

Designing a Decoder

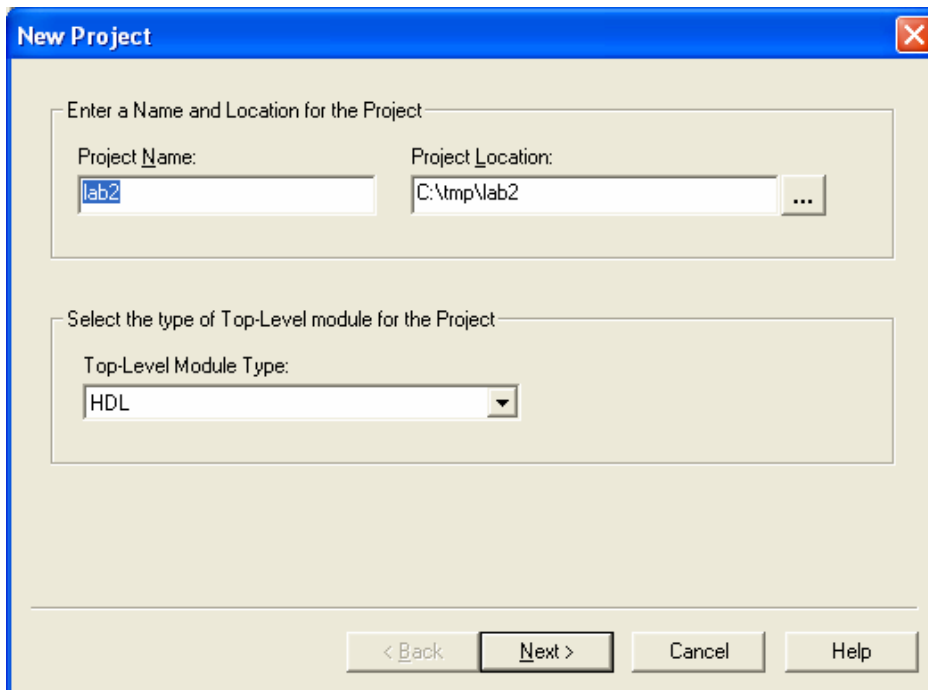
The goal of this lab is to introduce you to the Verilog language and to ModelSim, the logic simulator. The design itself is very easy.

You will design a Hex-to-Seven-Segment decoder. There is a good explanation of a similar decoder in your book, *Logic and Computer Design Fundamentals 3rd edition*, on page 107. The only difference between the decoder in your book and the one you are going to design is that you are not stopping at 9 as an output value, but rather hexadecimal F. You will be able to use this decoder to debug circuits you design later. For more on simple Verilog and a decoder design, refer to page 184 in the book.

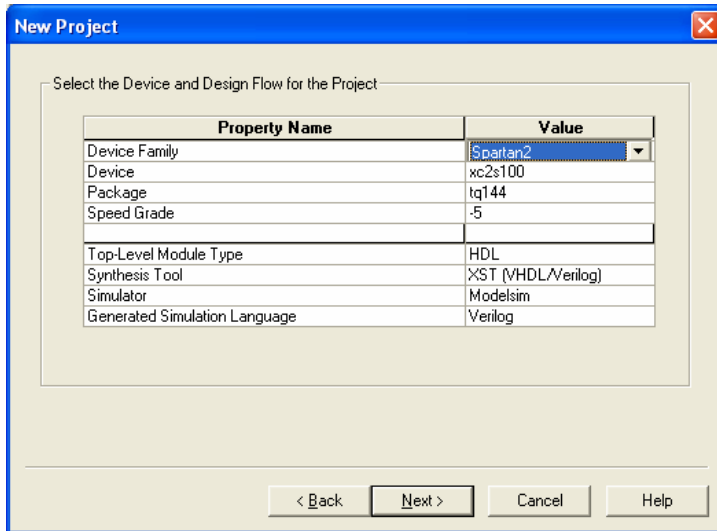
You'll first create a Verilog file, then create a testbench file (containing a set of input values), and finally simulate the behavior. You should also test it on an Xess card.

