Computer Structure

Last updated 8/30/21

Structure

- Components
  - Instruction Memory
    - Long term storage of program
  - Data Memory
    - Long term storage of data
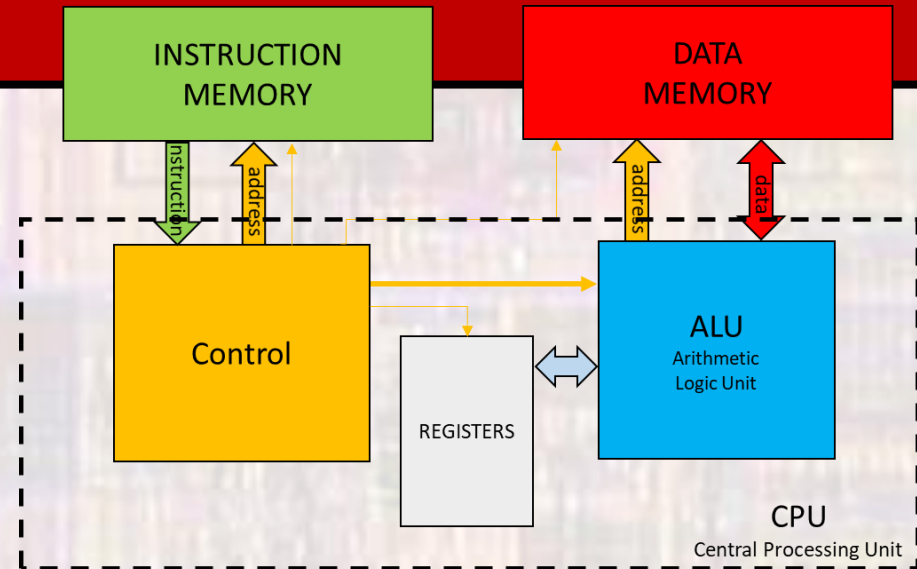 - Registers
    - Short term storage of data
  - ALU
    - Arithmetic Logic Unit
    - Performs operations on register data
  - Control
    - Manages the operation of the ALU
    - Manages the flow of data to/from memory and registers

Structure

- 1 line of code
  a = b + c;

  - what are a, b, and c?
  - where did b and c get their values?
  - how do we add them?
  - what happens to the result
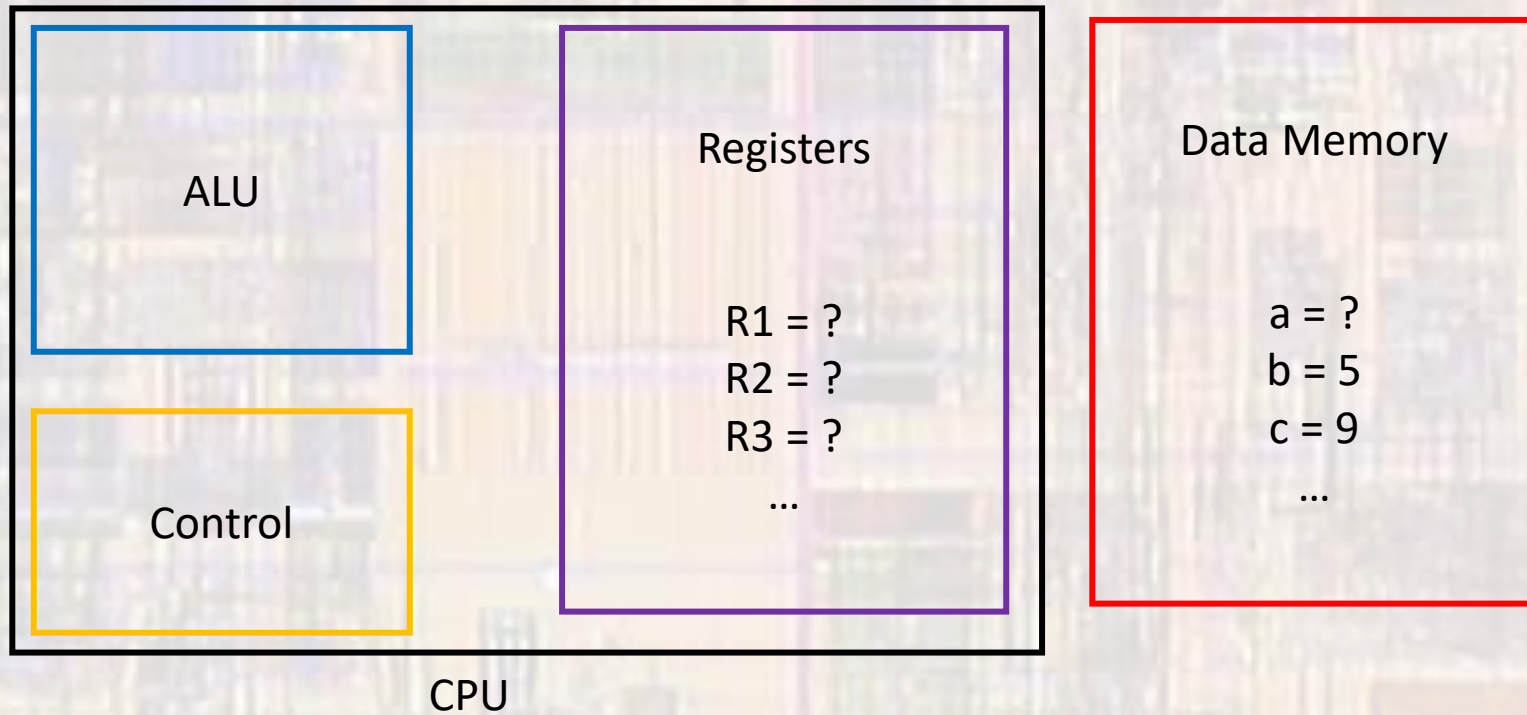
Structure

