Name: _____ Section: ____

Name: _____ Section: ____

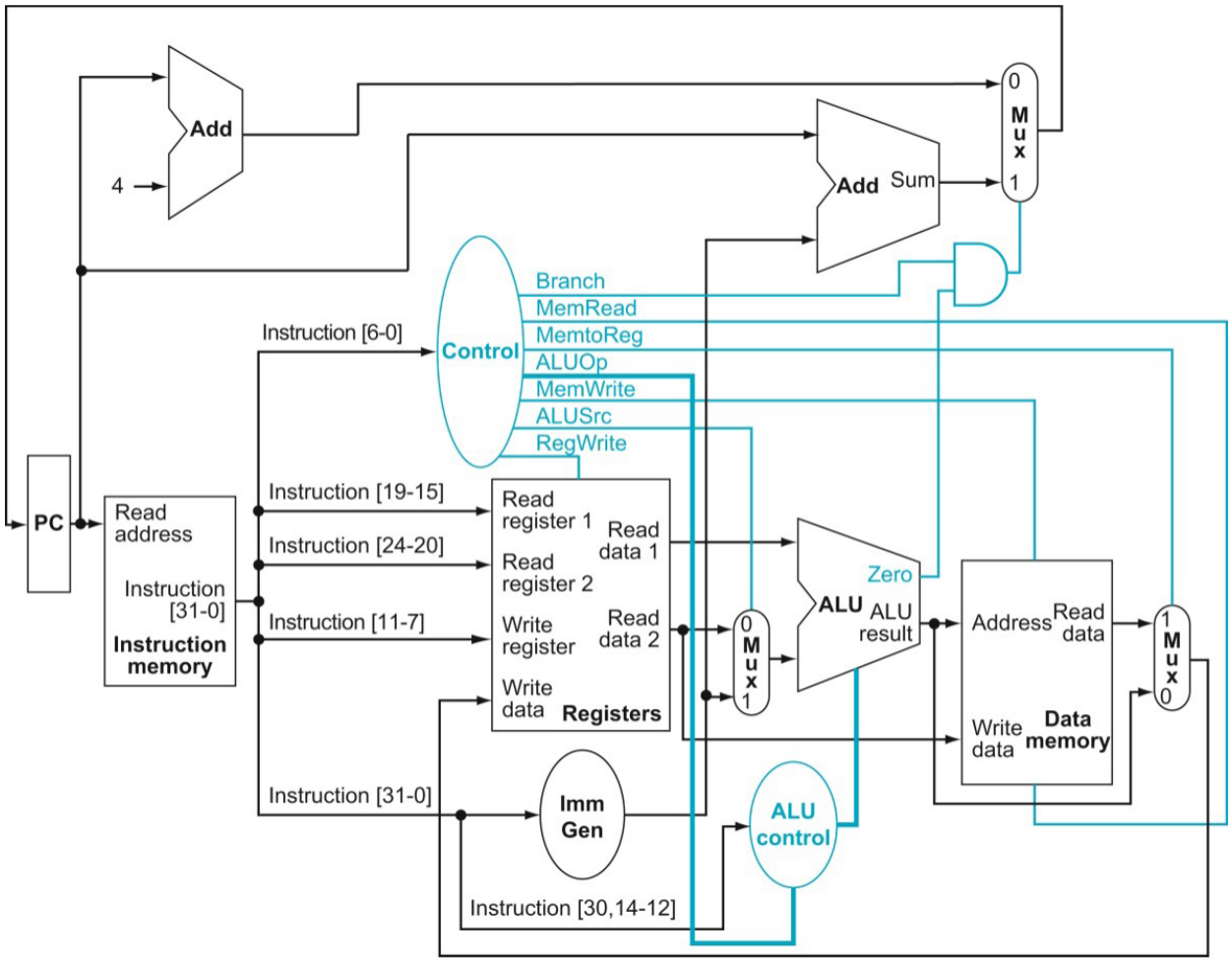
Name: _____ Section: ____

Practical 06

Read and perform the practical guide. Answer the questions after you have completed the practical. **Be sure to keep the formatting of 1-2 questions per page.**

[4] (Need) Write out the RTL for `lui`:

[4] (Need) Draw and highlight the datapath and control signals for `lui`:

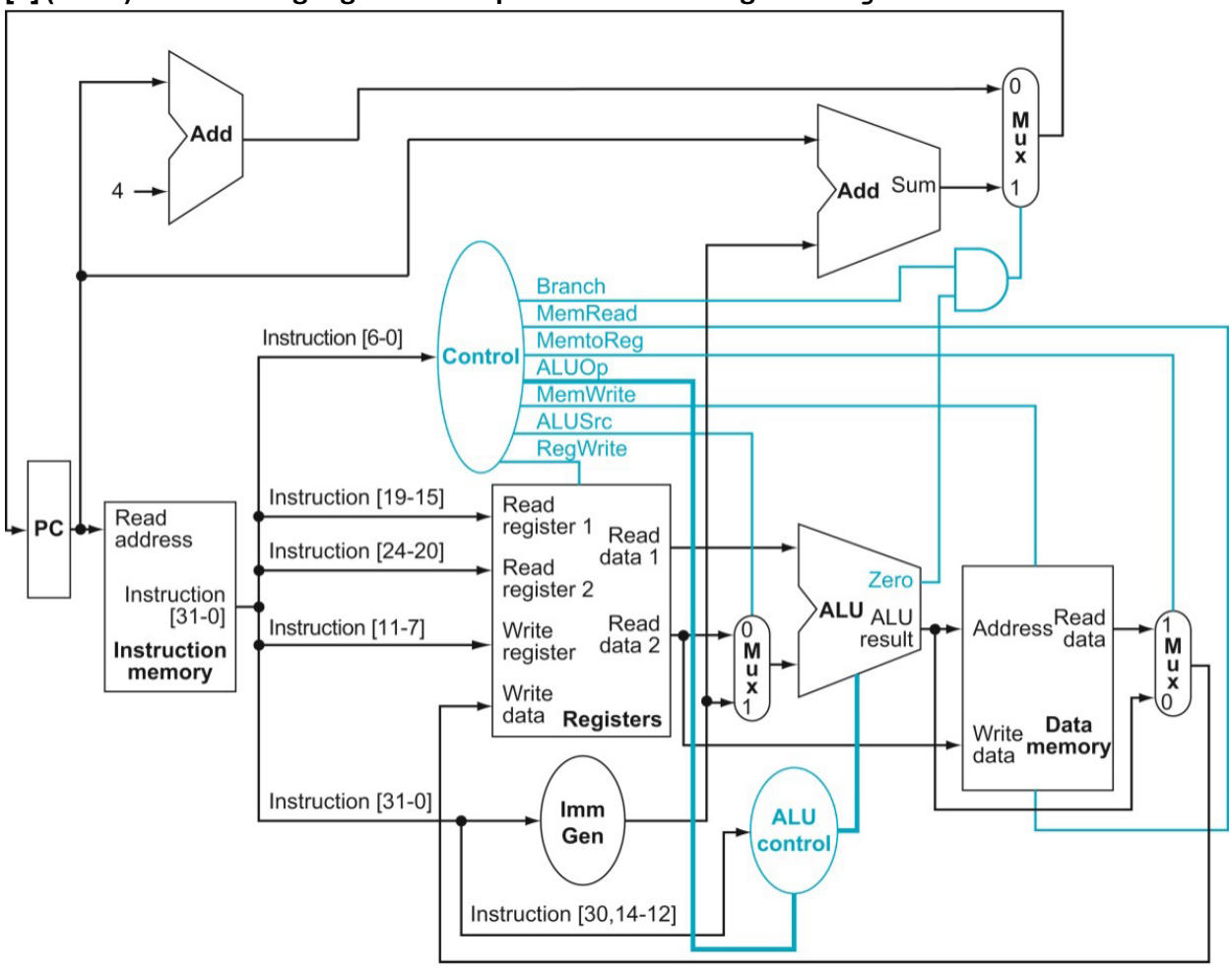


[4] (Correctness/Iteration) Take a screenshot of the ModelSim running `tb_processor` for `lui` tests. This should include the waveform with adequate signals grouped for organization as well as the transcript documenting the instructions you are running for the test.

Replace this textbox with your screenshot

[4] (Need) Write out the RTL for `jal`:

[4] (Need) Draw and highlight the datapath and control signals for `jal`:

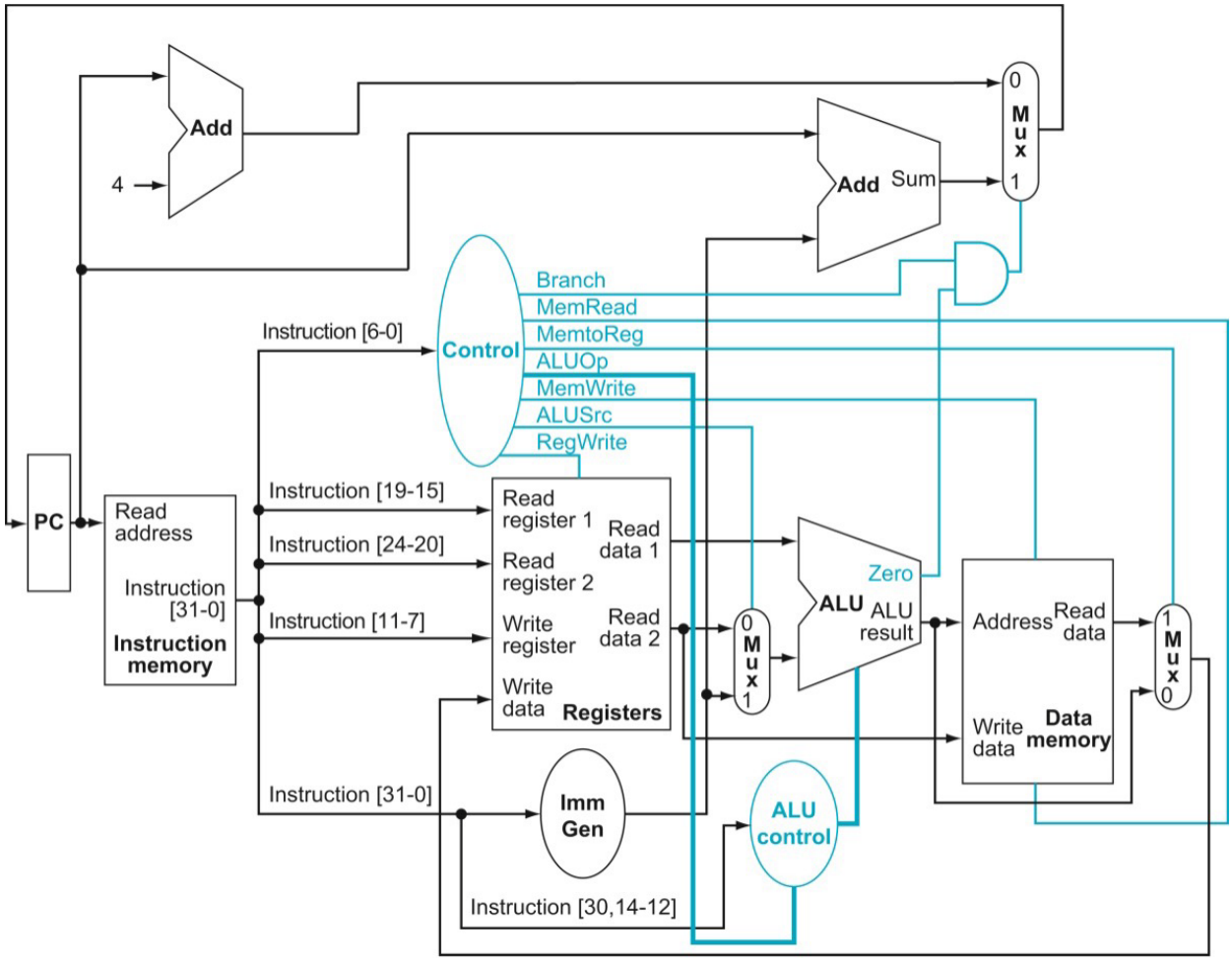


[4] (Correctness/Iteration) Take a screenshot of the ModelSim running `tb_processor` for `jal` tests. This should include the waveform with adequate signals grouped for organization as well as the transcript documenting the instructions you are running for the test.

Replace this textbox with your screenshot

[4] (Need) Write out the RTL for `jalr`:

[4] (Need) Draw and highlight the datapath and control signals for `jalr`:



[4] (Correctness/Iteration) Take a screenshot of the ModelSim running `tb_processor` for `jalr` tests. This should include the waveform with adequate signals grouped for organization as well as the transcript documenting the instructions you are running for the test.

