CSE 260 – Digital Computers: Organization and Logical Design

Lab 1

Doug Shook *Due 10/06/2015*

Introduction

In this lab you will design a very simple calculator. You calculator will only perform four operations: shift left, shift right, increment, and decrement. The operation that is perfomed will be chosen by an *opCode*.

If *opcode* = “00” shift left,
If *opcode* = “01” shift right,
If *opcode* = “10” increment by 1,
If *opcode* = “11” decrement by 1.

The values that will be shifted/incremented/decremented will be 3-bit binary values. This means you have three inputs and outputs: an opcode, and input, and an output. You will first be tasked with drawing a schematic for your circuit, then you must implement your circuit in VHDL and test it. Finally, you will deploy your circuit to hardware using an FPGA.

1. Design

Your first task is to design your circuit and draw a schematic. <u>You should draw your schematic on paper, not in Xilinx.</u> Keep the following in mind when designing your circuit:

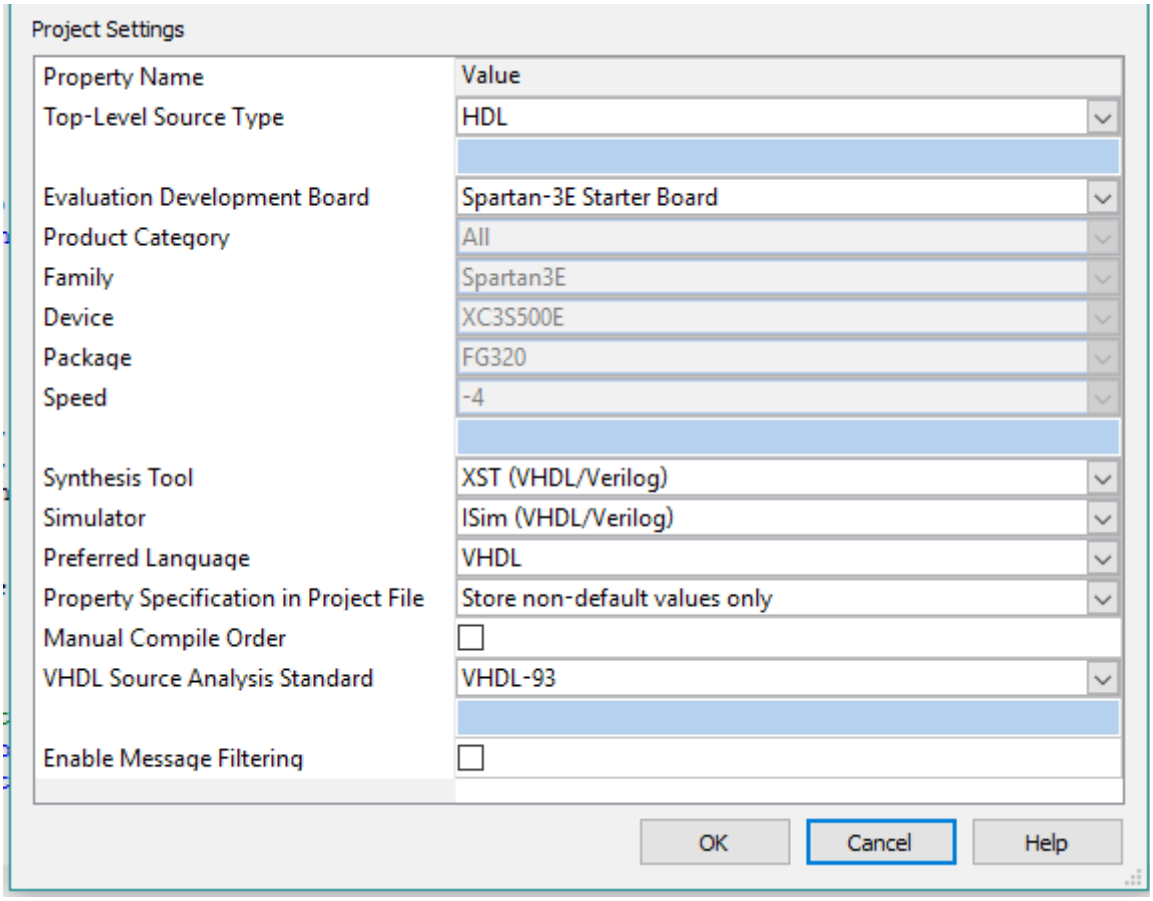
* You should use your 3-bit shifter from the previous assignment. Refer to that lab write up to see what a that component might look like within your circuit. For this lab, we do not care about the Shift Out output, so it does not need to be connected to anything.

