EE 331 Digital Systems with HDL
Oregon Tech, Wilsonville, Winter 2014

Lab (Homework) Assignment, due Week 3 (January 21)

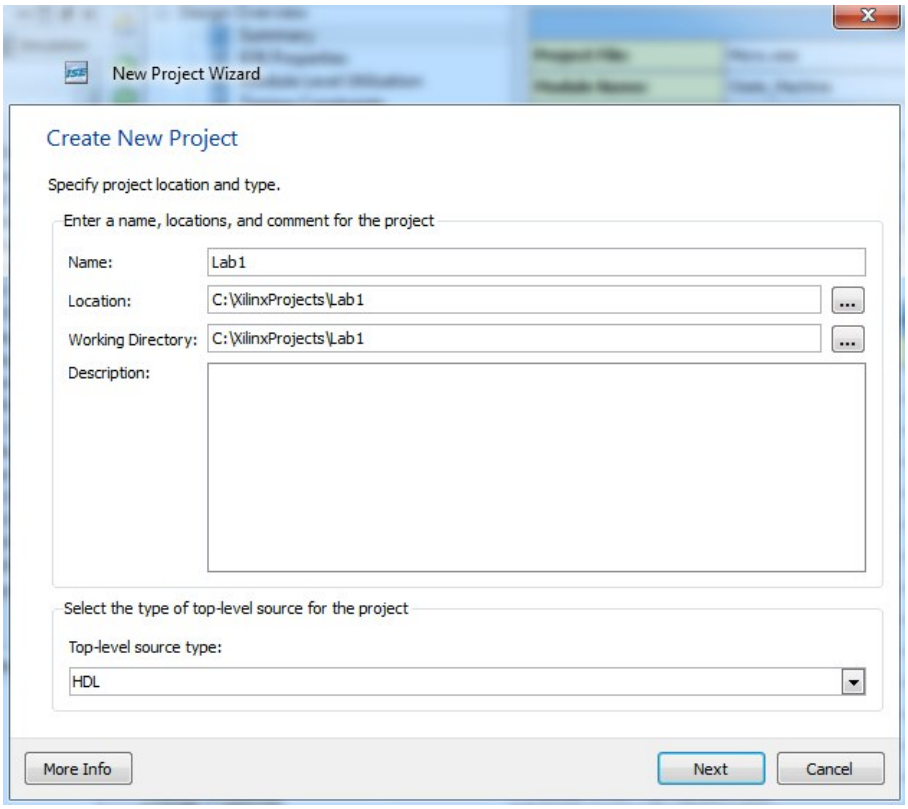
Before starting the assignment you must install the software as described in the document Software_Installation.pdf. This should be completed either during the first class session or by the second session so if you have any problems they can be resolved before the due date of the lab.

The Nexys board has 8 slide switches and 8 LEDs connected to the Xilinx FPGA. You are to create the logic for the FPGA and assign the FPGA pins so that if the right-most two switches are turned on the right-most LED will be turned on, otherwise it will be off. The process is to be repeated for each pair of switches up to if the left-most two switches are on the seventh LED will be turned on. Finally, if the left- and right-most switches are on the left-most LED will be turned on. Based on the labeling in the schematic, we can say:
LD0 is on if and only if SW0 and SW1 are both on.
LD1 is on if and only if SW1 and SW2 are both on.
LD2 is on if and only if SW2 and SW3 are both on.
LD3 is on if and only if SW3 and SW4 are both on.
LD4 is on if and only if SW4 and SW5 are both on.
LD5 is on if and only if SW5 and SW6 are both on.
LD6 is on if and only if SW6 and SW7 are both on.
LD7 is on if and only if SW7 and SW0 are both on.