New Project

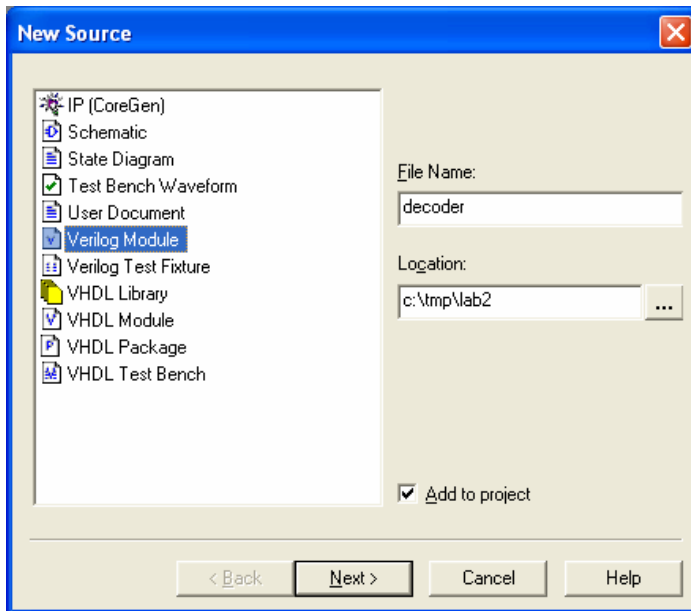
1. Start the project navigator and choose File->New Project. A window will pop up where you will name your project. Make sure you select HDL as the top-level module type.



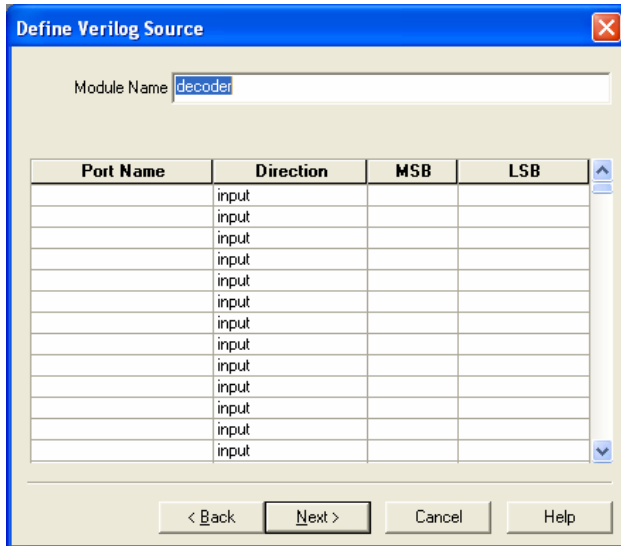
Choose Verilog as the Simulation Language and make sure all the FPGA info is correct.



2. Add a source file, select Verilog Module, and name it “decoder”.

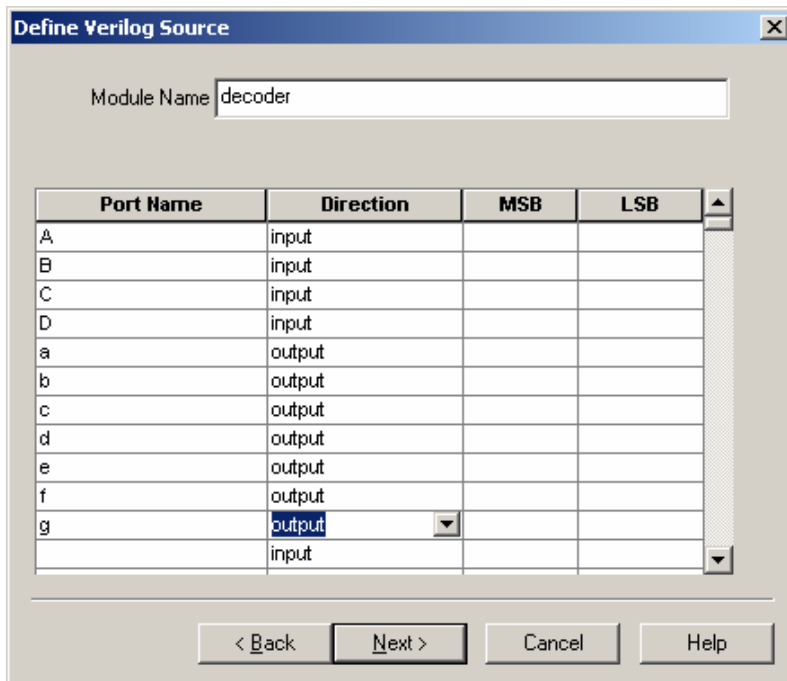


3. Click on next. The following window will pop up. You can either enter the inputs and outputs here, or in the Verilog code itself