* You should use multiplexors when it is appropriate to do so (hint: you will need at least one multiplexor, if not more). We are aiming for an efficient design, so if you avoid using multiplexors you will lose points.

* For this lab, the input, output, and opCode should be treated as multi-bit signals (buses). Your circuit should reflect this. You may split these signals up into individual bits if it is necessary for you to do so, but everything needs to be labeled appropriately.

2. Implementation

For this lab, you will be implementing your circuit using VHDL. Open Xilinx and create a new Project. Make sure to use the following properties:



In order to use your shifter from lab 1, you must first import it. Go to Project → Add Source, and select your shifter schematic file from lab 1. You can now instantiate a shifter component in VHDL. Create a new VHDL module by going to Project → New Source. You are now ready to define the entity and architecture portions of your circuit.

Remember that your inputs and outputs should all be multi-bit buses. You should have exactly three inputs and outputs to your component – no more and no less.

Note: in order to be able to perform the increment and decrement operations, you will probably need to include the following line at the top of your VHDL file:

use IEEE.STD_LOGIC_UNSIGNED.ALL;

You should also ensure that the circuit you designed in part 1 exactly matches the VHDL code that you have written. We will be looking for this when we grade your assignment.

3. Testing

Once you have finished writing code for your calculator, you need to create a testbench for it. Unlike the previous lab, you are not required to exhaustively test all of the possible inputs. While it would be possible to do so, there are enough input combinations for this circuit to make testing them all a bit annoying. Instead, you must test at least two input combinations for each operation. That means you need

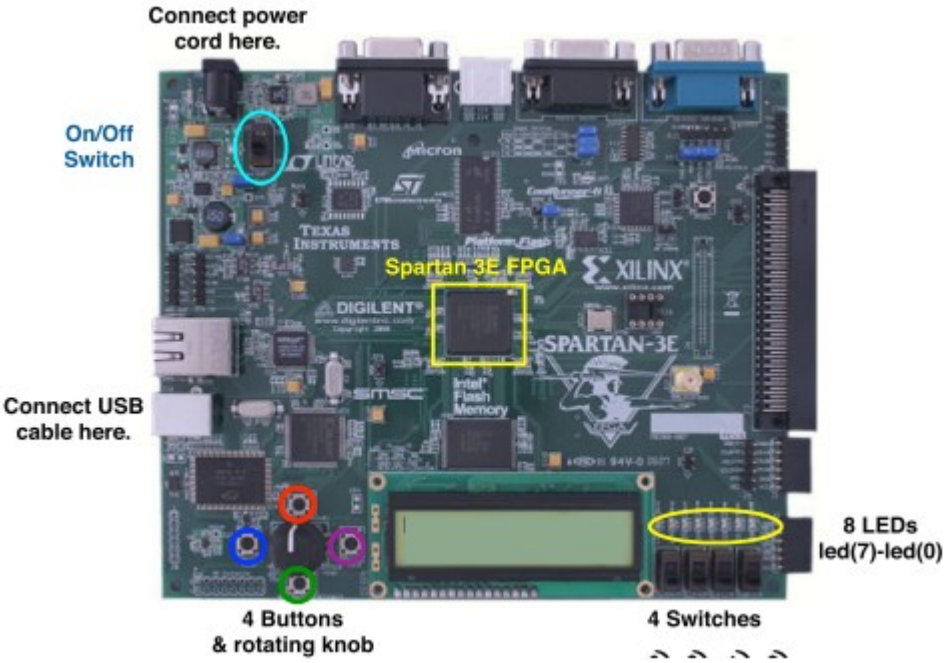
at least 8 tests, though you are welcome and encouraged to create more. When testing your operations try to pick at least one “normal case” - a case where nothing special happens – and one “edge” case – a case where something unique happens.