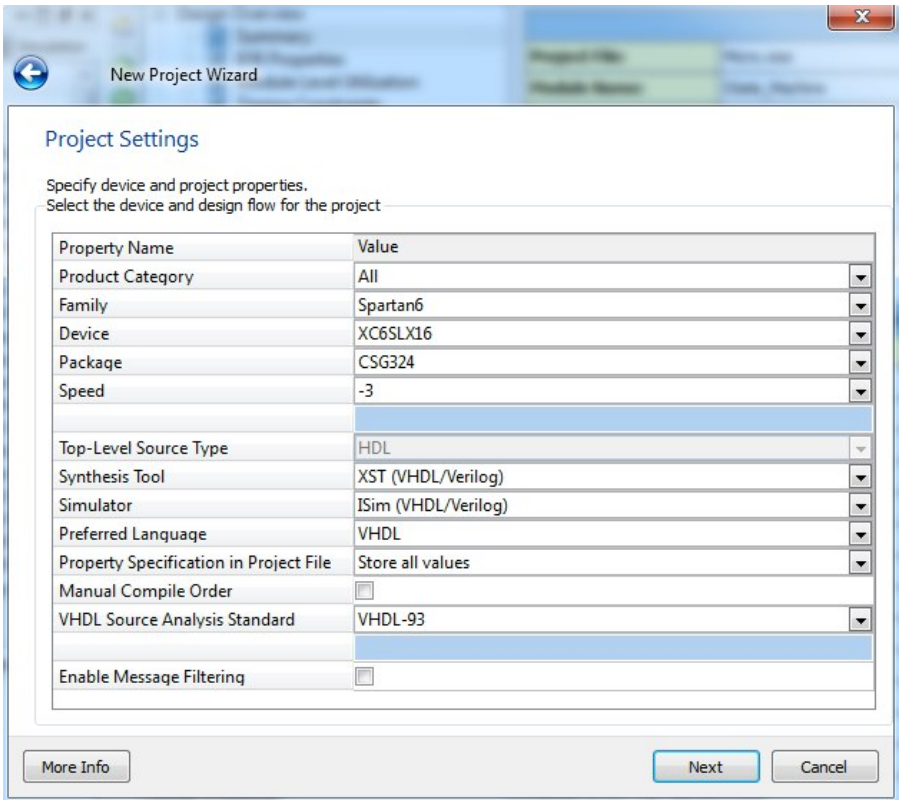
This assignment has three parts. In the first part you will create a project for the assignment and simulate operation using the provided test bench. In the second part you will synthesize the working design from the first part and execute it on your Nexys board. In the third part you will use PlanAhead to do the FPGA pin assignments.

Part 1 — Simulation

Start the Xilinx ISE program and create a new project (File->New Project...). The project name and path should contain no space characters. Make sure the Top-level source type is “HDL”.

