CSSE2010

# CSSE2010/CSSE7201
# Lecture 10A

# Memory and CPU Control Unit

School of Information Technology and Electrical Engineering
The University of Queensland

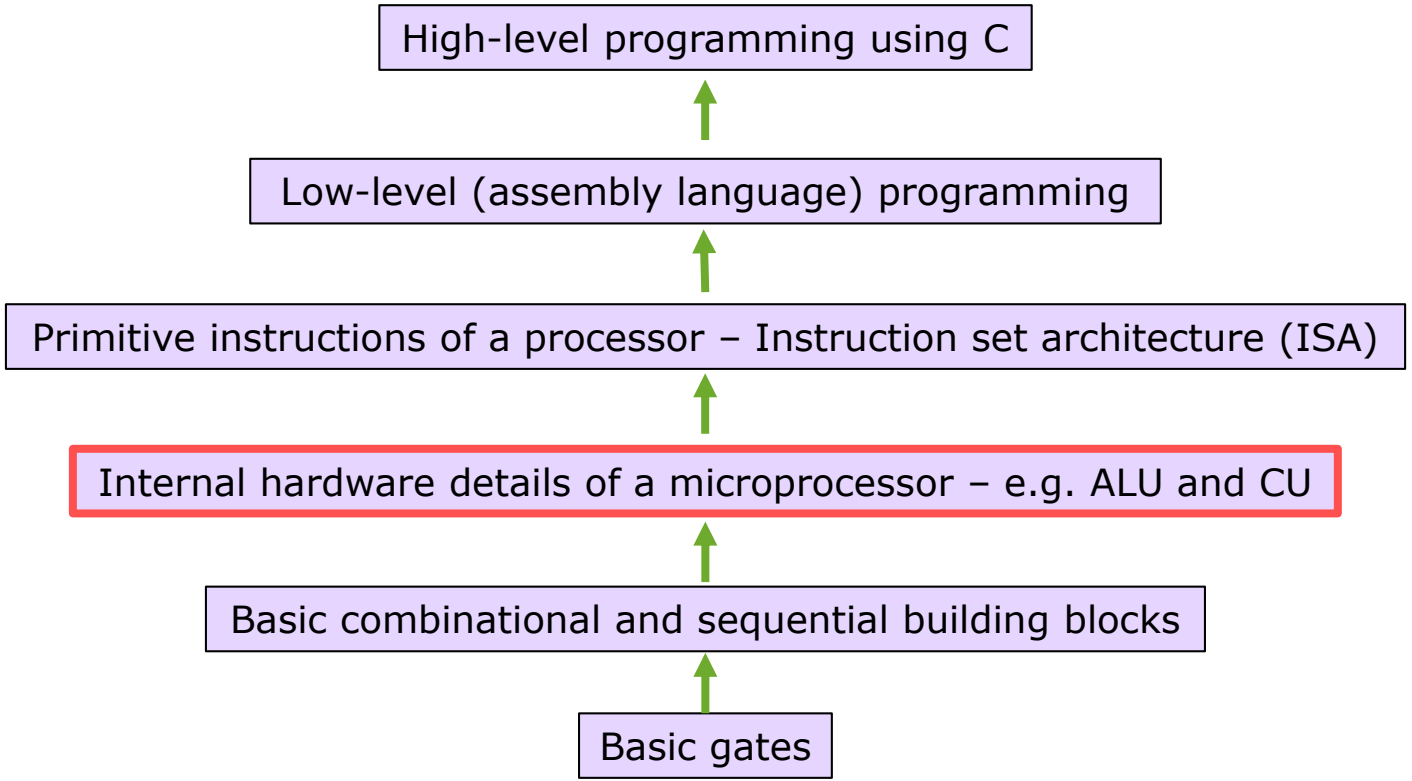
# Outline

- Admin
- Main Memory
- Fetch-Decode-Execute Cycle
- Controlling the CPU

# Admin

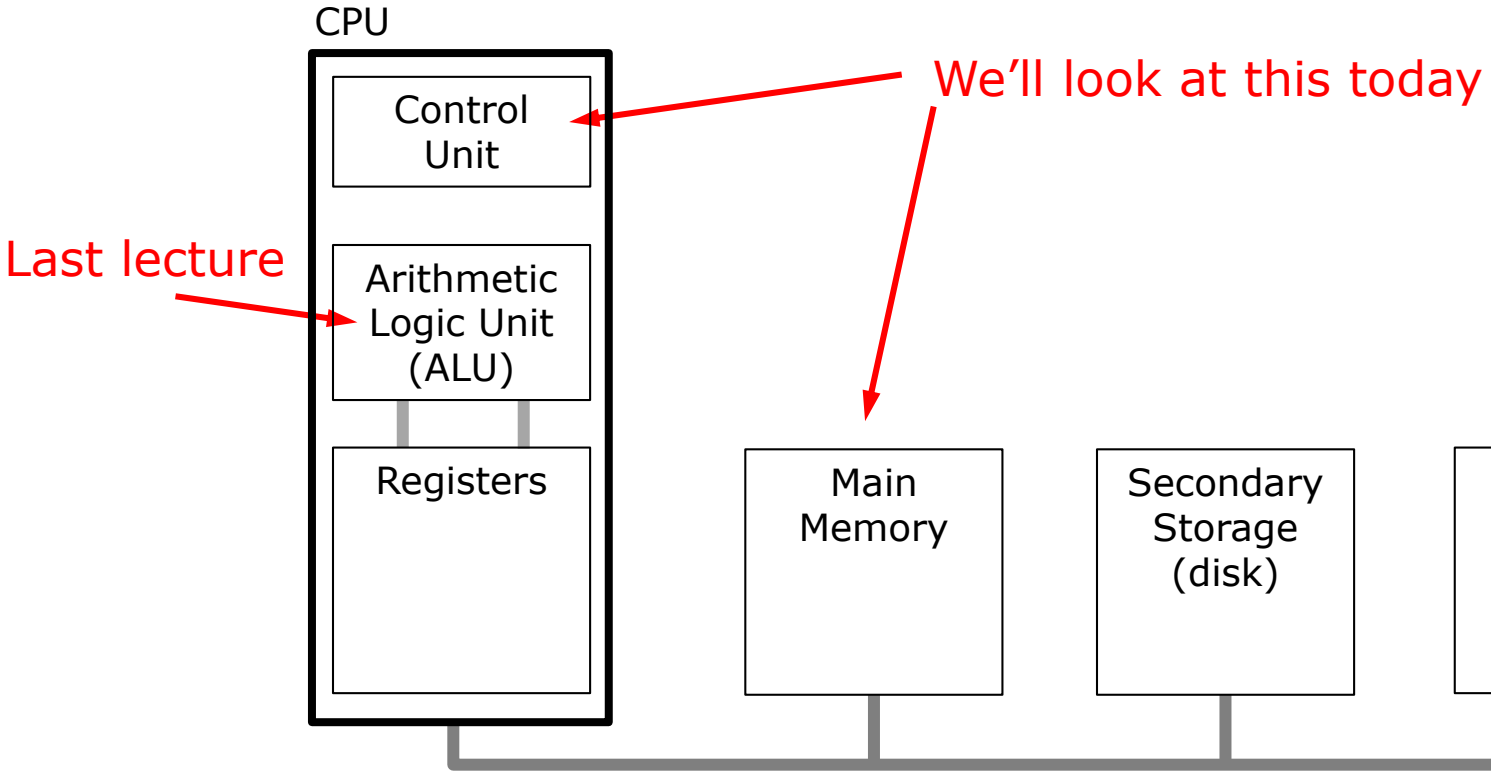
- Week 5 labs:
  - Tuesday – Lab 8 (state machines)
  - **Thursday/Friday – Catch up labs**
- Assignment 1:
  - Part 1: Timed 1-hour quiz, single attempt, 25 MCQs, must be completed before **Friday 3 Sep 4:00PM AEST**. Practice material: past mid-semester exams and practice test. Non-invigilated and open book.
  - Part 2: Design and Logisim simulation of a digital circuit. Submission due on **Monday 6 Sep 4:00 PM AEST.** Blackboard submission only.
- Week 6 – no new labs. You are supposed to work on assignment 1 during whole week 6. Get tutor support during lab sessions to clarify any doubts.
- Quiz 4 is due this week Friday 27 Aug 2021 4:00PM AEST
- No Quiz on week 6