Once you have developed your tests, you should simulate your circuit to make sure that it is functioning appropriately. Once you are satisfied with your simulation you should print it out and annotate each operation that takes place. For each operation in your simulation you should tell us which operation is happening, and show that the output is computed correctly. You will submit this annotated simulation with the rest of your lab materials.

4. Hardware Implementation

After successfully simulating and testing your circuit using ISim, you will implement your design using a field programmable gate array (FPGA). We will discuss FPGAs in detail later in the course. But, in general, an FPGA is a reconfigurable chip that allows you to configure the number and kinds of gates to use in your design. Thus, if you make a mistake in your design, you don’t need to replace wires or chips. You simply reconfigure the FPGA with each updated design.

There are several options to program a design on an FPGA. In our case, we will use the Spartan 3E Starter Board from Digilent, Inc., shown below.



This board includes, among other devices, a Spartan 3 FPGA, some switches, some light-emitting diodes (LEDs), an LCD, and other cool features. In this lab you will use the FPGA, four switches (labeled SW3 to SW0 on the lower right-hand corner of the board) and the LEDs right above the switches.

You can learn more about the Spartan 3E Starter Board from:

http://www.digilentinc.com/Products/Detail.cfm?
Nav1=Products&Nav2=Programmable&Prod=S3EBOARD

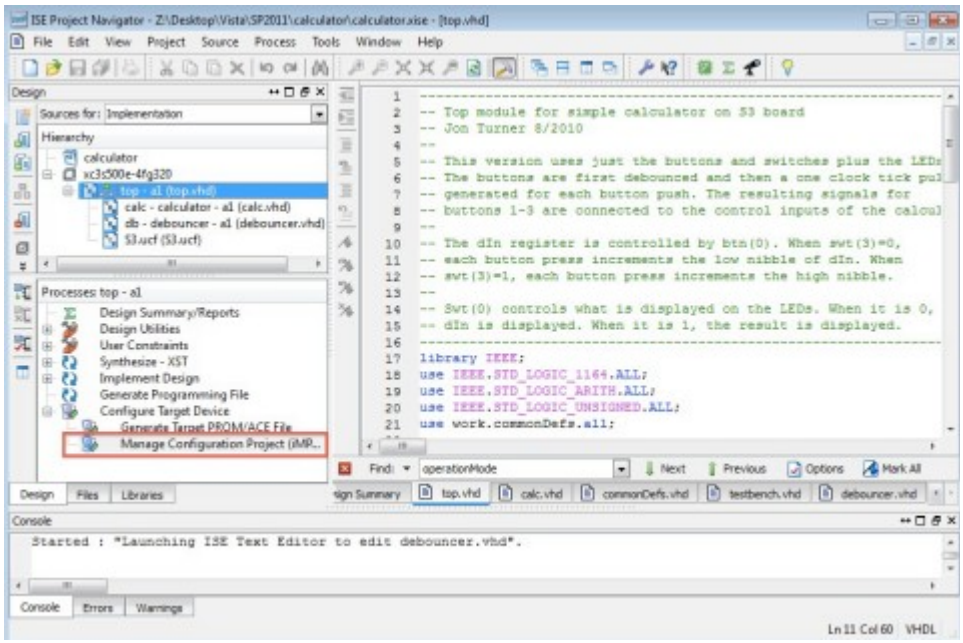
In the next steps, you will configure the FPGA to hold your design. First, you will need some new files that will associate the inputs and outputs of the code you wrote to the switches, knob, buttons, LEDs, and LCD screen on the board. Go to the following URL, download the file, unzip the file, and then add each of the source files to your project using Project → Add Source. Leave all of the options set to their defaults:

http://cse.wustl.edu/~dshook/cse260/labs/lab2.zip

The only change you need to make is to top.vhd. Open this file and change the name of lab02_xx to your actual calculator component name. There are two places in top.vhd where you need to do this.