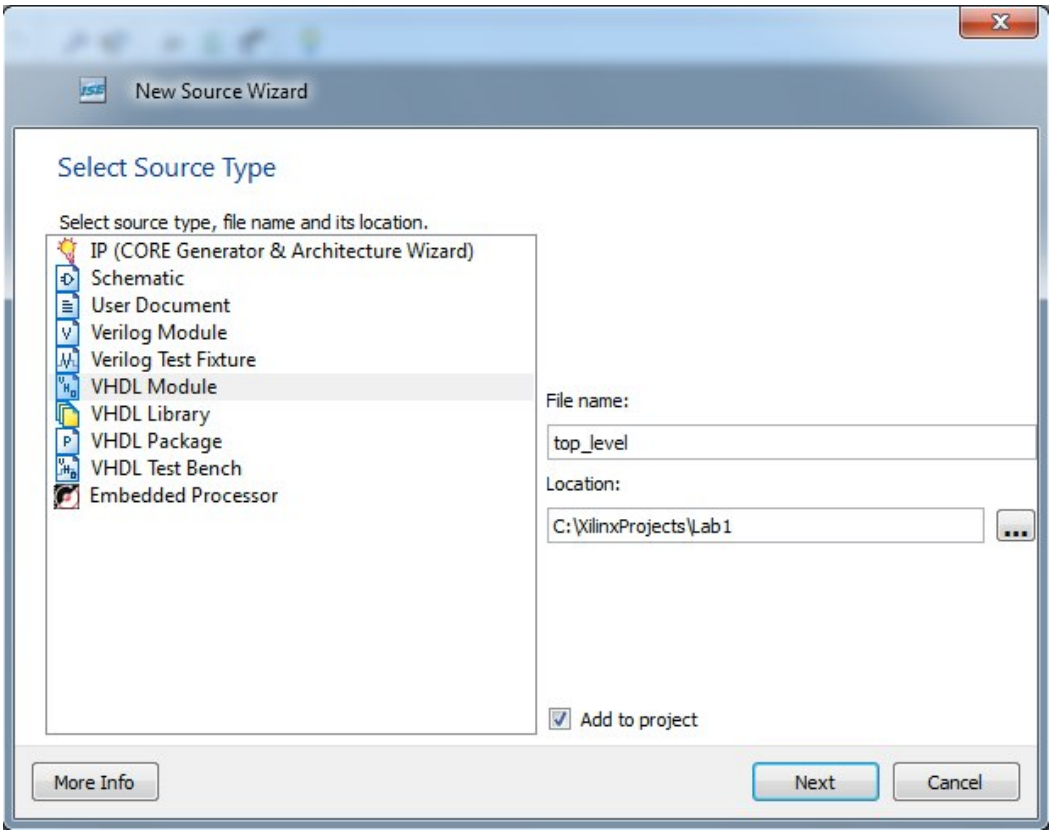


Click on "Next." The Project Settings page should be set to look like the following:



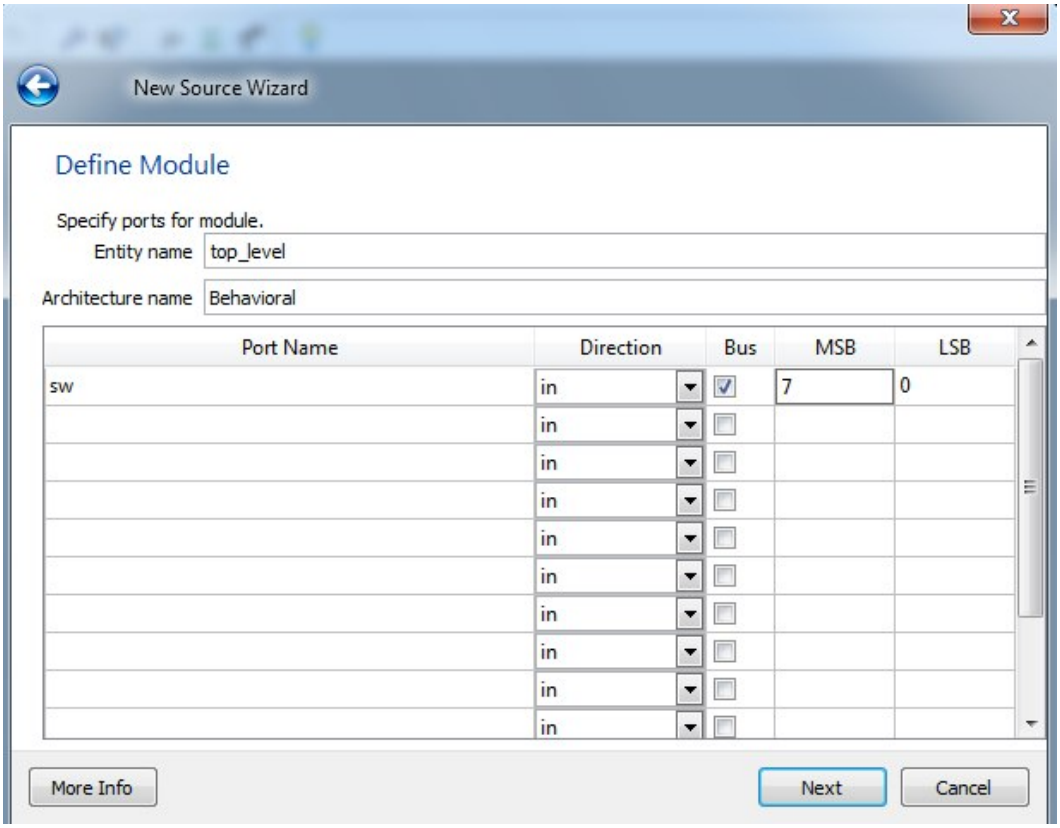
You should verify the settings for each project you make, although Xilinx ISE remembers the last settings for new designs. Now click on “Next” and then “Finished”.

