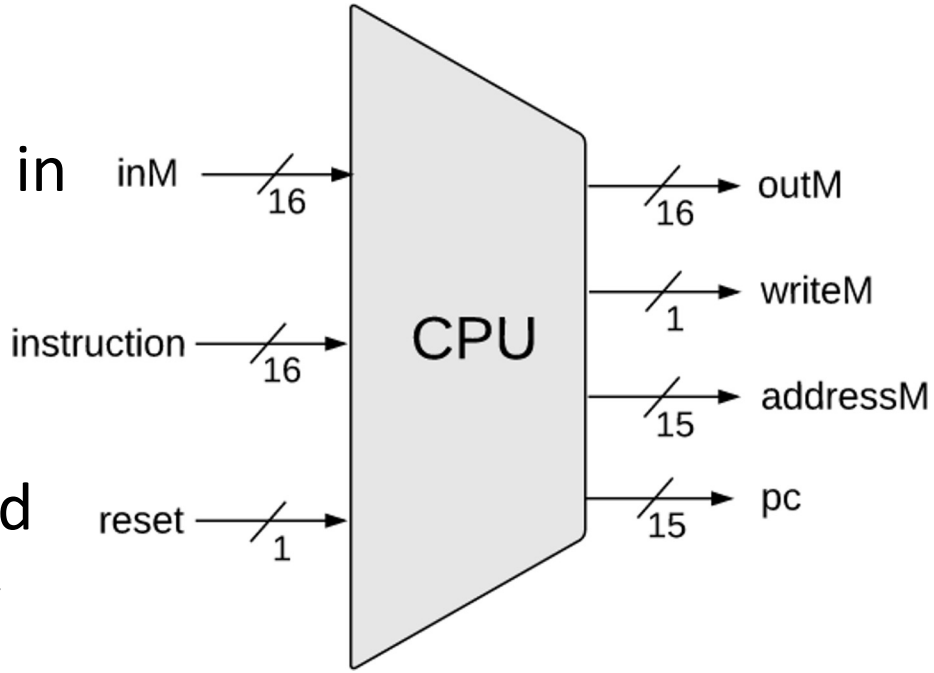
Hack CPU Interface Inputs

- **`inM`**: Value coming from memory
- **`instruction`**: 16-bit instruction
- **`reset`**: if 1, reset the program