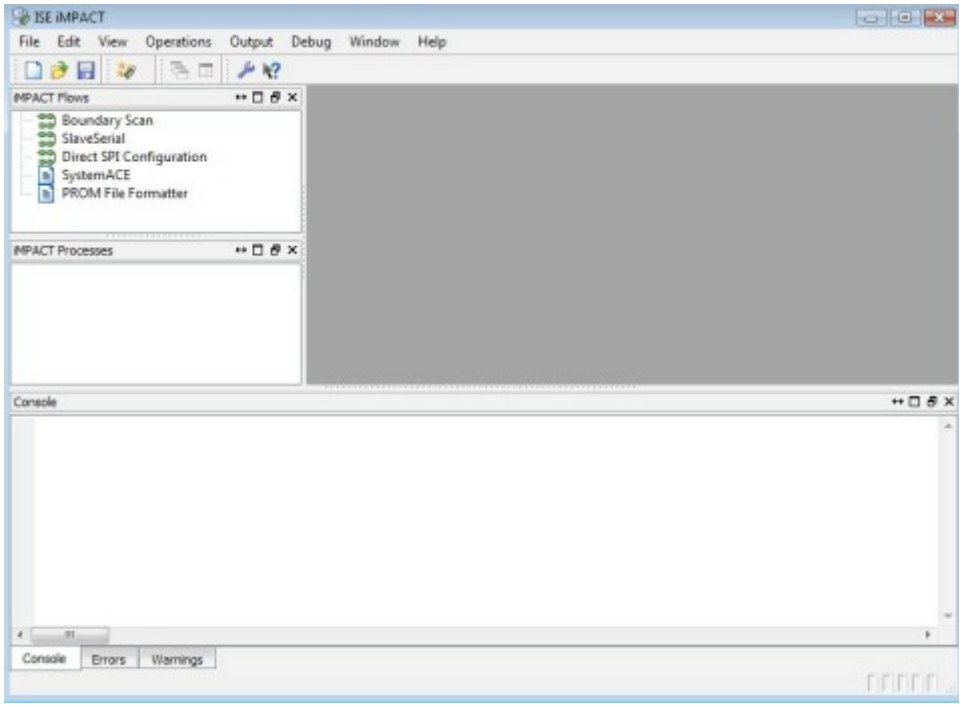
Before completing the next step, make sure the Xilinx Spartan 3E Starter board is on by plugging it into a power source and turning on the power switch in the upper left-hand corner. The power LED should light up, indicating that the board has power. Make sure the cable from the board is connected to the USB port of the computer. (In Bryan 316 you can connect the FPGA via the USB ports on the computer monitors.)

Highlight the text “top - a1 (top.vhd)" in the Hierarchy Window (top left). Now click on the + next to the text Configure Target Device In the Process Window (bottom left). Now double click on the final line Manage Configuration Project.

