Lab Assignment 1

ECE/CS 3700
Fall 2024
Assigned during the week of August 28, You are expected to demo your functioning design during your respective lab sessions during the week September 4. Final report due: September 13

The objective of this lab assignment is to learn the techniques of "Schematic Entry", "Verilog Design", "Simulation" and "Synthesis" in the context of digital logic design. Moreover, this lab will introduce to you an elementary 2-bit computer. Your job is to: i) design the logic circuit using a CAD tool called the "Schematic Editor"; ii) "Simulate" your design using a Verilog testbench; iii) Re-design the same circuit using Verilog description and simulate using the *same* test bench as before; and finally, iv) "Synthesize" the netlist and download the design onto the FPGA and demonstrate its correct operation. This way, you will learn the top-down design flow and prototyping using FPGAs.

Note: On the class website, I'm also uploading a few tutorials and manuals on each of the above exercises. These tutorials are created so that you can use the step-by-step procedures to install the CAD tools on your own computers, get acquainted with the CAD tools and associated resources. My suggestions would be to first recreate the (very simple) tutorial to see that you can invoke these tools readily from your accounts and then proceed to solve the problem given below. Hope you'll have fun with this lab.

I. The Two-Bit Computer

You are asked to design a circuit depicted in the black-box (block) diagram shown in Fig. 1. The circuit takes three (word-level) inputs $A[1:0]$,$B[1:0]$,$I[1:0]$; where A,B,I are each 2-bit vectors. In other words, $A[1:0] = \{a_1,a_0\}$, is a vector of two binary variables[1]. $A[1:0]$,$B[1:0]$ are the "data" inputs of the design and $I[1:0]$ corresponds to the "control" or the "instruction" input. The output is also a 2-bit vector, $F[1:0] = \{f_1,f_0\}$ – i.e. the circuit produces two binary Boolean outputs f_1 and f_0.

Circuit functionality: The values of the input control signals ($I[1:0]$) decide the operation of this circuit. The instructions are:

- When $I[1:0] = (i_1,i_0) = (0,0)$, the output $F[1:0]$ is a bit-wise AND of the inputs, i.e. $f_1 = a_1 \cdot b_1$; $f_0 = a_0 \cdot b_0$.
- When $I[1:0] = (i_1,i_0) = (0,1)$, the output $F[1:0]$ is a bit-wise OR of the inputs, i.e. $f_1 = a_1 + b_1$; $f_0 = a_0 + b_0$.
- When $I[1:0] = (i_1,i_0) = (1,0)$, the output $F[1:0]$ performs an *equality check* of the corresponding bits; i.e. $f_0 = 1$ if $a_0 = b_0$, otherwise $f_0 = 0$. Similarly, $f_1 = 1$ when $a_1 = b_1$, otherwise $f_1 = 0$.
- When $I[1:0] = (i_1,i_0) = (1,1)$, the output $F[1:0]$ perform the complement of $A[1:0]$; i.e. $f_0 = \overline{a_0}$ and $f_1 = \overline{a_1}$.

[1] For all practical purposes, your design has 6 binary variables as inputs: a_1,a_0,b_1,b_0,i_1,i_0. It is, however, customary to represent data as a group or a vector of bits, giving rise to a notation involving bit-vectors.

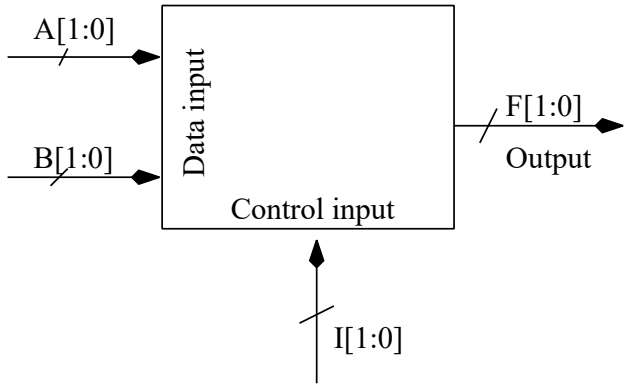

Fig. 1. Block Diagram of a two-bit Computer

II. The Assignment

You are asked to design a circuit that implements the above function. In other words, you have to derive Boolean functions for $f_1 f_0$, in terms of $a_i b_i i_k l = 0,1$. You are free to use any technique that you wish for circuit optimization. Spend a few minutes to think about the design, perhaps you may want to draw a circuit diagram on paper....