# Where Are We Heading To?



# Computer Organisation

- **Simplified** view:



# Computer Memory

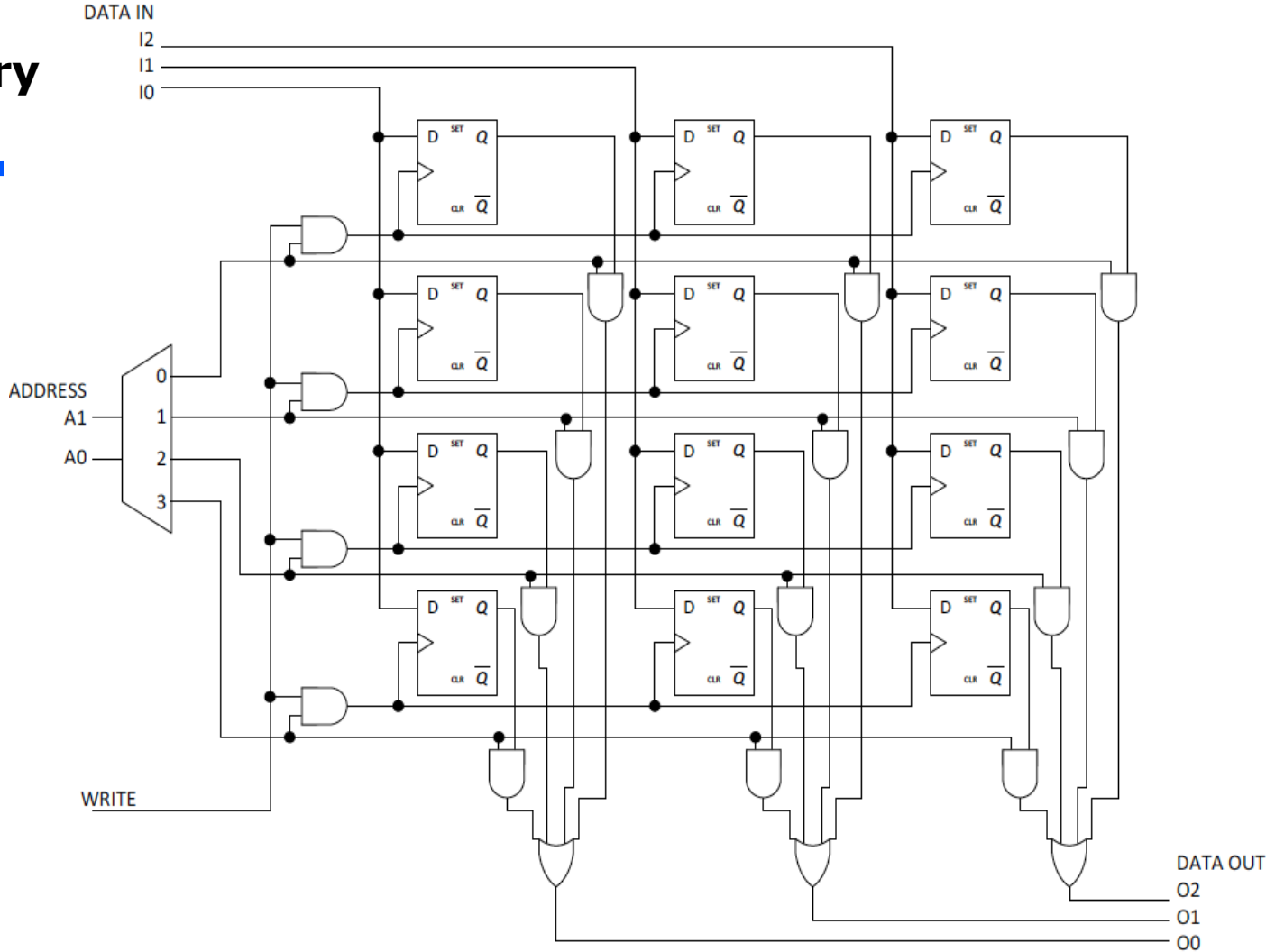
- Computers need memory to
  - Store temporary results
  - Store programs
  - Remember settings when power off
- Many different types
  - Volatile vs. non-volatile
  - Random access
  - Read-only
  - Static vs. dynamic
  - Primary vs. secondary
- We'll return to different types of memory in more detail in week 12