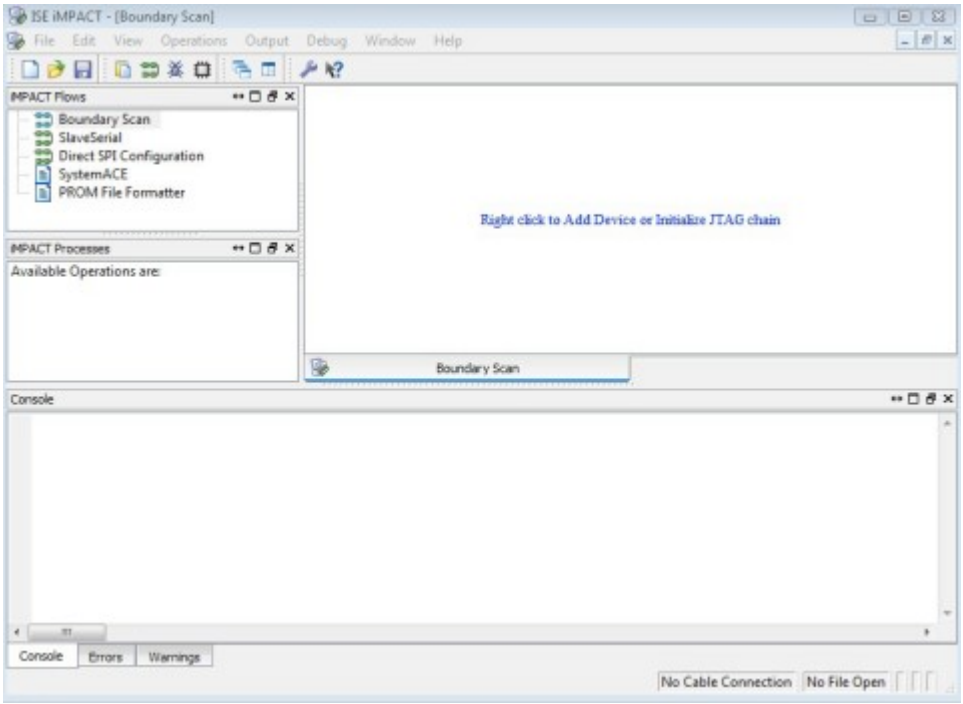


The first thing that will happen is that ISim will generate a bit file of your design. Assuming your design can be implemented without errors, a file named top.bit will be created.

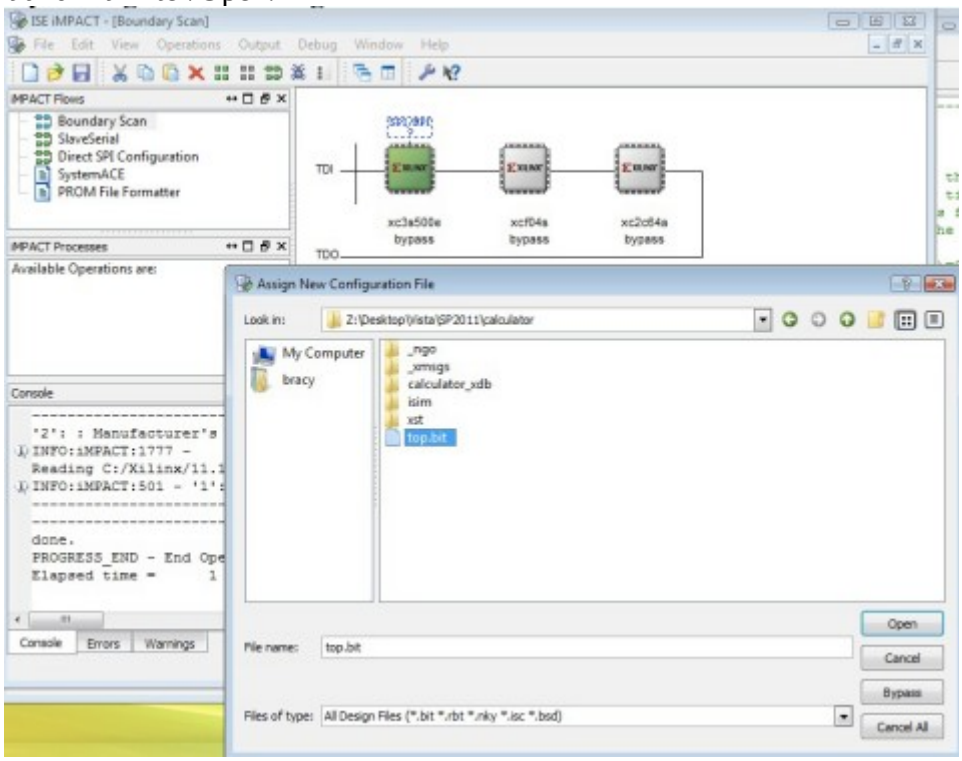
Next, a new window called ISE iMPACT will pop up. It looks like this:



Double click on the Boundary Scan, and you will see the following:



Now right click in the top right window (where it says to right click), and select Initialize Chain. A set of icons will appear, with the left one in green. A window will pop up that says Assign New Configuration File. Select top.bit and hit Enter/Open.