Once your design is ready, do the following:

- Schematic entry: Using the Quartus Schematic tool, you will create a new Block Diagram/Schematic File and draw the schematic corresponding to your design, using AND, OR, NOT, XOR, etc. gates. Do not use pre-designed multiplexors (MUXs). If your design is valid then you can generate a Verilog file from it by navigating to File > Create / Update > Create HDL Design File from current file.
- Structural Verilog: Once your schematic entry is complete, you will now write a structural Verilog description of the circuit. Structural description means that you will use AND/OR/NOT/XOR/etc. gate instantiations (Verilog primitives, as we covered in the class, also as shown in Figs. 2.37, 2.38 in the 3rd edition textbook) so that your Verilog corresponds to your schematic.
- Functional Verilog: As the third exercise, you will now re-design the same circuit using continuous assign statements (as shown in Figs. 2.40 and 2.41 in the textbook). This is often called “functional” Verilog as we are describing Boolean functions.
- In this lab, you are not allowed to use behavioral constructs such as the ‘always’ statement. We will do that in later labs.
- Simulation: For the fourth exercise, use a Verilog testbench to apply ALL possible inputs to the three Verilog (.v) files you created. Your goal in this step is to confirm two things. First, that your design is accurate. Second, that all three of the files you created are logically equivalent to each other. Take a second to think about what you have done up to this point. You have generated the same function using three different techniques. In your future projects, it will be up to you to decide which technique to use.
- Synthesis: Finally, you will synthesize (compile) the circuit, generate the schematics, place and route the hardware and generate the bit-file (*.sof) to program the FPGA. Using the Quartus programmer tool, you will download the circuit on the FPGA and show its correct operation. You may choose any slide switches/LEDs from your FPGA board. The TAs will demonstrate this to you in the lab.

A. Lab report submissions

You have to submit a lab report for this assignment. You are expected to document your labs in a professional manner. There is no specific template for the lab report: you may use a standard, single-column, technical report style, and anything between 6-10 pages (including screenshots) would be an appropriate page length for your submission.

Your lab report should have an appropriate title and authorship, and sections on Lab objective, Approach, Design Schematic, Simulation and Synthesis Results section, and a final Conclusion section. To elaborate: Briefly describe your design objective and your approach to the design. Give the logic equations (Boolean functions), show your optimizations (if any), turn in the schematic of your circuit – CAD tools can print the schematics for you. Also, turn in your Verilog code, simulation testbenches and the simulation results.

Take care to draw the schematics neatly and professionally. Organize your layout, the wires and components should be spaced as evenly as possible. Eliminate jogs in wires, line-up the components, etc. You get the idea.

Once you synthesize the circuit, please go through the synthesis report generated by Quartus. It tells you how many Logic-Elements (LE's) your design has occupied and also tells you the longest timing path(s) from input to output and its delay. Use this information in your report.

This is a two week lab. In the first week, make sure you are able to invoke the tools properly, and you learn the design flow. The TAs will help you in getting started with the CAD tools. Once that hurdle is crossed, then the design assignment will not be too much of a problem. For final check-offs, show your design, schematic and the code, simulations, as well as a functioning circuit on the FPGA, to the TAs. Submit the final report on Canvas by the due date.

Once you learn the design and synthesis process in this lab, you'll be ready to start working on some really interesting and non-trivial digital design applications (next lab onwards).

PS: In the next couple of lectures, I will also introduce the Verilog, simulation and synthesis concepts in the class. So don't panic if you don't know how to begin on the first day.