CSE 260 – Digital Computers: Organization and Logical Design

Lab 1

Doug Shook *Due 9/15/2015*

Introduction

In this lab you will design a *logical shifter*. This lab is divided into three parts: design, schematic, and simulation. The first part can be completed on paper, but the second and third part will require the use of Xilinx tools. Please see the last section of this write up for instructions on what you should submit and how you should submit it.

Xilinx Software is available in the SEAS computer labs (including the Urbauer labs and Bryan 316). You may also download and install this software on your own personal machine. Instructions on how to do this can be found in the *Resources* section of the course website. Alternatively, you can use Remote Desktop to access the computer lab computers from your personal machine (https://engineering.wustl.edu/our-school/leadership/offices/engineering-it/networks-remote-access/Pages/connect-to-a-computer-on-campus.aspx)

How does a logical shifter work?

A logical shifter takes a binary value and *shifts* the digits to the left or to the right. For example, if you are given the four digit binary number:

0111

Then a logical shift to the left would be:

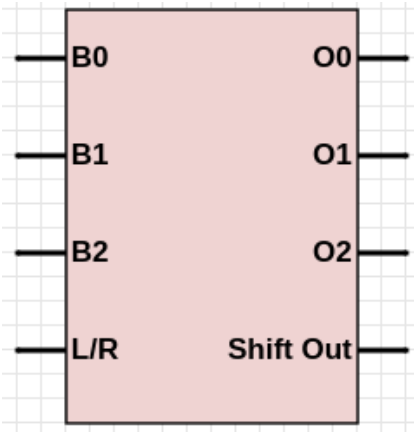
1110

And a logical shift to the right would be:

0011

This has the effect of multiplying by 2 (shift left) or dividing by 2 (shift right). You can multiply or divide binary values by additional powers of 2 by performing multiple shifts. Notice that when a digit is shifted "off the edge" of the number, it is always replaced by a zero.

For this lab you will be constructing a 3-bit logical shifter. A diagram for such a shifter is shown below:



The pins on the left side represent inputs, and the pins on the right hand side represent outputs. B_0, B_1, and B_2 represent the three bits in the binary number to be shifted, where B_0 is the rightmost (least significant) bit. L/R represents whether the shift should be left or right. If L/R is high then the shift should move to the left. If L/R is low, then the shift should move to the right. O_0, O_1, and O_2 represent the three bits of the result, where O_0 is the rightmost (least-significant) bit. Shift Out represents the bit that gets shifted “off the edge” - when shifting right this value represents the remainder after division, when shifting left, this value can be used to verify that the multiplication was performed correctly.

1. Design

Complete the truth table below so your circuit will perform correct shifting. **You must submit this truth table with the rest of your lab**.

L/R	B_2	B_1	B_0	O_2	O_1	O_0	Shift Out
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Using only two-input logic gates (AND, OR, XOR) and inverters (NOT), design circuits based on your truth table. Try to use as few gates as possible. A good approach might be as follows: write out the expression for only 1 of the four outputs (such as Shift Out), then try to minimize the expression using the concepts we discussed in class. Repeat this process for the other outputs, then create your circuit, reusing pieces as appropriate.