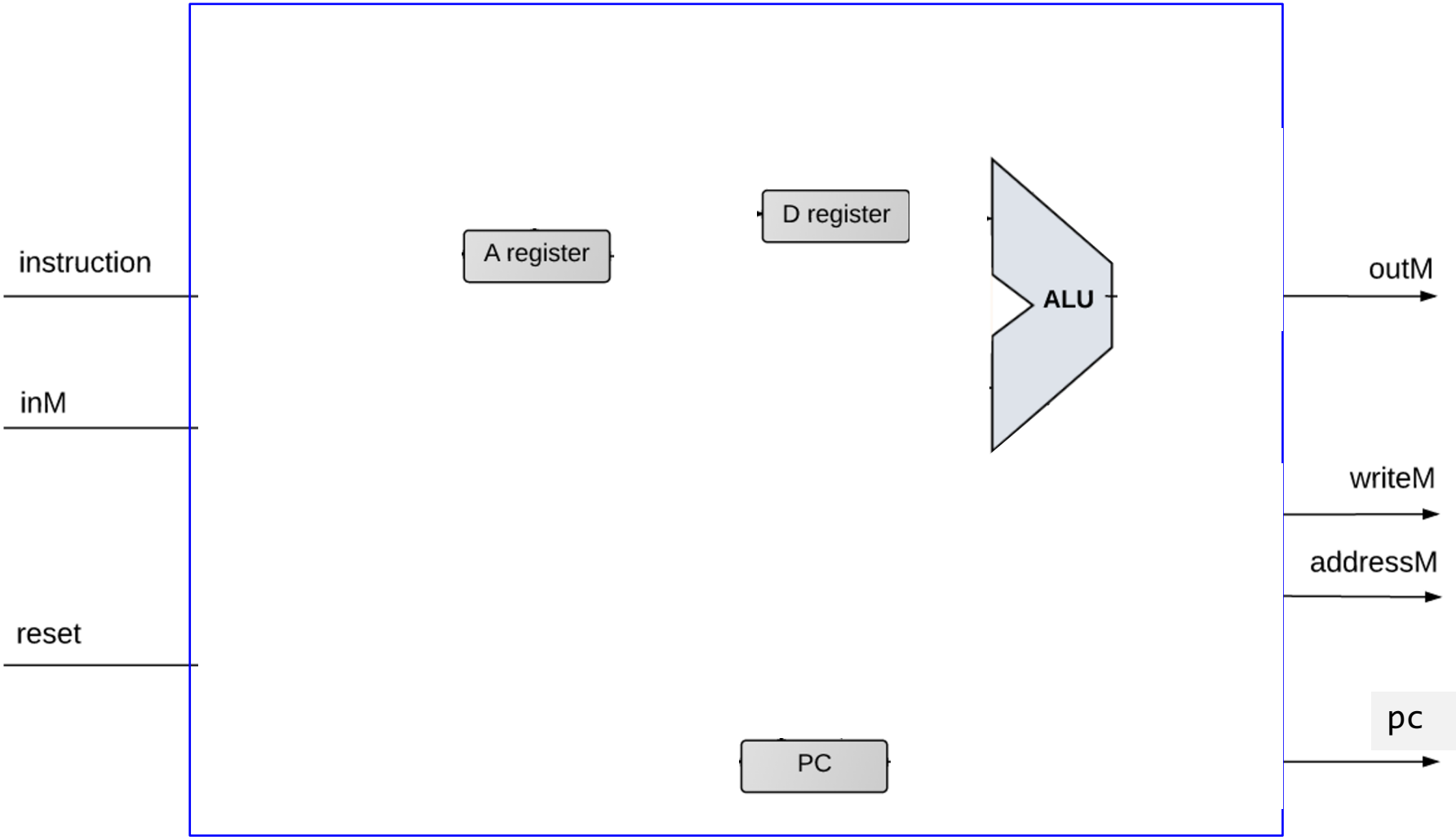


Hack CPU Interface Outputs

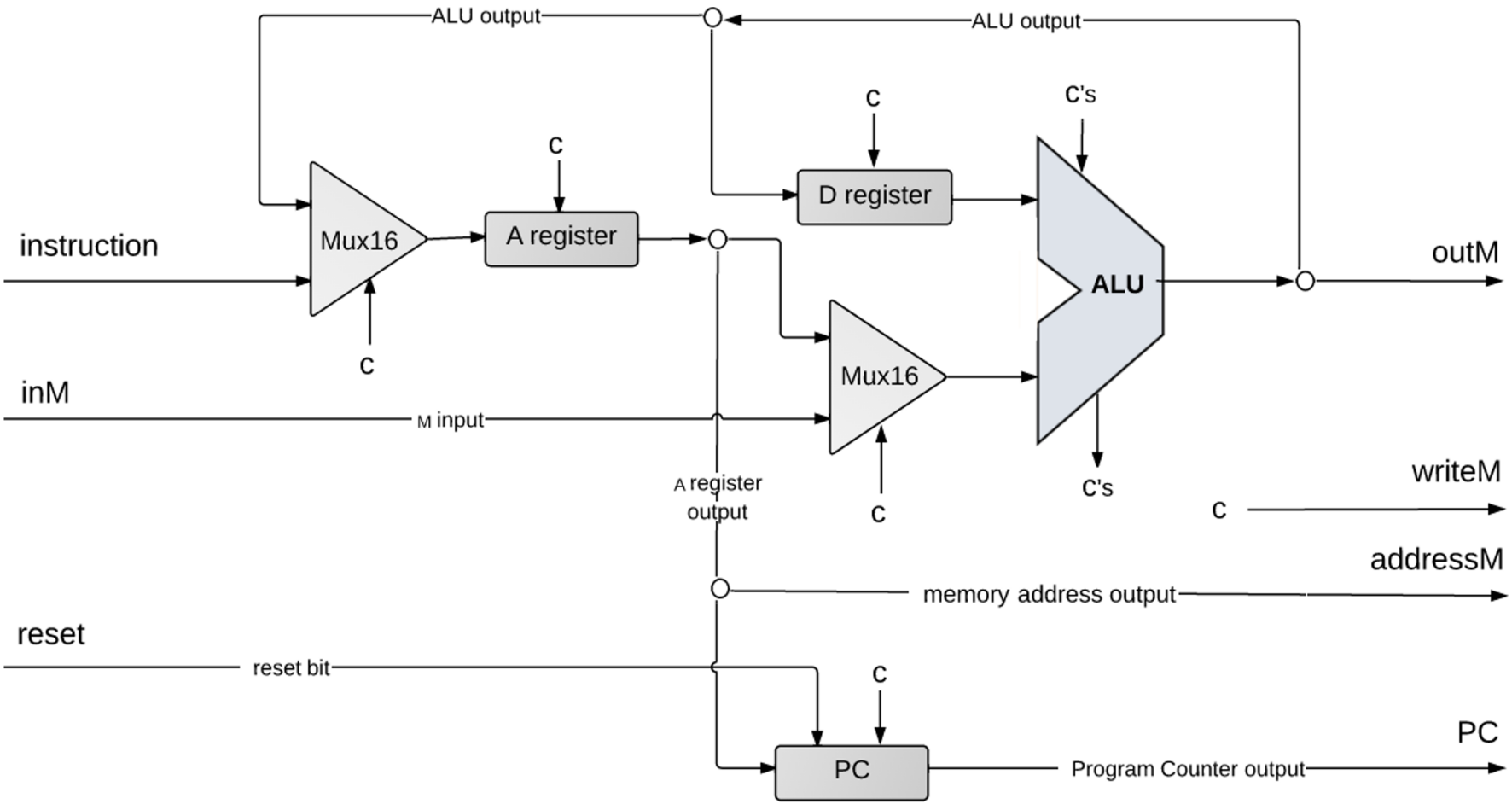
- **`outM`**: value used to update memory if writeM is 1
- **`writeM`**: if 1, update value in memory at addressM with outM
- **`addressM`**: address to read from or write to in memory
- **`pc`**: address of next instruction to be fetched from memory



Hack CPU Implementation



Hack CPU Implementation



(each "C" symbol represents a control bit)

Lecture Outline

- Building a Computer
  - Architecture, Fetch and Execute Cycle
- Hack CPU Interface
  - Implementation and Operations
- **CSE 390B Midterm Practice Problems**
  - **Circuit Design, Writing Assembly, Tracing Assembly**
- Project 6 Overview
  - Project Tips and Workflow

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?

CSE 390B Review Session

- Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly

- For each problem:
  - Step 1: Spend 10 minutes working on the problem individually
  - Step 2: Spend 10 minutes discussing the problem as a group, describing tips, approaches, and test-taking strategies
  - Step 3: As a group, present to the class your discussion from step 2 and lead the class in working through 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, 22wi, 22sp are more similar to what this quarter's midterm will look like
- 20sp midterm recommended to become familiar with problem types
- 21wi, 21sp, 22wi, and 22sp midterms recommended for practicing a timed exam
  - Set a timer for 60 minutes and take the exam in its entirety
  - Helps practice time management and simulate exam environment

Lecture Outline

- Building a Computer
  - Architecture, Fetch and Execute Cycle
- Hack CPU Interface
  - Implementation and Operations
- CSE 390B Midterm Practice Problems
  - Circuit Design, Writing Assembly, Tracing Assembly
- **Project 6 Overview**
  - **Project Tips and Workflow**

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

Post-Lecture 11 Reminders

- **Project 5 due tonight (11/3) at 11:59pm**
  - Remember to check that you have tagged the right commit
- CSE 390B midterm next Thursday (11/10) during lecture
- Project 6 (Mock Exam Problem & Building a Computer) released today, due in two Thursdays (11/17) at 11:59pm
- Course Staff Support
  - Eric has office hours in CSE2 153 today after lecture
  - Post your questions on the Ed discussion board