

# Assignment 5

Friday, October 27, 2023 1:20 PM

## CS219 Assignment #5

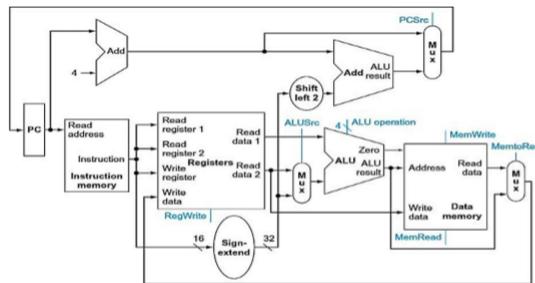
Purpose: Become familiar with the MIPS architecture instruction formats, control signals, Datapath elements.

Points: 110

**Reading/References:** Chapter 4, Class Lecture / Lecture Notes, Textbook

- 1) What are the steps to be followed to execute an instruction? [10 pts]
- 2) Define the following: [10 pts, 2.5 pts each]
  - a) Clock signal
  - b) Control signal
  - c) Data
  - d) How an edge-triggered methodology allows a state element to be read and written in the same clock cycle
- 3) Define combinational elements and give few examples [5 pts]
- 4) Define sequential (storage) elements and give few examples [5 pts]
- 5) Explain the Datapath elements that are needed by every instruction. Include the diagrams of each Datapath element and explain what is the role of that element to execute an instruction? (Note: You can copy and paste the diagrams from the textbook/PPT as needed in this assignment. No need of hand drawing) [20 pts]

[Hint: Refer the PPTs and/or textbook. Explaining (purpose, what it takes as input and what it gives as output, etc.) each element in the following diagram addresses Question 5]



- 6) Build a datapath that can support a R-format instruction, for example, a datapath to execute an *add rd, rs, rt* instruction. You no need to draw/copy each block individually, but just draw/copy the final datapath for a R-format instruction. List all the steps in executing the instruction, starting from PC. [Hint: Refer Ch4 PPT] [20 pts]
- 7) Build a datapath that can support an I-format instruction, for example, a datapath to execute a *lw rt, imm(rs)* instruction. You no need to draw/copy each block individually, but just draw/copy the final datapath for an I-format instruction. List all the steps in executing the instruction, starting from PC. [Hint: Refer Ch4 PPT] [20 pts]
- 8) Build a datapath that can support a J-format instruction, for example, a datapath to execute a *j targetAddress* instruction. You no need to draw/copy each block individually, but just draw/copy the final datapath for a J-format instruction. List all the steps in executing the instruction, starting from PC. [Hint: Refer Ch4 PPT] [20 pts]

1) The steps to execute an instruction are instruction fetch, instruction decode and register file read, execution or address calculations, data memory access, write back.

2) a) Clock Signal: signal produced by a clock mechanism in an electronic circuit that oscillates between a high state and a low state at a constant frequency.

b) Control Signal: inputs that determine MUX or logic block actions provided by the control unit.

c) Data: information bivaluer that are used in operations either arithmetic or logical

D) How an edge-triggered methodology allows a state element to be read and written in the same clock cycle:

By using the edge-triggered methodology, a clock cycle includes an edge up and an edge down. This allows for a read on the up and a write operation on the edge down.

3) Combinational elements are operational elements such as AND gates or an ALU. These elements mean their outputs depend on their outputs.

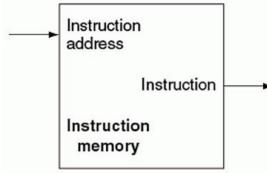
4) Logic components that contain state are also called sequential, because their outputs depend on both their inputs and their contents of the internal state. Some sequential elements are registers & a memory block.

5)

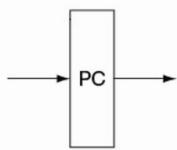


→ The Instruction memory is a state element that provides read-access to the instruction

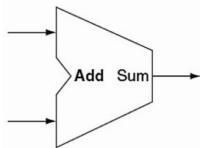
5)



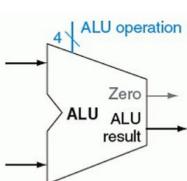
→ The Instruction memory is a state element that provides read-access to the instructions of a program and given an address as an input, supplies the corresponding instruction at that address.