Replace this textbox with your screenshot

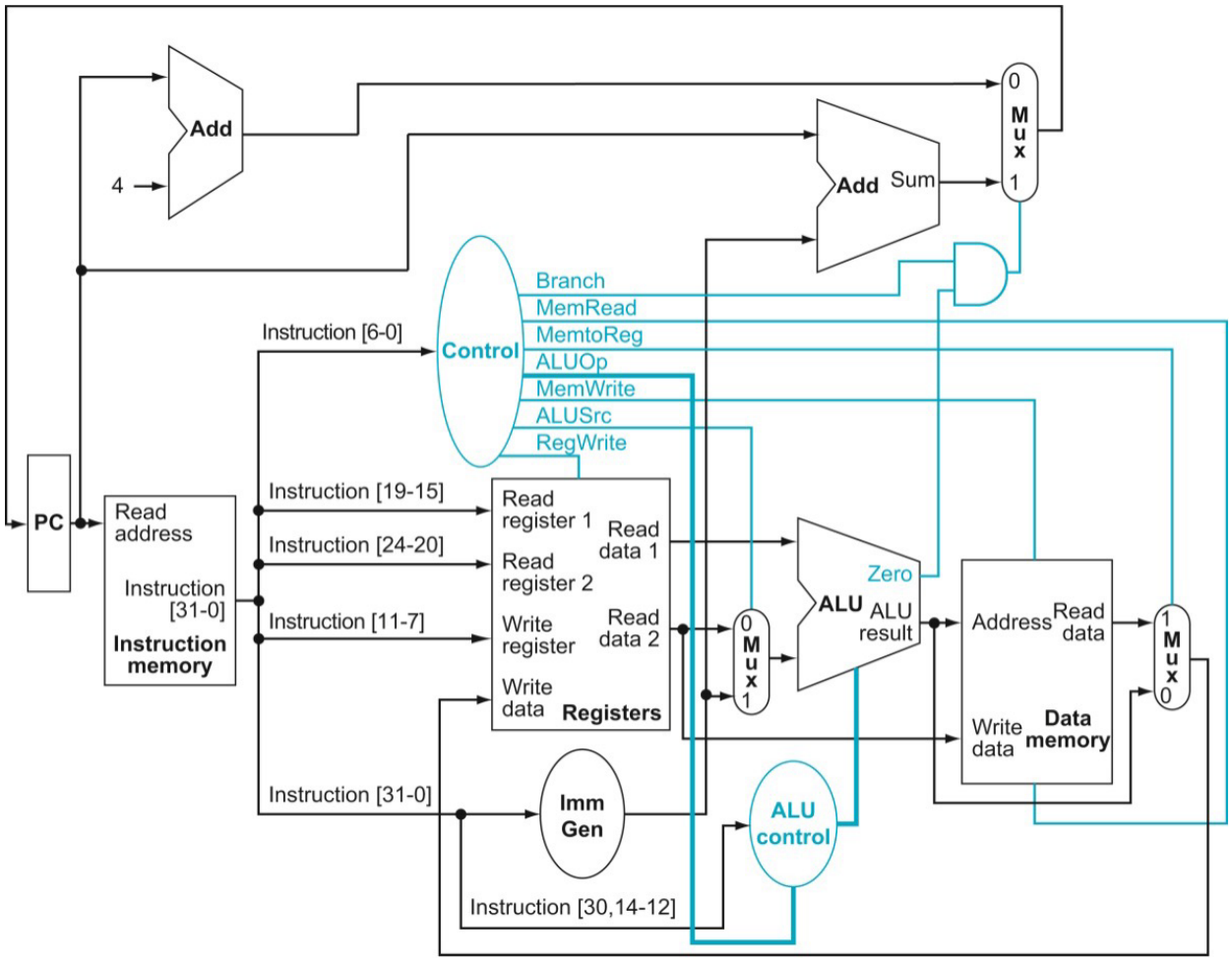
[8] (Correctness) How did you choose what instructions to include for your testbenches? What strategy/thinking process did you follow in order to ensure your set of testbench instructions sufficiently checks the validity of your processor (make sure to discuss this with specific reference to each instruction type implemented)? Did you change your testbench design strategy compared to the previous practical?

[4] (Need) Describe your new instruction. Make sure you include its name, command, syntax, and purpose.

[4] (Need) Draw out your new instructions' instruction format. Be sure to label each field of bits and explain what each field does (including how it is being addressed if applicable). If you need to create a new addressing mode, describe it here as well.

[4] (Need) Write out the RTL for your new instruction.

[4] (Need) Draw and highlight the datapath for your new instruction.



[4] (Correctness/Iteration) Take a screenshot of the ModelSim running `tb_processor` testing your new instruction. This should include the waveform with adequate signals grouped for organization as well as the transcript documenting the instructions you are running for the test.

Replace this textbox with your screenshot

[4] (Correctness) What instructions did you create for your instruction's testbench? Briefly explain why this set of instructions substantially checks for your processor's correctness for your new instruction.

[4] (Need) Explain how you addressed the timing problem with the link address for `jal` and `jalr`. How did you overcome the "PC gets changed to jump target before we save PC+4" dilemma?

[4] (Performance) Does the addition of your new instruction make your processor slower? Explain how you know it doesn't or why you think it does.

[4] (Correctness) When you added your new instruction, what steps did you take to ensure you did not break the previously implemented instructions?[8]

What was the biggest challenge in implementing and testing your new instruction? Explain in 100 words or less.

[10] What is the single biggest thing you learned from designing and implementing your new instruction? Explain in 100 words or less.

[0] What is the git commit ID for your final commit of your code. This is required to pass the assignment. Check Practical 1 for instructions on how to get the correct commit ID.