CSE 141L Lab #3: 9-bit Graphics CPU
due Wednesday, February 27

In this assignment, you will finally design a single-cycle implementation of a processor to execute your 9-bit ISA. At a minimum, your design will have a program counter (PC), a PC incrementer, an ALU, memory, and probably some internal storage. The ALU and internal storage should not be significantly changed from the last lab. The CPU design will execute the programs you wrote (hopefully correctly) for Lab 1.

Some things to keep in mind for this 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.
• Isolate control. Follow the text's lead by generating all control signals in one place from the opcode. This should make the design easier conceptually, at least.
• You will create a ROM to hold instruction memory and a pair of RAMs to hold data memory and graphics memory. All will be initialized – the ROM to hold your program, the data RAM to hold your input data, and the graphics RAM to all zeroes. Instructions to create memory parts in Xilinx are on the web page.
• I want you to have an init signal that sets the PC to some predetermined value (probably zero). I also want a done signal that goes low at the init signal and high at the halt instruction.
• I want you to have a cycle counter that we will use to reliably determine dynamic instruction counts. It should initialize to 0 at the init signal, and count up until the done signal goes high.
• Keep in mind that you may have to debug your program! Think about how to make your life easier before it happens.
• In the questions for the lab report, I am no longer looking for you to convince me that your design is wonderful (unless it is), but rather I am looking at how effectively you critique your own design.
• Remember that it is your responsibility to convince us that your CPU works, not our responsibility to figure that out ourselves. Providing sufficient and clear results and information is crucial.
• We may have a mechanism for demonstrating your lab 3 to the TAs. If you do that, we still expect a full lab report, but you’ll be allowed to skip the screenshots and have a less complete timing diagram.

What you turn in:
I. A review of your ISA (if you turned in the executive summary, and it hasn’t changed, you can skip this).
II. Schematics of all circuits (including those presented in lab 2), hierarchically organized. The highest-level design needs to have all of the signals necessary to demonstrate correct program execution via the timing diagram.
III. A timing diagram for the program demonstrating correct operation and other important data. It should at a minimum show results being generated (e.g., particular graphics words read, then written, with the correct pixel values), the cycle counter, the PC. It need not show the whole
execution of the program, but certainly the beginning and end and some execution of the main loops. It, once again, should be heavily annotated so we can figure out what is going on.

IV. The assembly and machine code for your program.
V. A screenshot of the final picture, generated from the graphics memory dump, and a screenshot of the picture sometime in the middle of execution (instructions and tools on the web page).
VI. Answers to the following questions:
1. Have you made any changes to your ISA? What were they? Why did you make them?
2. What is your dynamic instruction count for the program?
3. What could you have done differently to better optimize for dynamic instruction count?
4. How successful were you at optimizing for ease of design? Give examples.
5. What could you have done differently to better optimize for ease of design?
6. How easy/difficult would it be to extend your design to a multicycle implementation? A pipelined implementation? Give examples.
7. What might you have done differently if a priority was ease of programming? Give examples.
8. What instruction takes longest on your machine (and thus would set the cycle time)? (Use rough estimates, e.g. assume each device introduces a constant delay).
9. What might you have done differently if a priority was short cycle time? (again, use rough estimates). Give examples.

The input

Data Memory location values (word values starting at address 0, all in decimal):

1, 33, 59, 28, 63,
1, 28, 63, 18, 49,
1, 18, 49, 24, 43,
1, 24, 43, 35, 52,
1, 36, 52, 32, 52,
4, 48, 64, 65, 43,
2, 80, 55, 11,
1, 98, 44, 115, 58,
1, 115, 58, 98, 66
1, 98, 66, 98, 44
3, 118, 50, 124, 72,
2, 121, 42, 6,
1, 84, 141, 80, 148,
2, 50, 130, 25,
4, 30, 133, 8, 118,
0