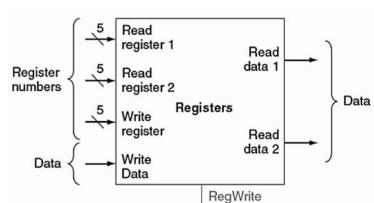
→ The PC is a state element that holds the address of the current instruction.



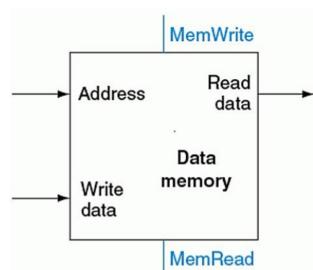
→ The adder is responsible for incrementing the PC to hold the counter of the next instruction.



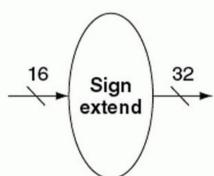
→ The ALU performs the operations indicated by the instruction (e.g. add, cr, and ... etc)



→ The register file is responsible for storing data into set registers to be read by other elements or to set bits in selected registers.

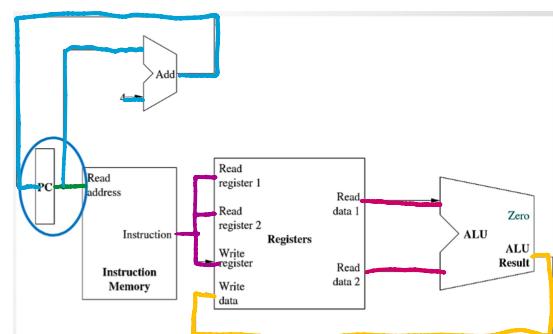


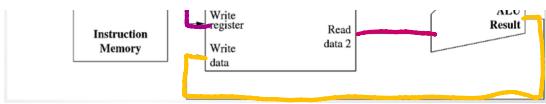
→ Data memory is a state block that both retrieves & writes data to long term storage.



→ sign-extend takes the sign of the msb & replicate it to fill the 32-bits of the rest of the address.

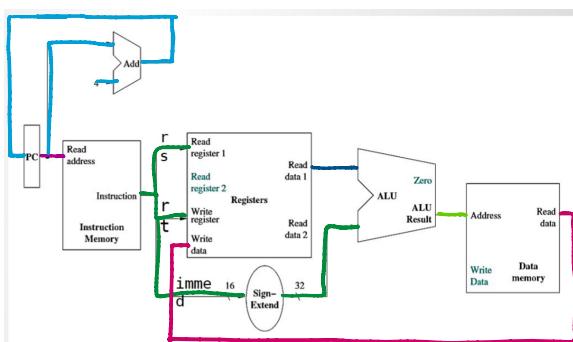
6)





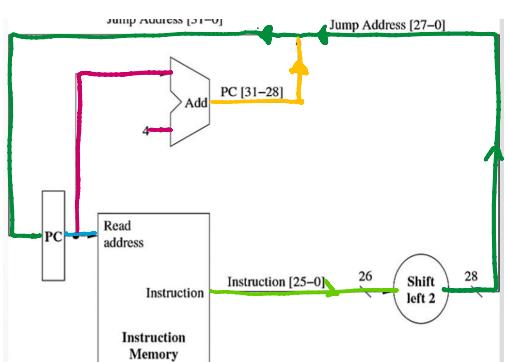
↑ this diagram shows all the elements needed to perform an R-formatted instruction. The PC element stores the current Instruction address, the instruction memory stores the contents of the instruction provided it to the register file which then provides data to the ALU who performs a logic operation on the data & the register file writes back the output back into the register. The PC then provides the address of the current instruction to the adder, who adds 4 to the address fetched the new address into the PC.

7)



→ At PC feeds the instruction memory the address of the current instruction. The IM feeds the instruction memory the address of the current instruction. The IM feeds the Register File data of the sign-extended elements. They feed the ALU data for the operation. The ALU reads the data memory on address to put the read data into the register for the Load Word instruction. The PC feeds the adder to increment the address, & the address back into the PC.

8)



→ from PC Reads the read address. The IM sends the instruction data to the sign-extend which shifts the jump address to left three. At the same time the PC feeds to the adder. The Adder increments the address of PC, & takes the sign-extended address for jump & sends it back to PC.