- 1 line of code
  a = b + c;

- what are a,b, and c?
  - values stored in memory or a register
    - what is a memory or register?
      - locations to store binary bits
- where did b and c get their values?
  - previous instructions
- how do we add them?
  - The CPU has an Arithmetic Logic Unit (ALU)
    - How does this work?
      - It grabs value from 2 registers and adds them
        - How did they get into the registers?
          - We load them from memory into the register (LOAD)
- what happens to the result
  - The result is written back to a register
    - Does it ever get stored back to a memory?
      - Yes – when we need to use the register for something else (STORE)

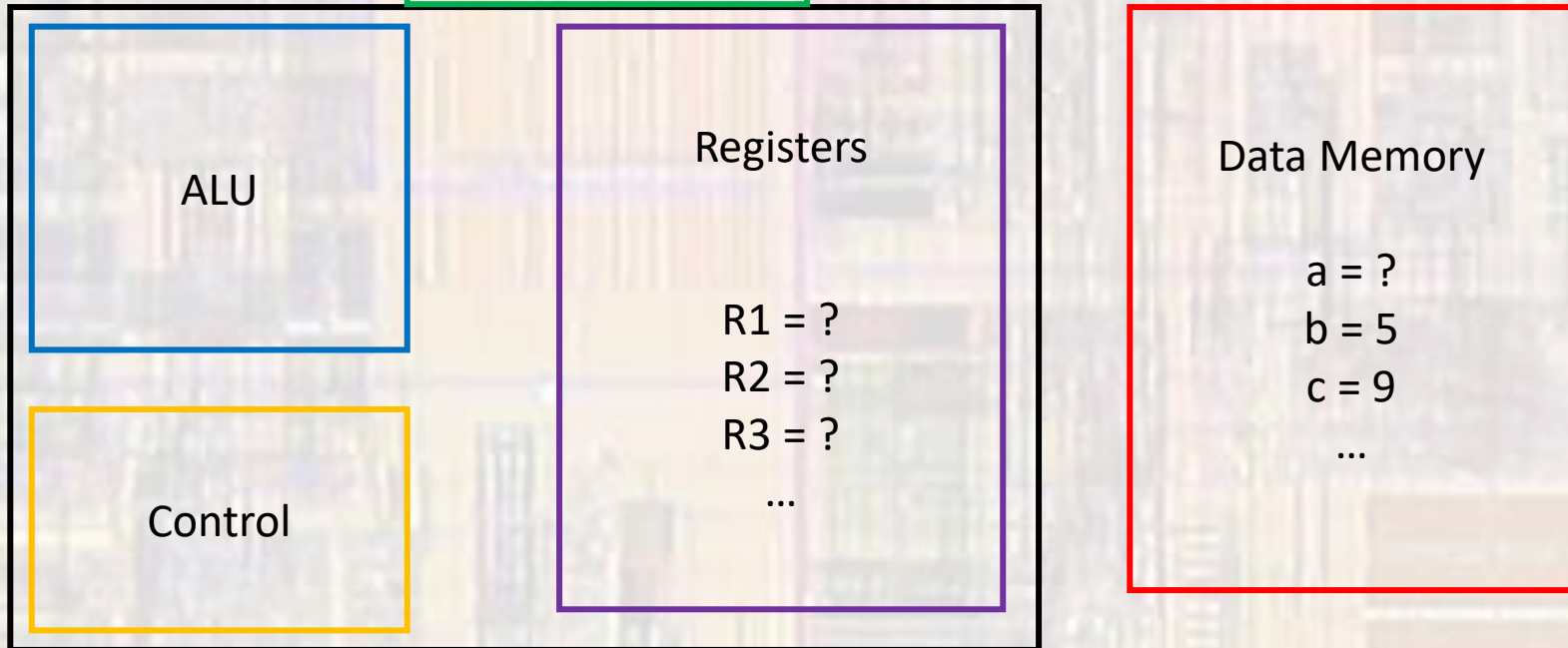# Structure

- 1 line of code – simplified version
  a = b + c;

CPU contains: ALU, Control, Registers