# Memory Organisation

- Computer memory is conceptually a set of addressable cells
  - Cells usually 8-bits wide (i.e. hold one byte)
  - Various cell widths have been used in past

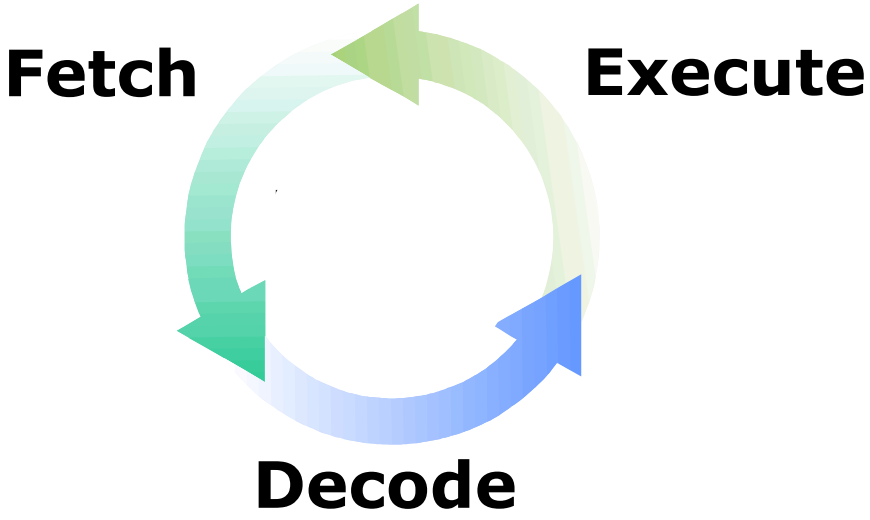

- Next slide shows 4 x 3 memory
  - 4 addresses, 3-bit cells

# 4 x 3 Memory


DATA IN
I2
I1
I0
ADDRESS
A1
A0
0
1
2
3
WRITE
D
SET
Q
CLR
DATA OUT
O2
O1
O0

# Fetch-Decode-Execute Cycle

- Control Unit operates in continuous cycle on the datapath:




**Fetch: fetch an instruction from memory and bring it inside the CPU. Two special purpose registers involved – PC and IR**