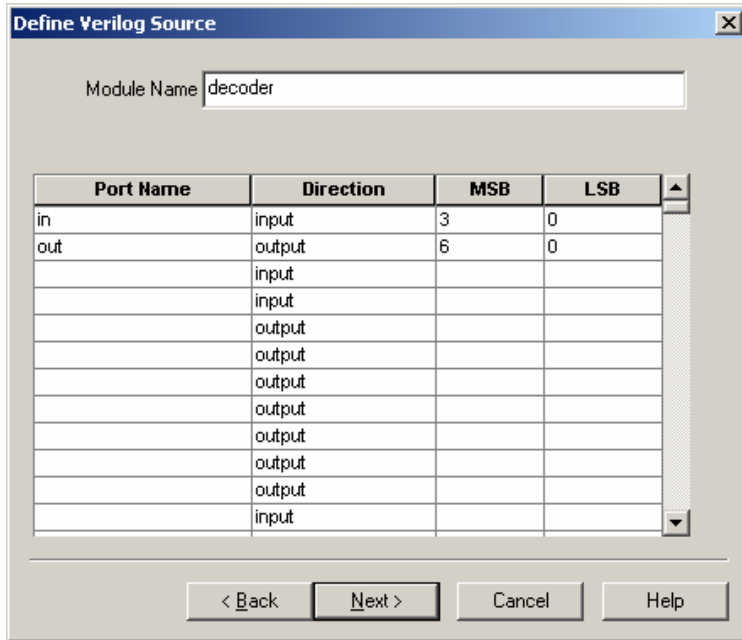


For the decoder, you have 4 inputs and 7 outputs. Another way of thinking about it is that you have 1 number as input (which is 4 wide) and 1 as output (which is 7 wide). Therefore, depending how you think of it will depend what you will type in the Define Verilog Source window.

If you choose to implement 4 inputs and 7 outputs then the window will look like this.



If you choose to implement as 1 input (size 4, numbered 0 to 3) and 1 output (numbered 0 to 6) then,



As you'll see in the Testbench section, there's an advantage to implementing the input as a single variable.

4. Either way, click on Next and Finish. Back in project navigator, a Verilog file will be generated. The code is left for you to finish. If you have chosen to implement the decoder with 1 input and 1 output, you can access each wire as *name[0]*, *name[1]*, and so on.

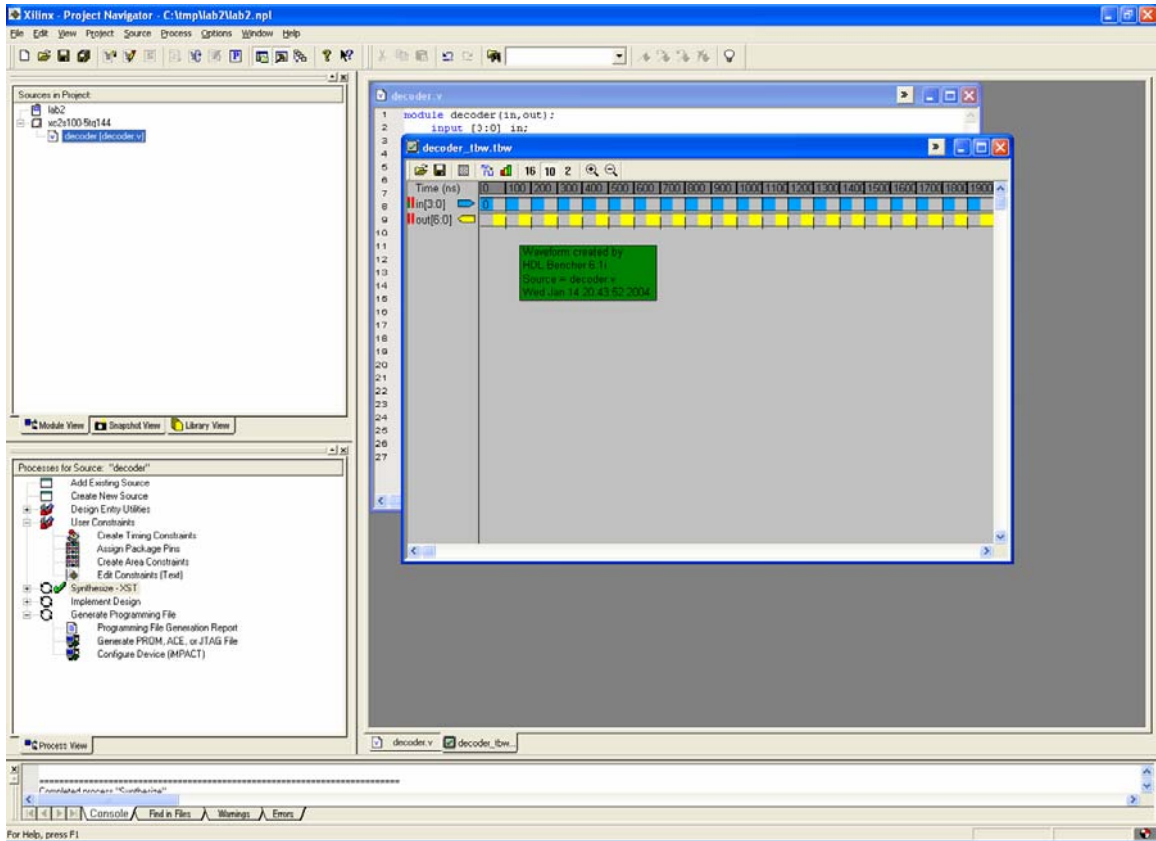
Synthesis

After you have written the code, double click on the Synthesize button in the Processes for Source pane. If you have any errors it will report them in the log file.