Registers
R1 = ?
R2 = ?
R3 = ?
…

Data Memory
a = ?
b = 5
c = 9
…

Structure

- 1 line of code – simplified version

a = b + c; → | ld R1, mem(b)   Instruction Memory
              ld R2, mem(c)
              add R3, R1, R2
              st mem(a), R3

ALU

Control

Registers

R1 = ?
R2 = ?
R3 = ?
...

Data Memory

a = ?
b = 5
c = 9
...

CPU

Structure

- 1 line of code – simplified version

a = b + c; → Instruction Memory

```
ld R1, mem(b)
ld R2, mem(c)
add R3, R1, R2
st mem(a), R3
```

first need to get the values into the registers (LOAD)

ALU

Control

Registers

R1 = 5
R2 = 9
R3 = ?
...

Data Memory

a = ?
b = 5
c = 9
...

CPU

Structure

- ## 1 line of code – simplified version

a = b + c; → | ld R1, mem(b)
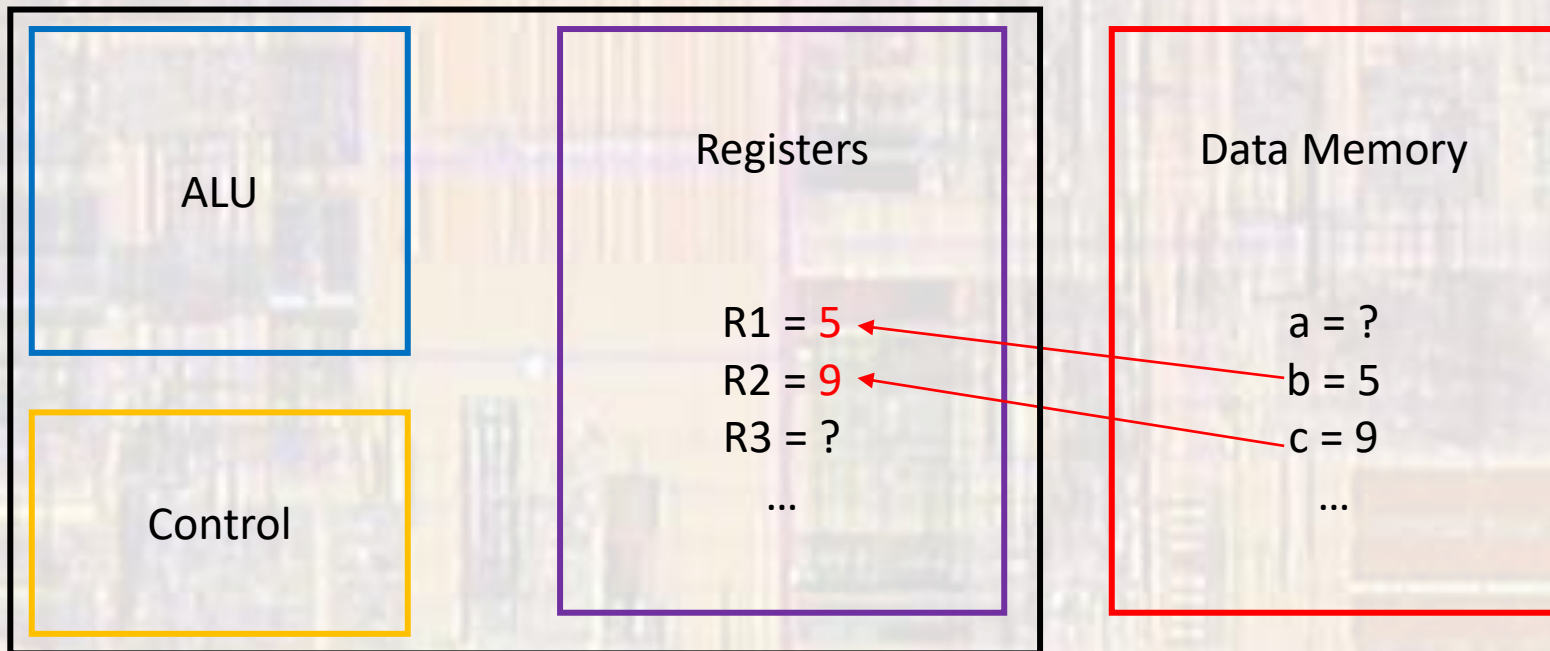ld R2, mem(c)
add R3, R1, R2
st mem(a), R3 | Instruction Memory

