In this assignment, you will complete the design a single-cycle implementation of a processor to execute your 8-bit ISA. At a minimum, your design will have a program counter (PC), a PC incrementor (if not in main ALU), a register file, an ALU and memory. The ALU and register file should not be significantly changed from the last lab. YOU WILL BE REQUIRED TO DEMO/TEST YOUR DESIGN FOR THE TA.

Some things to keep in mind for the lab:
- Use hierarchical design (subcircuits) to make your design easier to understand and think about. The highest-level schematic should mostly be functional elements (register file, ALU, memory, etc.) and wires/buses. The lowest levels should be in terms of basic Xilinx/Logic works components.
- Isolate control. Follow the text's lead by generating control signals in one place from the opcode. This should make the design easier conceptually, at least. For your single cycle processor, you only need to use combinational logic to set up the control lines.
- Keep in mind that you will have to debug your design! Think about how to make your life easier before it happens.

Memory and other new things to build:
- You will create a ROM to hold instruction memory and RAM(s) to hold data memory. Both will be initialized – the ROM to hold your program, the RAM(s) to hold your input data. Instructions to create memory parts in Xilinx/Logic works will be on the web page.
  - Only use one data memory RAM at a time, but you may want to have a few that you can “swap” in and out to try different input test cases—e.g. 255 found vs not found, etc).
- I want you to have an init signal that sets the PC to some 0, possibly initializes some registers to 0, etc. We also recommend that you have “done signal” that goes high at the halt instruction. This “done signal” should be shown on your waveform output.
- I also want you to have a cycle counter, which you will use to reliably determine dynamic instruction counts. It should be initialized to 0 at the start with your init signal (as part of the stuff that happens on the init signal).
- You should CHECK that your RAM has the correct output at the end of each test program (more details below). But you don’t need to provide any paper output.
You need to turn in the following items:

**Things to turn in electronically:**
1. Your Xilinx/Logicworks schematics files.

**Things to print and bring with you to the interview:**
1. Print out schematics of all circuits, hierarchically organized (for reference during the interview). The highest-level design needs to have all of the signals necessary to demonstrate correct program execution via the timing diagram.
2. The description of your ISA and your assembly language program, commented and documented (for reference during the interview).
3. Your written responses to the questions below (as with Lab 2, this will be collected at the end of the interview).

During the interview, the TA will be testing the correctness of your design, as well as asking a few questions about your design decisions. Design questions will be similar to the written questions, but more specific to your group's project.

**Written Questions:**

0. Brief general introduction and comments.
1. If you made any changes to your ISA and/or assembly language program -- what were they? Why did you make them? (we hope you didn’t, but if you did, this is the place to explain).
2. What could have been done differently to better optimize for dynamic instruction count? Give examples.
3. How successful were you at optimizing for ease of design and what was particularly difficult to design? For example, did you have any unpleasant surprises in which something about your Lab 1 ISA turned out to cause more difficulty in hardware than you had anticipated? Give 3 examples of success or “less-than-success” in this area. For each, what made it successful or what could you have done differently to better optimize for ease of design?
5. How easy/difficult would it be to extend your design to a multicycle implementation? A pipelined implementation? Be specific/give examples.
6. What might you have done differently if the priority was optimizing the processor area? Give examples.
7. If you are allowed to have a data memory with N read ports, how many read ports at most would you utilize to improve your dynamic count? How would that affect your cycle time?