Name: _____ Section: ____

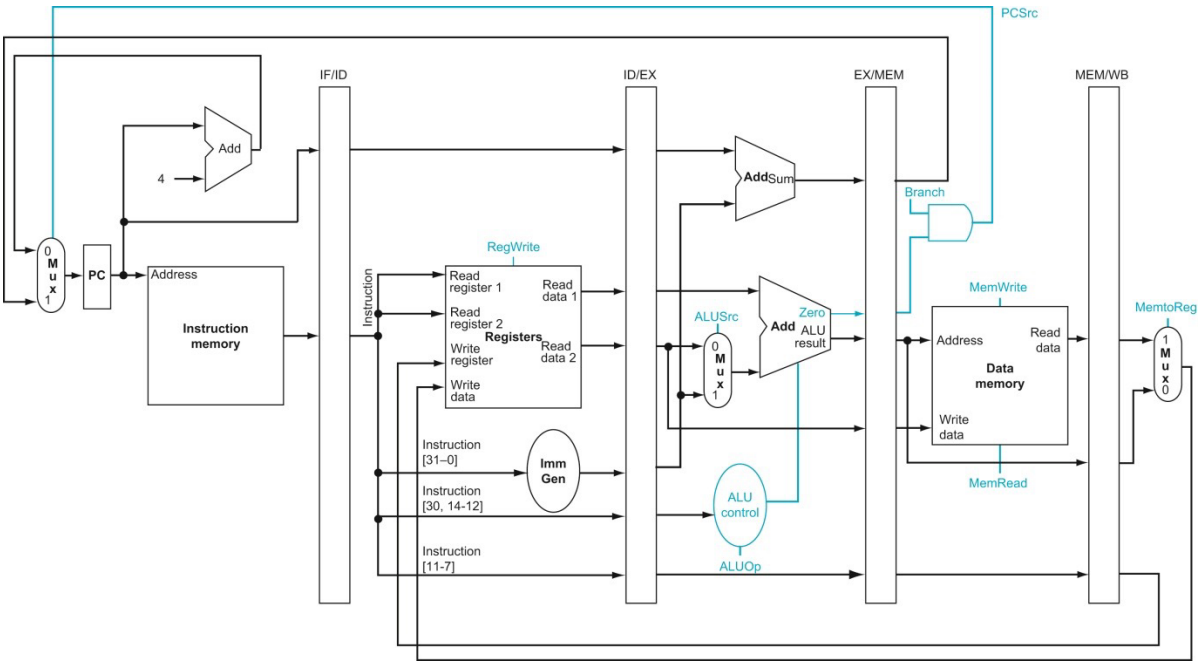
Name: _____ Section: ____

Name: _____ Section: ____

Practical 07

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.**

Here is the pipelined datapath for your reference:



[4] (Need) What values need to be stored in the registers in each buffer file to support R types? Be sure to include both data values and control signals. Fill out the table below. Fill this out before you implement `Processor.v`.

Buffer	Name	Bits	Use
IF/ID	**inst**	**32**	**To be decoded into control signals, register values, funct3/funct7.**
ID/EX			
EX/MEM			
MEM/WB			

[4] (Iteration) What registers, if any, were missing from the table above after you implemented R types in `Processor.v`? Why are these additions necessary?

[4] (Need) What <u>additional</u> registers are needed in each buffer file to support I types? Be sure to include both data values and control signals. Fill out the table below. Fill this out <u>before</u> you implement `Processor.v`.

Buffer	Name	Bits	Use
IF/ID	**N/A**	**N/A**	**N/A**
ID/EX			
EX/MEM			
MEM/WB			

[4] (Iteration) What registers, if any, were missing from the table above after you implemented I types in `Processor.v`? Why are these additions necessary?

[4] (Correctness) What instructions did you decide to test for the `test_I_type_nohaz` task in the testbench to substantially check the correctness for I types?

[4] (Correctness) In your `test_I_type_nohaz` task, did you check any opcode, func3, func7, ALUOp control signal values in each stage? Did you use the CHECK_PIPE_STAGES task provided to you in `pipeline_test_tools.vh`? Why or why not did you include this set of checks?

[4] (Correctness) Were there any differences in approach between writing a testbench for a pipelined processor compared to a single-cycle processor? What were the differences in approach, thinking process, and/or strategy?

[4] (Need) What <u>additional</u> values needed to be stored in the registers in each buffer file to support `lw` and `sw`? Be sure to include both data values and control signals. Fill out the table below. Fill this out <u>before</u> you implement `Processor.v`.

Buffer	Name	Bits	Use
IF/ID			
ID/EX			
EX/MEM			
MEM/WB			

[4] (Iteration) What registers, if any, were missing from the table above after you implemented `lw` and `sw` in `Processor.v`? Why are these additions necessary?

[4] (Correctness) What additional instructions would you add to the `test_mem_type_nohaz` task in the testbench to further build confidence on your processor's correctness? Why did you select these instructions to add? You do not have to add these into the testbench (you can if you'd like). If there are no additional instructions you foresee adding to the testbench, explain why the testbench provided to you already substantially tests mem types.

[4] (Need) What <u>additional</u> values needed to be stored in the registers in each buffer file to support U types? Be sure to include both data values and control signals. Fill out the table below. Fill this out <u>before</u> you implement `Processor.v`.

Buffer	Name	Bits	Use
IF/ID			
ID/EX			
EX/MEM			
MEM/WB			

[4] (Iteration) What registers, if any, were missing from the table above after you implemented U types in `Processor.v`? Why are these additions necessary?

[4] (Need/Performance) How does a pipelined architecture improve the performance of a processor? Should every processor be pipelined? Explain your reasoning.

[6] (Performance) Assume the cycle time of your single cycle processor is 8ns, assume the cycle time of your pipelined processor is 2ns. Assume there are no hazards in the code described below.

1. **How long would each processor take to execute a program that is only 1 instruction long? Report your answer both in number of cycles and number of seconds.**
2. **How long would each processor take to execute a program that is 100 instructions long? Report your answer both in number of cycles and number of seconds.**
3. **Explain in your own words why pipelined processors have a faster execution time for most programs.**

[4] (Iteration) Describe one error you uncovered during your implementation process and steps you took to resolve this error in your datapath.

[8] What was the biggest challenge in implementing and testing your base pipeline processor? Explain in 100 words or less.

[10] What is the single biggest thing you learned from designing and implementing a pipelined processor? 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.