The design has no modules. Create a top-level module that will contain all the logic for your design. Go to Project->New Source... select “VHDL Module” and give it a name. Call it “top_level”. When you name your own modules you may use underscores in the name but do not use any spaces.



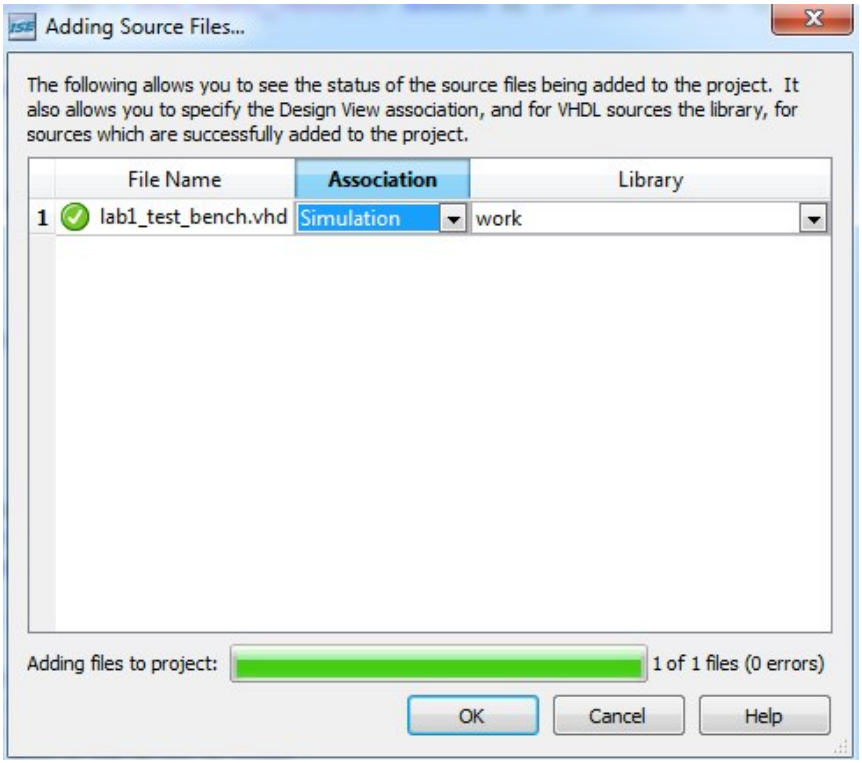
When you click “Next” you will be given the opportunity to describe the ports connecting the module to the outside, either the simulation or the FPGA pins. Because the 8 switches and 8 LEDs are easily described as 8-wire busses, we will describe them as busses. In the image below I’ve filled in the first line for the 8 switches. You will need to also add a line for the 8 LEDs, which are outputs and must be named Led. Why “must?” Well, the connections have already been named elsewhere for you, and the names must match just like the name of the module.

In this course we will standardize bit numbering with the least significant bit being bit 0. To access sw0, we use sw(0). To access sw1, we use sw(1), and so forth.



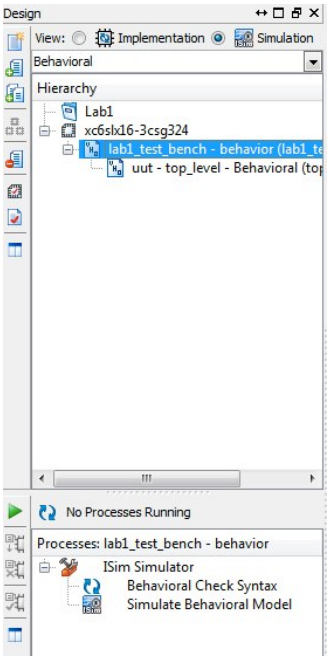
When you finish the New Source Wizard you will have a top_level.vhd file containing the skeleton for a module named top_level with the correct ports already declared. You need to fill it in with the correct logic for the problem. With the top_level module selected in the Hierarchy pane (upper left) you can use the "Check Syntax" process under Synthesize-XST to check your design syntax before proceeding to simulation.