ALU adds the 2 values (execute)

| ALU

5 + 9 = 14

Control | Registers

R1 = 5
R2 = 9
R3 = ?

… | Data Memory

a = ?
b = 5
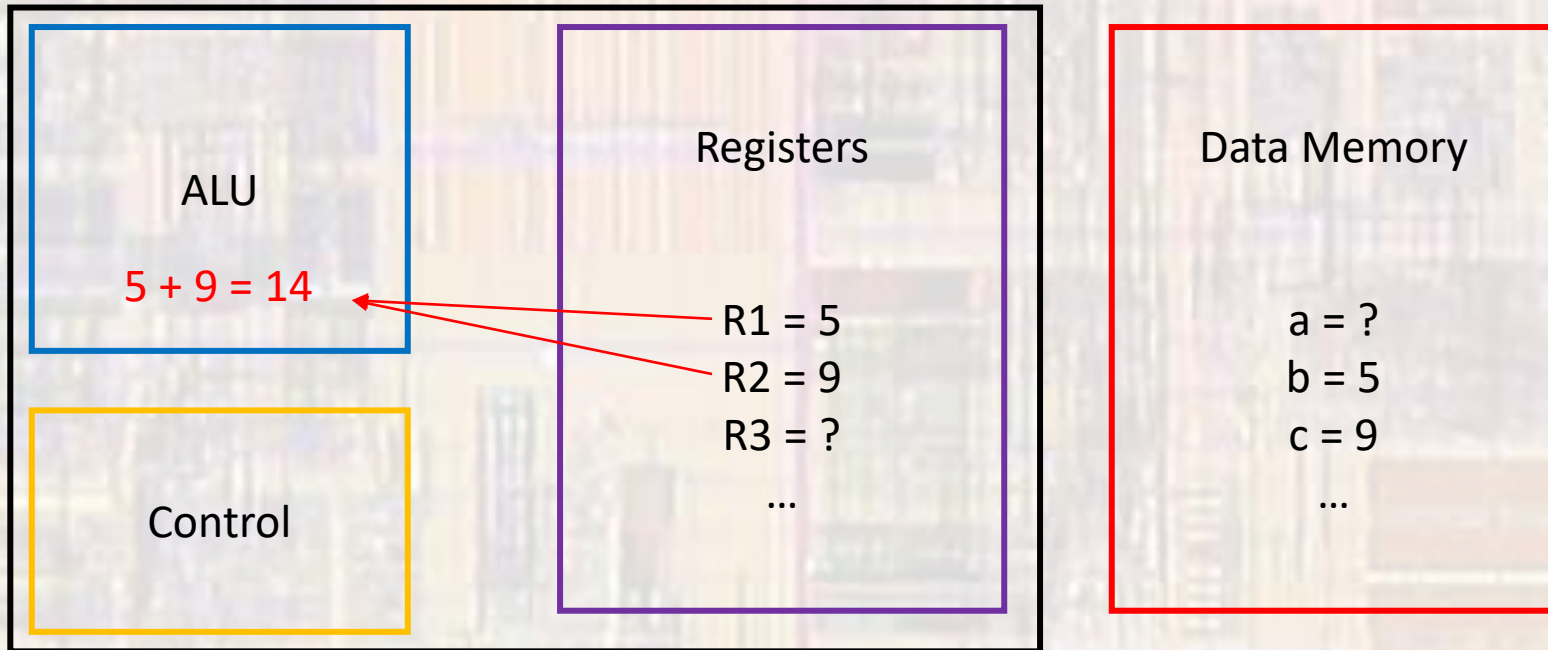c = 9
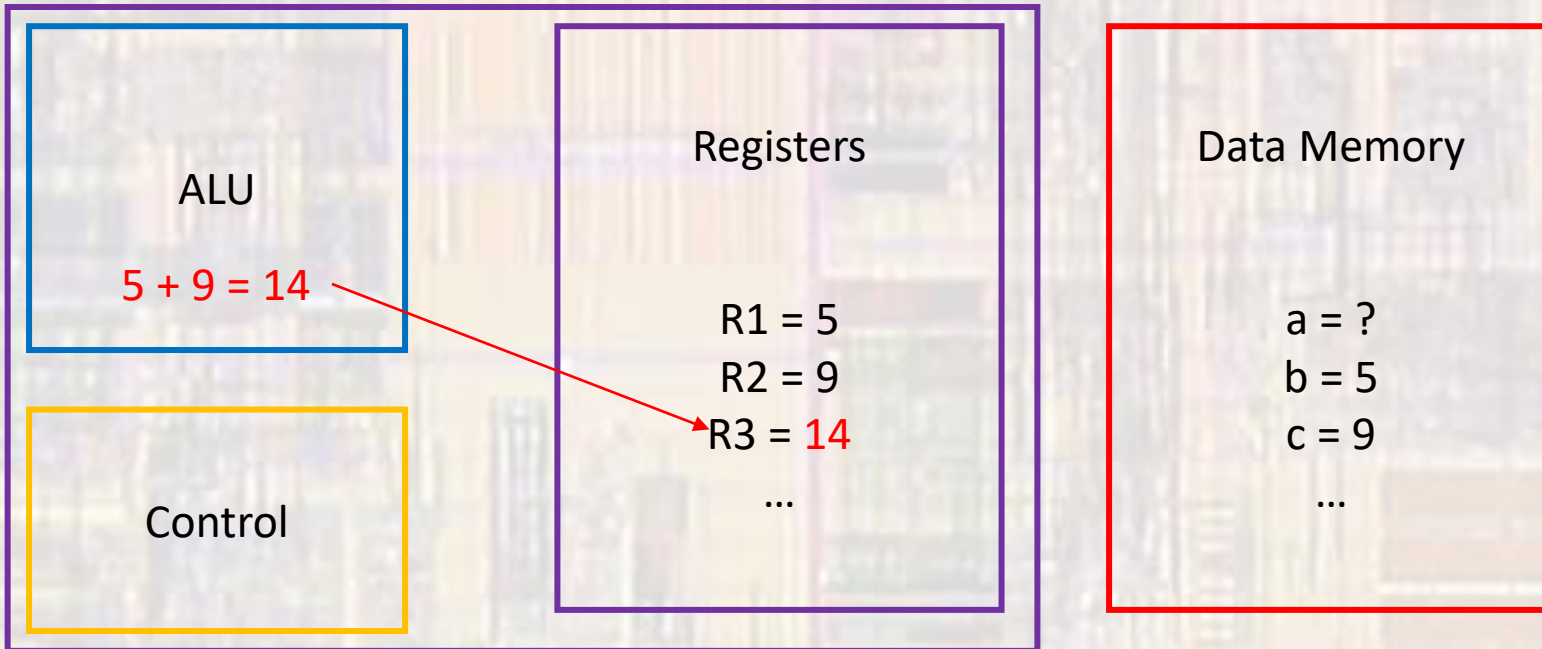
… |