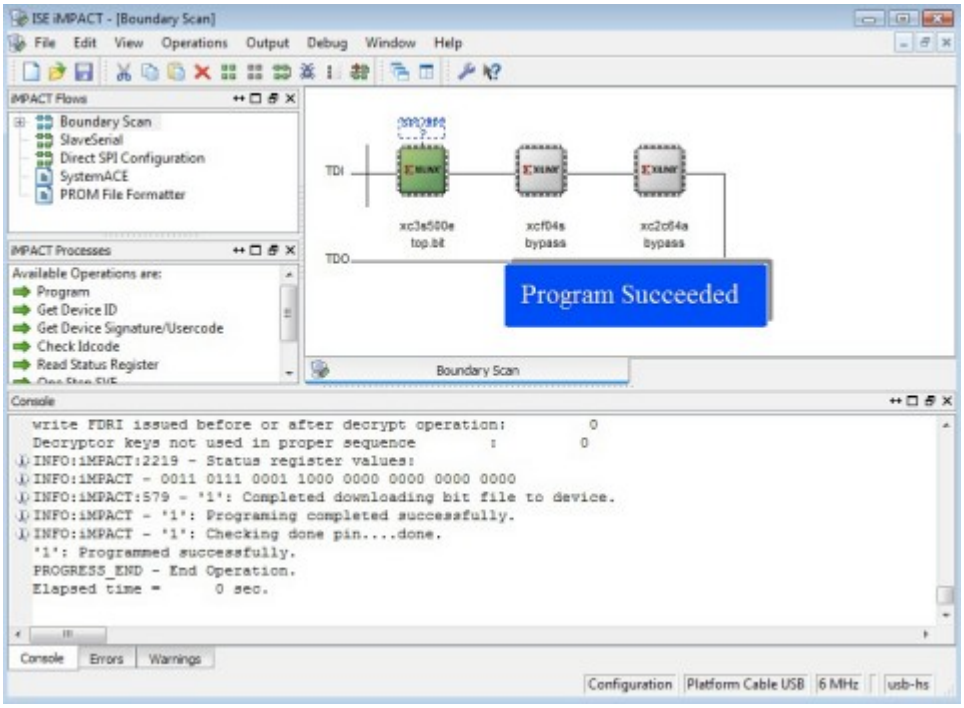
You will be asked whether you want to attach a PROM to this device. Say no.

Now the middle icon will be green and you will again be asked to Assign a New Configuration File. You do not want to do this. Hit Cancel.

Now the right icon will be green and you will again be asked to Assign a New Configuration File. You do not want to do this. Hit Cancel.

Now you will be asked to Okay the Programming Properties. Hit Okay.

Now Right-Click the far left icon (with the text top.bit underneath it) and hit Program. If successful, you will see:



Now you can check out your design on the board. You can use the knob to change the input value. This value will show up on the top row of the LCD screen as well as the 4 leftmost LEDs. The output is represented by the bottom row of the LCD screen as well as the 4 rightmost LEDs. The opcode is represented by the two rightmost switches. As you manipulate the switches you should see the LCD screen and LEDs update depending on which operation is selected.

What to Turn In

You must turn in a hard copy of the following:

* Your circuit schematic.

* A paragraph or two explaining the design choices you made when creating your circuit. For example: how did you decide to implement the opcode? The shifter? The increment decrement? Why did you make the decisions you made? Be sure to comment on anything that may be unexpected or unusual.

* An annotated printout of your simulation, as described above.

* A completed demo form (see below).

You must turn in a digital copy of the following (on Blackboard):

* Your VHDL code for the calculator.

* Your VHDL code for the testbench.

Demo

You must demo your lab on an FPGA before you turn it in. Print out a copy of the demo form from the course website and hand it to the TA. They will walk you through the demo, fill out the form, and then give it back to you. Staple the completed form to the front of your report when you hand it in.