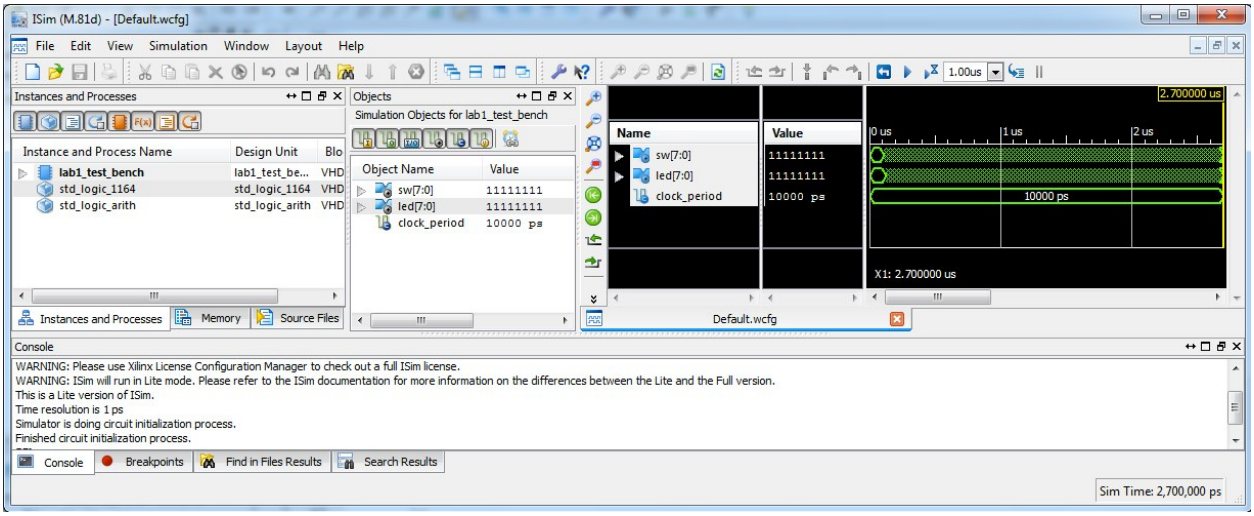
A simulation test bench file has been provided for you to use. The file, lab1_test_bench.vhd is provided on the distribution. You need to add it to your project using Project->Add Copy of Source... Associate the file with Simulation:



Go to the Simulation View and select the test bench:

Right click on the process "Simulate Behavioral Model" and select "Process Properties...". Then change the Simulation Run Time to 2700 ns. Then click on Simulate Behavioral Model to run the simulation. You will get error messages if the port or module names do not match. The Console area at the bottom of the simulator window will indicate if your design did not work properly. You can also pan and zoom in the waveform display to observe operation. The test bench tries every combination of switch inputs and verifies that the LED are lighting properly.





Part 2 — Synthesis

To synthesize the design we need to assign the signals to pins on the FPGA. This is done using a "User Constraint File". Digilent provides one for the Nexys 3 board named Nexys3_Master.ucf which is provided with the course materials. You need to add that file to the project, as was done for the test bench, this time specifying that it is for the Implementation and not the Simulation. Locate the file in the hierarchy pane and double-click to open the file. You need to uncomment (remove the leading hash character) the lines that define the pins you are using. In this case it is the Net lines for the Leds and the Switches. Save the file, select the top_level file in the hierarchy, and then right-click on Generate Programming File in Processes and select Process Properties... Under Startup Options, change the FPGA Start-Up Clock to JTAG Clock. Click OK.

Now double-click on Generate Program File and sit back as the design is synthesized, placed, and routed for the FPGA.