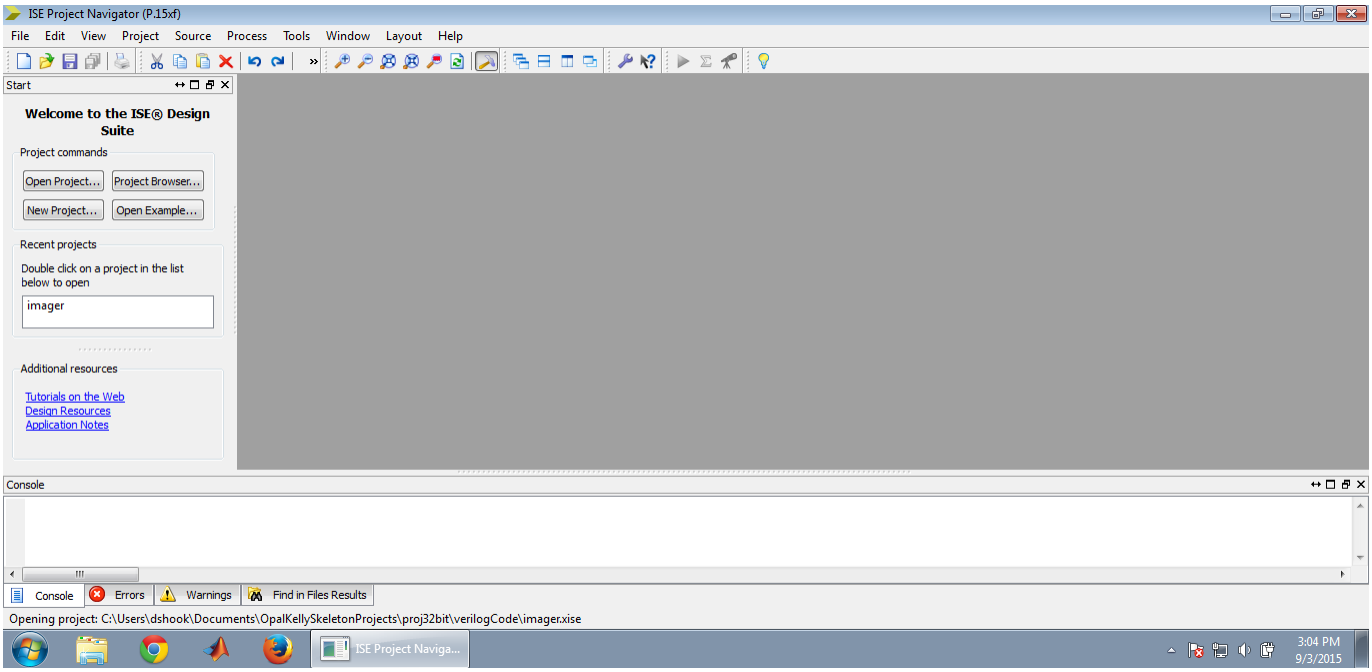
2. Schematic

Now that you have your design, it is time to implement and test it. We will be using Xilinx tools for this purpose. Xilinx will allow us to enter our schematic and create tests for it that can be simulated before attempting to run our circuit on actual hardware.

To use Xilinx you must first create a project. If you are on a SEAS machine, you can start Xilinx by selecting:

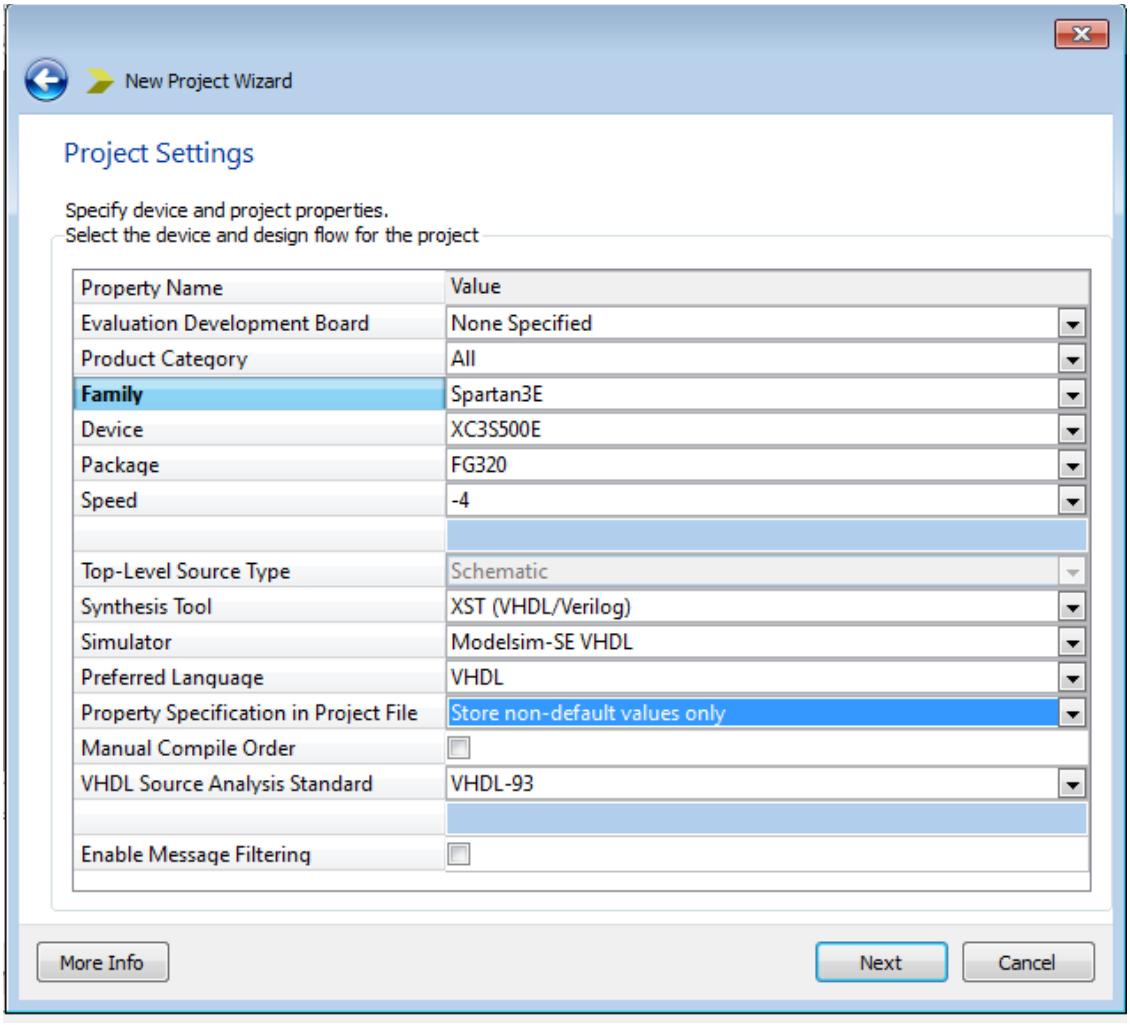
Start -> All Programs -> Engineering -> Xilinx Design Tools -> ISE Design Suite -> ISE Design Tools -> 64-bit Project Navigator

This will open **Project Navigator** as seen below:



In Project Navigator, select File -> New Project to create a new project. In the window that pops up, select a location where the project should be saved (if you are using a lab machine, it should probably be somewhere on your warehouse drive). Next, enter a name for your project (such as “lab01”). For the Top-Level Source Type option, select “Schematic,” then click Next.

The following window asks you to input properties about the FPGA you will be using. You can see the correct options in the screenshot below. Don't worry if you don't know what all of these mean, we will discuss FPGAs in more detail in future classes.