Structure

- ## 1 line of code – simplified version

a = b + c; → | ld R1, mem(b) | Instruction
            | ld R2, mem(c) | Memory
            | add R3, R1, R2 |
            | st mem(a), R3 |

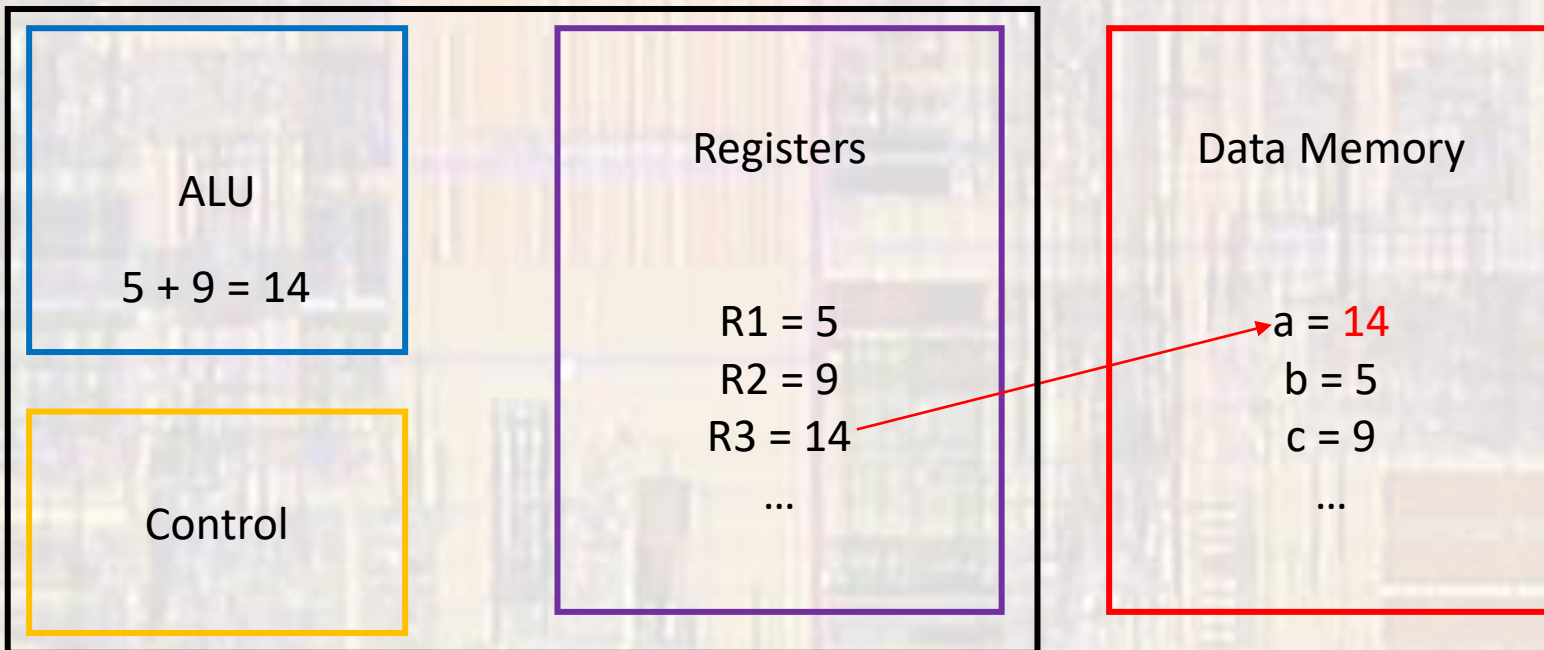ALU places the value back into a register (writeback)

ALU		Registers		Data Memory
5 + 9 = 14		R1 = 5		a = ?
		R2 = 9		b = 5
		R3 = 14		c = 9
Control		...		...

CPU

Structure

- ## 1 line of code – simplified version

a = b + c; →

Instruction Memory
ld R1, mem(b)
ld R2, mem(c)
add R3, R1, R2
st mem(a), R3

At some later time the value may be stored in memory (STORE)

ALU

5 + 9 = 14

Control

Registers

R1 = 5
R2 = 9
R3 = 14
…

Data Memory

a = 14
b = 5
c = 9
…

CPU

Structure

- Processor Structure