Name: _____ Section: ____

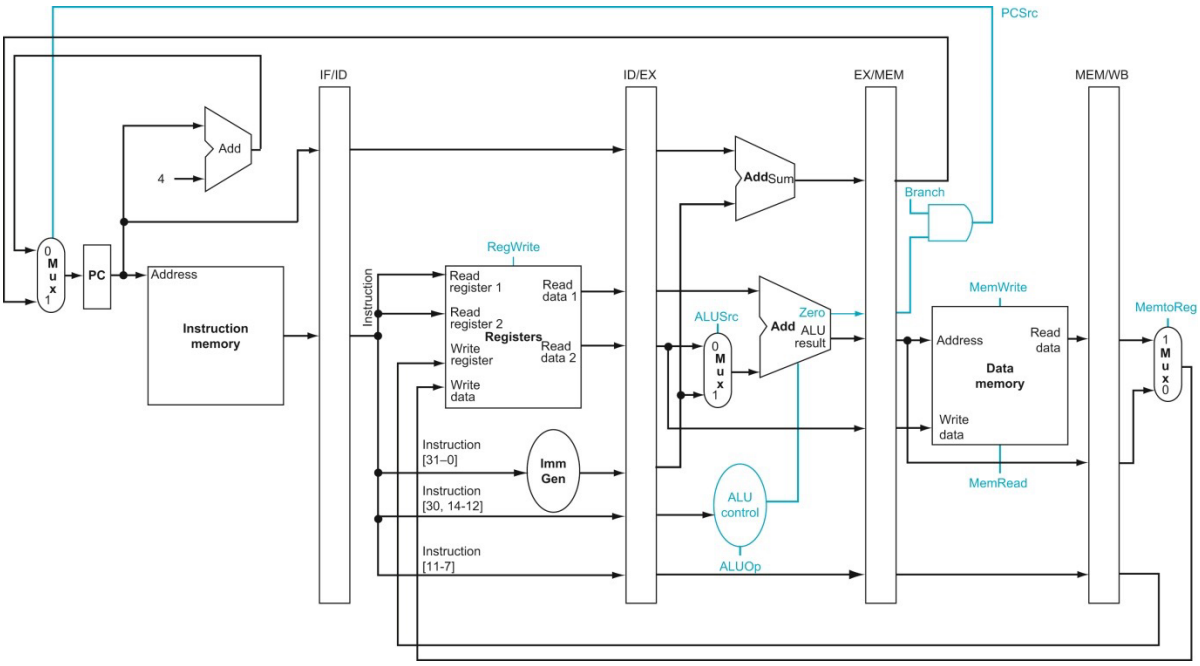
Name: _____ Section: ____

Name: _____ Section: ____

Practical 08

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 <u>additional</u> values need to be stored in the registers in each pipeline stage register (buffer) file to support branches? Fill out the table below. Fill this out <u>before</u> you implement `Processor.v`.

Buffer	Name	Bits	Use
IF/ID	**PC**	**32**	**To calculate `PC+imm` in the decode stage**
ID/EX			
EX/MEM			
MEM/WB			

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

[4] (Need) What <u>additional</u> values need to be stored in the registers in each pipeline stage register (buffer) file to support `jal` and `jalr`? Fill out the table below. Fill this out <u>before</u> you implement `Processor.v`.

Buffer	Name	Bits	Use
IF/ID	**PC**	**32**	**To calculate `PC+imm` in the decode stage**
ID/EX			
EX/MEM			
MEM/WB			

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

[4] (Correctness/Iteration) Take a screenshot of the ModelSim running `tb_Processor_Program` for the `test_gcd` task. 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/Iteration) What argument values did you call `gcd` with? Did you get the corrected expected result the first time you ran the test? If not, what did you have to change to get a correct `gcd` implementation?

[4] (Correctness/Iteration) Take a screenshot of the ModelSim running `tb_Processor_Program` for the `test_relPrime` task. 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/Iteration) What argument value did you call `relPrime` with? Did you get the corrected expected result the first time you ran the test? If not, what did you have to change to get a correct `relPrime` implementation?

[4] (Need) The testbenches you examined for branches and jumps are likely designed different than the testbenches you wrote for Practical 6 (single-cycle processor). Is it necessary to write the testbenches this way or were the testbenches you designed in Practical 6 sufficient? Explain why or why not.

[4] (Need/Iteration) Was it necessary to plan out the contents of each buffer (as you did in the worksheet) before implementing the pipeline in Verilog? Did this planning process help with your implementation process, or was it a hindrance? Explain your reasoning as to why it did or did not help.

[4] (Performance) How many cycles does it take for your processor to run `relPrime(360)` (divide the execution time reported in ModelSim by the length of one cycle)? This can definitely be improved. Discuss what are the current inefficiencies in your `relPrime` assembly program and this will be improved in the future.

[4] (Performance/Iteration) Examine your implementation for `relPrime` in HW10. Are there any pairs of consecutive instructions that can be combined as a single new instruction to improve the performance? Identify at least two pairs and briefly discuss how feasible it would be to implement this as a new instruction.

[4] (Iteration) Why do you think the practicals have had you implement one instruction type into your processor at a time? Did you ever break this plan by attempting to implement multiple instruction types at once? Discuss the advantages and disadvantages you experienced with your approach.

[8] What was the biggest challenge in implementing and testing branches and jumps? Explain in 100 words or less.

[10] What is the single biggest thing you learned from designing and implementing these instructions? 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.