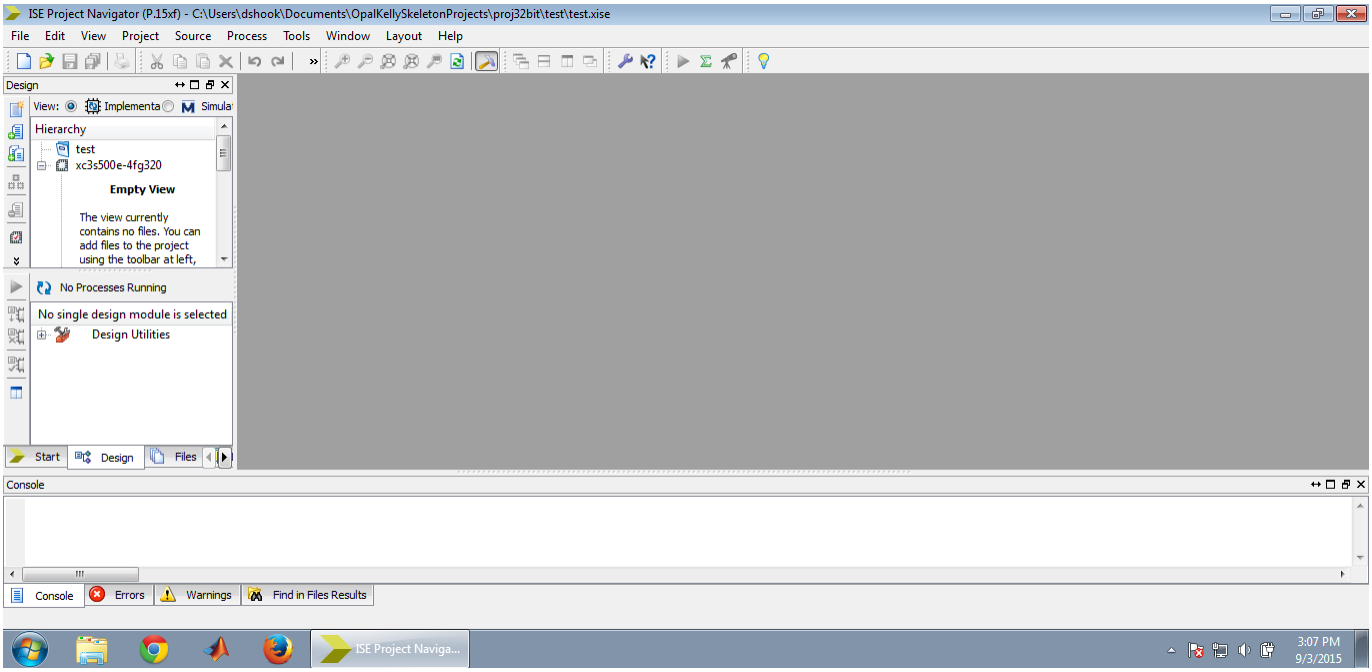


Once your project has been created you must create a source. Select Project -> New Source and choose “Schematic” on the New Source Wizard window that pops up. Give your schematic a name, then click Next and Finish.

The screenshot below shows the main Xilinx interface. You should see four main panes:

- **Design:** Lists the current project's sources (schematics, gates, etc.)
- **Processes:** Lists the processes to perform on the source that is selected (such as simulations)
- **Transcript:** Lists the output of processes, including errors and warnings
- **Working Pane:** Shows open sources and the design summary

Take a few minutes to explore this interface – there's a lot going on! When you're ready to continue, double click the schematic that you created to open it. You should see a window like the one below:



We want to create our circuit by placing gates and drawing wires between them. To do this, first make sure that the Symbols tab is open and selected. If it is not open, you can open it by going to Add -> Symbol.

You'll notice that there are a lot of symbols to choose from. For this lab we will only need four different kinds: *and2*, *or2*, *xor2*, and *inv*. You should be able to locate this by scrolling the the list of symbols, or by typing the name of the symbol you want into the Symbol Name Filter.

To add a symbol to your schematic, first highlight the symbol in the symbol list, then click anywhere on the blank schematic on the right side of the screen. You can rotate the symbol with Ctrl+R. You can use the zoom in and zoom out buttons on the toolbar to examine different parts of your circuit.

To add wires that connect symbols together, select Add -> Wire from the menu, or click the “Add Wire” button on the toolbar. You can then click and drag the mouse to draw wires. If a wire shows up in red that means it is not connected to anything, and will cause an error when you try to simulate your circuit.

Once you are finished drawing wires, you need to define the ports (inputs and outputs). Select Add -> I/O Marker from the menu or click the “Add I/O Marker” button on the toolbar. The “Options” tab in the processes pane allows you to choose which type of marker to place (automatic is fine). Place your inputs and outputs by clicking on wires in your schematic. To change the name of an input or output, simply double click it. For this lab, you should create each input and output exactly <u>once</u>. If you need to use an input multiple times, you will need to connect multiple wires to that input.