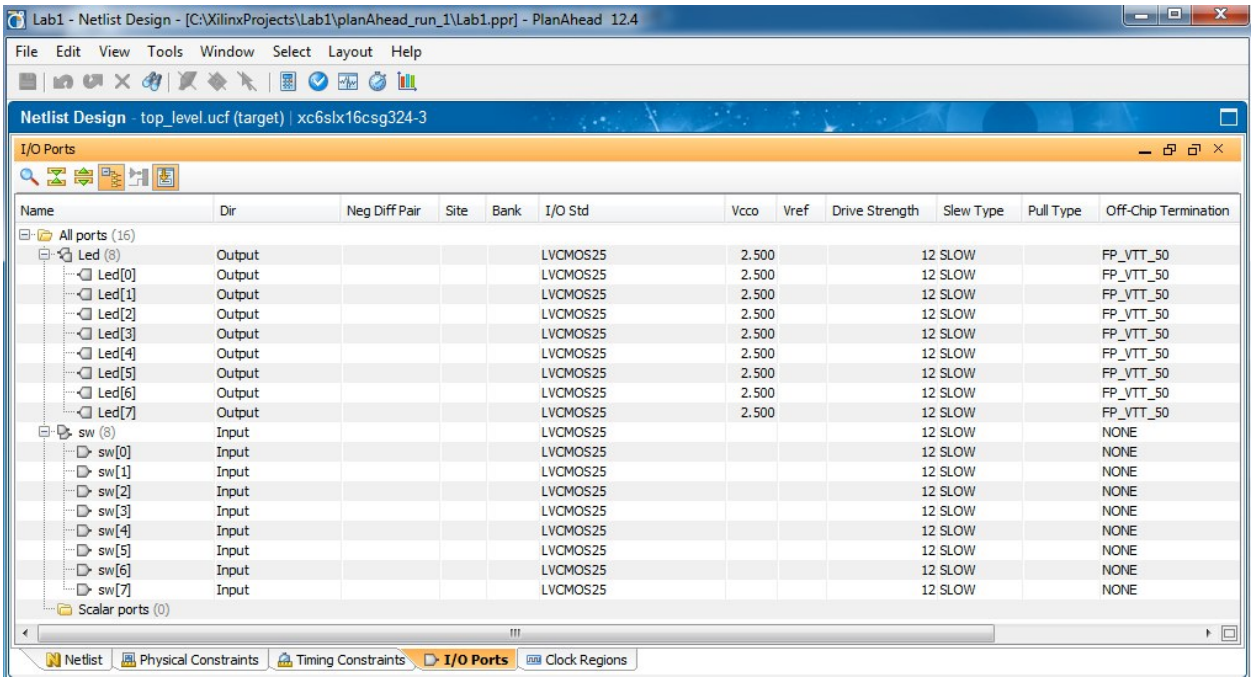
When it says the process has completed successfully, attach the Nexys 3 board to the computer. This will cause the installation of Digilent Adept to complete. Start the Adept program. It should show that it was able to find the FPGA on the Nexys 3 board. Click on Browse and locate the file top_level.bit in the project folder. Click on Program and your design should now be running on your Nexys 3 board!

Part 3 — Using the PlanAhead program for pin assignments

In this part you will use the PlanAhead program instead of the Nexys3_Master.ucf file to do the pin assignments. First you will need to select the Nexys3_Master.ucf file again, right-click and Remove from the project. You have just undone most of your work for part 2!

Now with the top_level file selected in the hierarchy, click on the process under User Constraints named "I/O Pin Planning (PlanAhead — Post-Synthesis)." When the

program has started, enlarge the window on the upper left and select the pane titled I/O Ports.



This shows all the ports and will allow you to assign pins to each of them. Select the row Led(8), and change the I/O Std to LVCMOS33. Press Enter and all the LED rows will change. Do the same for the sw row. Now click on the site for Led[0] and type in U16, the pin name for LD0. Repeat the process for all the LED and switch pins. You can find the assignments in the Nexys 3 users manual, the silkscreen on the Nexys 3 itself, or the Nexys3_Master.ucf file! When you are done, close PlanAhead.

Repeat the synthesis process you did in part 2, and observe operation of the board.

To Turn In:

1. A couple of paragraphs discussing what you did, in a Word (or OpenOffice or Pages or PDF) document.
2. The top level VHDL file, incorporated into the document. Make sure it is in Courier or some other mono-spaced font.
3. Snapshot showing the simulation result (like I did in this assignment sheet). Incorporate into the document.
4. Snapshot of the filled in PlanAhead screen, like the (not filled in) one above. Incorporate into the document.
5. Design summary (for either part 2 or 3). To do this click on Design Summary among the listed files, Make sure the Design Overview Summary is showing. Right click and select "View Report as HTML". When the HTML summary appears, do a File-->Save As and save the HTML file. It is not easy to incorporate this in a document, but if you can manage, go ahead!