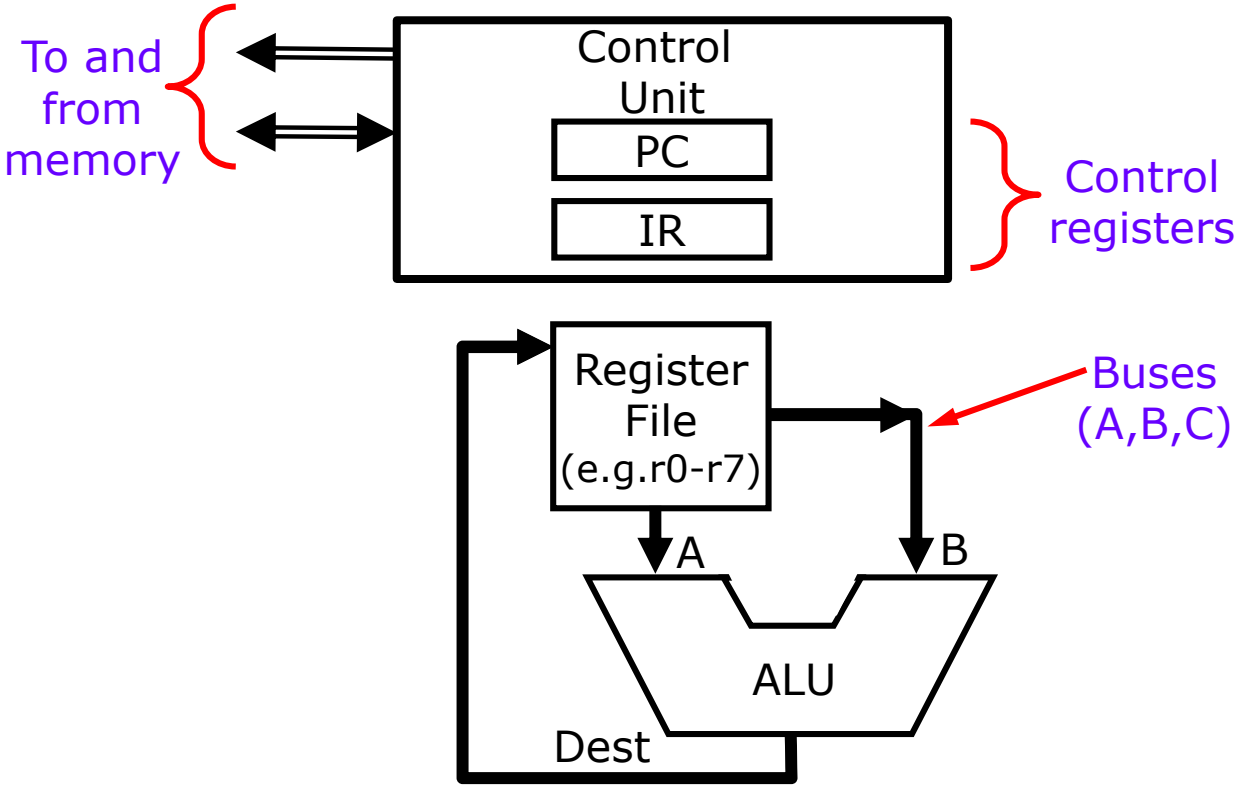
**Decode: resolve the operation (opcode) and operands (variables)**

**Execute: Control signals are issued and operation takes place in data path**

# Fetch-Decode-Execute Cycle (cont.)

1. Fetch next instruction from memory into Instruction Register (**IR**)
2. Change Program Counter (**PC**) to point to the next instruction
3. Determine type of (decode) instruction just fetched (i.e. look at OpCode)
4. Execute the instruction
5. Return to step 1