Make sure to save your work frequently by selecting File -> Save from the menu or clicking the “Save” button on the toolbar.

3. Simulation

To simulate your circuit, first return to your list of sources by selecting the “Design” tab in the Sources pane, the upper left pane. You must then create a Testbench, which uses VHDL to simulate your schematic. Select Project -> New Source and then choose “VHDL Test Bench” from the window that pops up. Give your test bench a name then click “Next”, “Next”, and “Finish.”

You will now write VHDL code to set values to the inputs every 100ns. You can see an example of two of the possible 16 inputs in the screenshot below:

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
      B0 <= '0';
      B1 <= '0';
      B2 <= '0';
      LR <= '0';
      WAIT for 100 ns;

      B0 <= '0';
      B1 <= '0';
      B2 <= '0';
      LR <= '1';
      WAIT for 100 ns;
      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

Copy these inputs into your test bench, and add the remaining 14 inputs. Its best if you enter them in the same order that they appear in the truth table. Once you are finished, make sure to save your code.

Now we wish to run the test bench and simulate the circuite. Click the “Design” tab on the left to view the project's sources. There are two possible views for this tab. **You will be able to see and do different things based on which view you are using**.

- **Implementation:** shows the schematics associated with the project
- **Simulation:** shows the simulation files (test benches) associated with the project

Select the “simulation” view. You should see your test bench file. Click the testbench file once. You should now see “ModelSim Simulator” as one of the options in the Processes pane. Expand “ModelSim Simulator” by clicking the “+” and then double click “Simulate Behavioral Model.”

Note: The above process will sometimes fail. If you get an error message, try the following: go to Edit -> Preferences. Select “Integrated Tools” from the column on the left. Under “Model Tech Simulator,” enter “C:\modeltech64_10.3a\win64\modelsim.exe” if you are on a lab machine. If you are using your personal machine you will need to locate the modelsim.exe file yourself and use the appropriate location.

If you still have an error, ask the instructor or a TA for help.

When ModelSim is launched, you should be able to see the waveform of the inputs and outputs over time. You can use these timing diagrams to verify that your design is working properly. Take some time to experiment with ModelSim and try out some of the various options. We will be using this tool a lot!

What to Submit

You must turn in a hard copy of the following:

1. Your truth table. This can be handwritten or done on the computer.

2. A print out of your ModelSim simulation. This can be produced using the File -> Print feature within ModelSim. Try to use a few pages as possible – ask a TA if you need help with this.

3. A print out of your schematic. This can be created by using the File -> Print feature within the schematic editor. Try to use as few pages as possible – ask a TA if you need help with this.

You must turn in the following files on Blackboard:

1. Your schematic file.

2. Your test bench file.

If you are working with a partner you only need to turn in one copy of everything, but make sure both partner's names are on everything or you may not receive credit.

You may submit the printouts in class or in office hours, or you may put them in Professor Shook's mailbox which is in Bryan 509.

Demo

To recieve any credit on this assignment, you (and your partner) must demo your work to a TA during office hours. If you are working with a partner, you both must be present at the demo! Print out the last page of this write up and bring it with you to your demo. The TA will fill it out and give it back to you to turn in with the rest of your lab.

CSE260: Introduction to Digital Logic
Lab #1 Coversheet

Due: Tuesday, Sep. 15Th

Your Name _____

Your Partner's Name _____

Names of any students you discussed this lab with:

This section is to be filled out during the demo. TA Name_____

Component	Points (0-3) (3=perfect, 2= minor errors, 1=big problems, 0=not done)	Comments (if less than 3 points, the TA should explain why)
Truth Table	/ 3	
Schematic	/ 3	
Test Bench	/3	
Simulation	/3	
Sub-Total	/3	**TA Signature: _____**

This section is to be filled out after the lab is turned in.

Hardcopy items: ☐ Truth Table ☐ Schematic ☐ Simulation

Digital items: ☐ Testbench ☐ Schematic

TA Signature: _____