Testbench

1. Once you have it synthesized, add a new source. Choose test bench waveform and name it *decoder_tbw*. It will be associated with "decoder" because that's the only Verilog file you have. In general, you may have to make the association.

2. A window called Initialize Timing pops up. Ignore the timing for now and click OK. This will take you to the main program. On the top half of the screen you will see the following window.

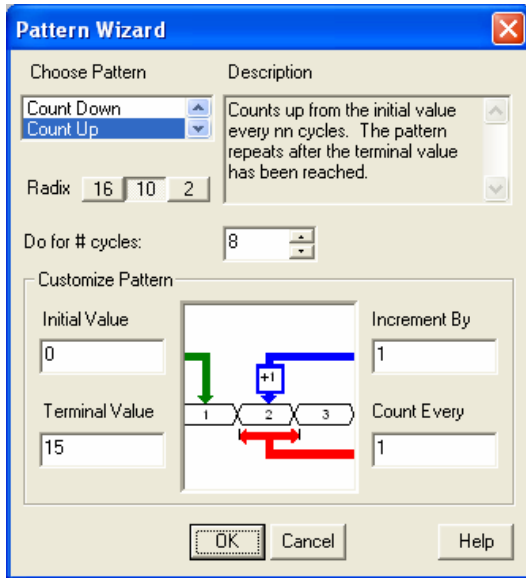


If you like, you can undock the window by clicking the icon with the two greater-than symbols.

Again, how this window looks will depend on how you chose to represent your inputs and outputs. In this particular example, there is one input and one output. Outputs are not important for now.

3. To set the inputs to desired value, click on the blue box of each input. If you have inputs that are only one wide (which will be the case here if you have chosen multiple inputs), when you click on the first blue box, that will toggle the value to 1 or 0. Clicking on the second, or third or one of the following boxes of the same input will toggle that value to zero or one again.

4. If you have chosen to have only 1 input that is 4 wide, there's a simpler way. Clicking on the first blue box will bring up a pattern button. Click on it, another window will pop up.



This function will automatically generate a set of inputs. Set it to run for 16 cycles.

5. Right click at the end of the inputs and select “Set End of Testbench”.

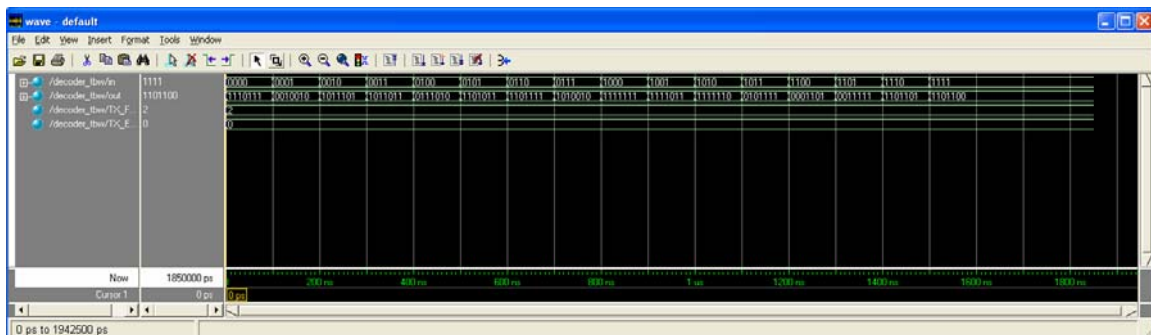
6. Save the testbench waveform file and exit.

ModelSim

1. Make sure the decoder_tbw file is highlighted. In the Processes for Source window, expand the ModelSim Simulator group. From the list double-click Simulate Behavioral Verilog Model.

2. This action will pop up 4 different windows. For right now, the most important one is the Wave Window, so make it larger. For a particular input are you getting the corresponding result? If so, then you have implemented this decoder correctly. If not, then go back and check your logic.

The filled magnifying glass will zoom so that the waveform fits in the window.



Downloading the Design

You should also check to see if you are getting the right result by downloading the design onto the board. But before you download your design, don't forget to connect inputs and outputs to the appropriate ports.

Once your design is downloaded onto the board you should be able to flip the switches and get the appropriate value on the display. Make sure that the LA sees your design implemented on the board.

Mail the instructor an image of the Modelsim window showing the test. Also, send a Zip archive of the project.

Things you learned:

- Some simple Verilog syntax
- How to use Test Bench Waveform
- How to simulate your design using ModelSim

Anselmo Lastra and Aleksandra Krstic
Original: 11/13/02. Updated 1/12/04