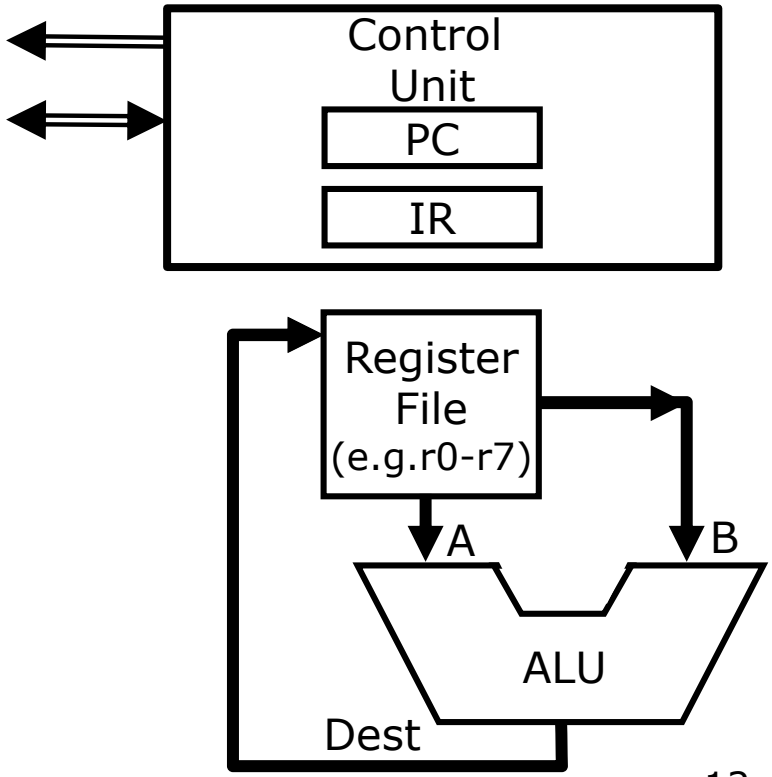
# Fetch-Decode-Execute: Data Path



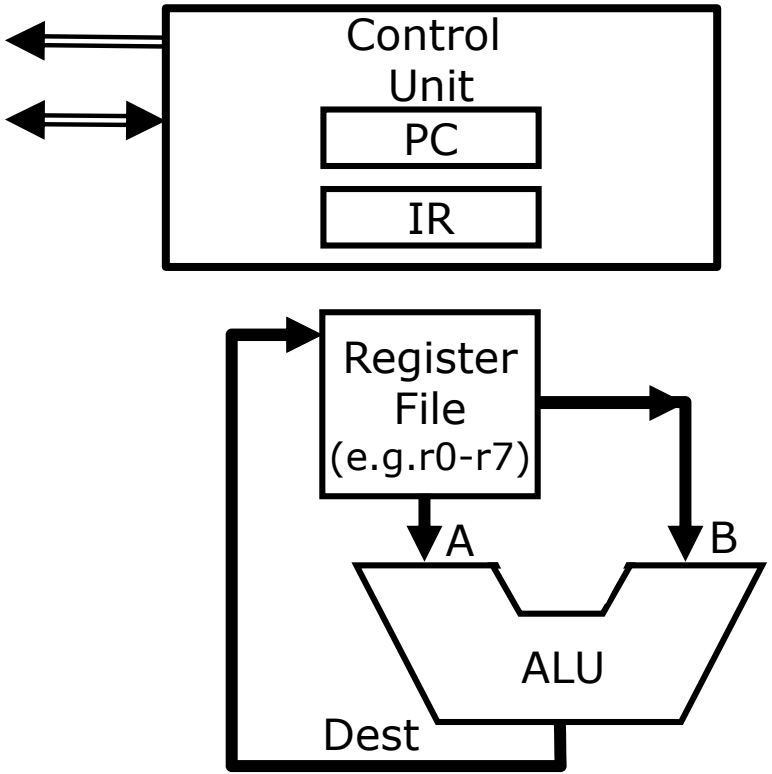
# How do we control this?

Specifically

- How do we tell the ALU what to do?
- How do we choose what register values are placed on the A and B buses?
- How do we choose which register receives the result from the Destination bus?
- How do we know when to read or write memory?



# Control Signals




*(Figure to be annotated in class.)*

# Control Unit

- Control signals come from the **control unit**
- Control unit must generate the control signals in the right order for a given instruction
  - Instruction determined by contents of the Instruction Register (IR)

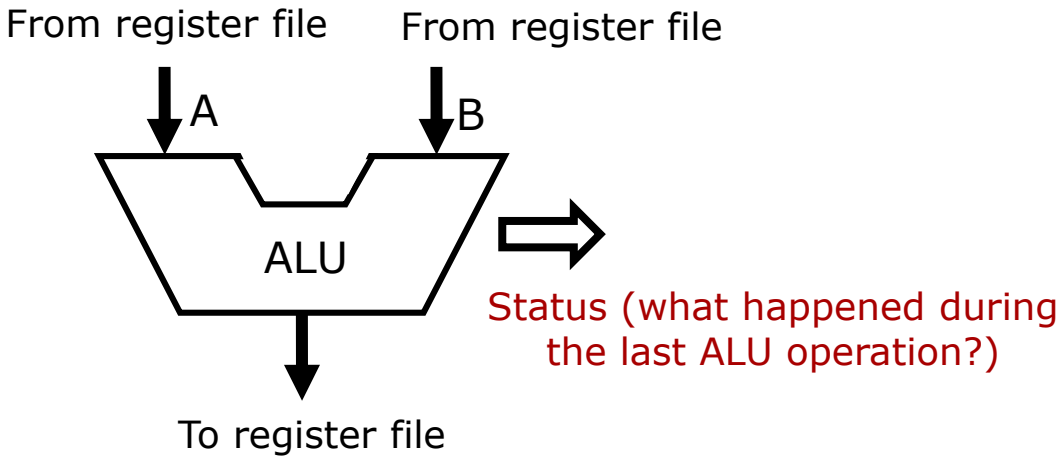
# Data influencing control

- Need to know more than what's in the instruction register
- Control unit can't operate without knowing something about the data
  - e.g. if a for-loop variable is supposed to go from 10 to 0, how does the control unit know when it gets there?

➔ **Status Register**

- Determined by ALU operations
- Stores status of last ALU operation

# ALU Status




Status (what happened during the last ALU operation?)

ALU status flags:

Z – Zero flag

V – Overflow flag

C – Carry flag

N – Negative flag

Typically stored in a special purpose register called STATUS register which is available to control unit.

*(Figure to be annotated in class.)*

# Status Register

- Typically includes
  - **Z** – zero bit – was the last result 0?
  - **V** – overflow bit – did the last addition/subtraction operation overflow (assuming it was a two's complement operation)?
  - **C** – carry bit – was a carry out generated?
  - **N** – negative bit – was the result negative if considered as two's complement (i.e. what's the sign bit)?

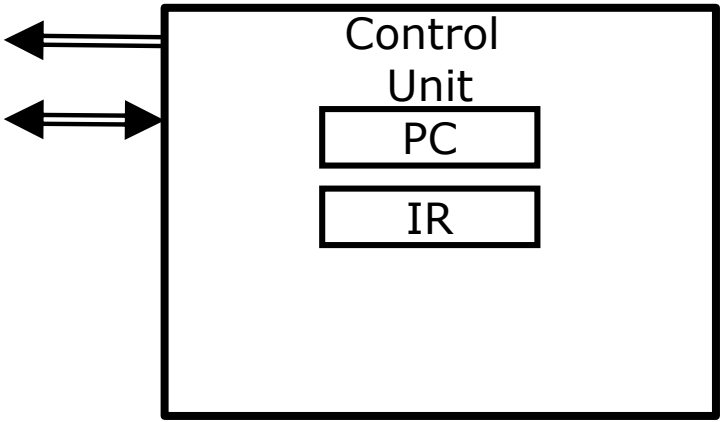
# Example

- What values are in the status register after operation $8_{10}$ - $9_{10}$ for a 5 bit ALU?

# What values are in the status register after operation 110 + 010 (binary addition) for a 3-bit ALU?

1. Z=0 V=0 **C=1** **N=1**
2. Z=0 **V=1** **C=1** N=0
3. Z=0 **V=1** C=0 **N=1**
4. **Z=1** V=0 **C=1** N=0
5. **Z=1** **V=1** C=0 N=0
6. **Z=1** V=0 C=0 N=0
7. **Z=1** **V=1** **C=1** N=0

# Control Unit




*(Figure to be annotated in class.)*