Xilinx Tutorial – Basic Walk-through

Introduction to Digital Logic Design with FPGA's:

Digital logic circuits form the basis of all digital electronic devices. FPGAs (Field Programmable Gate Array) are large programmable digital electronic devices. They are very generic and non-functional until the digital designer programs its functionality. Instead of wiring chips together the logic is connected using a CAD (Computer Aided Design) program to make the connections required to implement the function(s) the designer has in mind. There are different ways that this task can be accomplished. One is using a text language called a HDL (Hardware Description Language). Another is called schematic entry.

This tutorial will walk you through the steps to implement a function that will detect a prime number between 0 and 15 using a FPGA. The numbers are 1, 2, 3, 5, 7, 11, & 13; or in binary 0001, 0010, 0011, 0101, 0111, 1011 & 1101, respectively. The input will be a 4-bit (bit = binary digit) number called a bit-vector or bus. This input will be supplied by the student using the switches SW3-SW0. The outputs will be examined by observing LED's on the board.

Steps Involved:

1. Creating the (a) schematic or (b) HDL model using CAD software,
2. Synthesizing the circuit for the FPGA (analogous to compiling a software program),
3. Simulation (or test phase),
4. Configuring the FPGA with our .bit file, and
5. Testing the design.

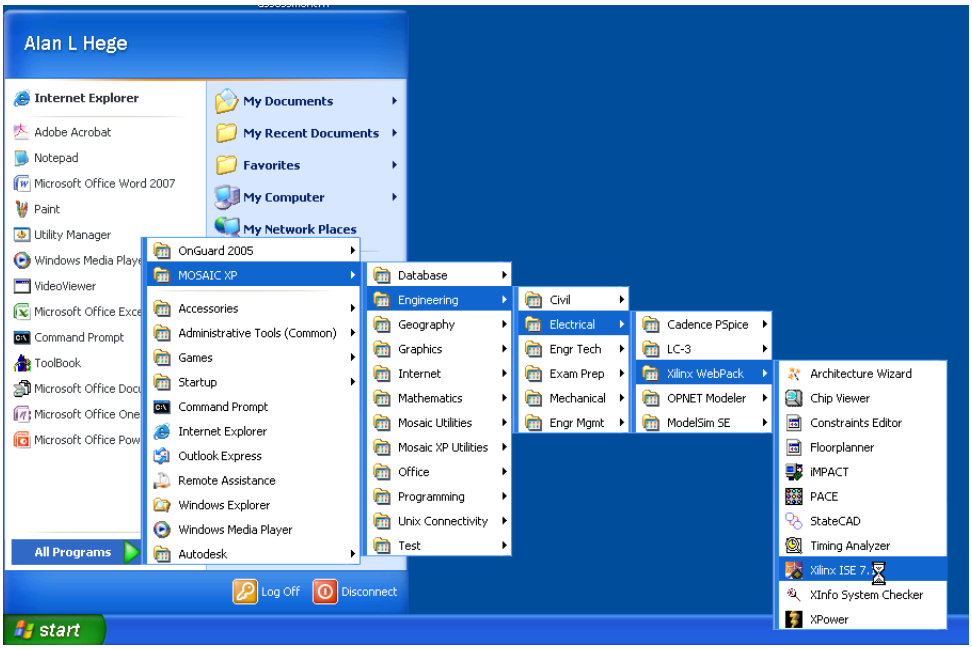
Step 1 – Starting a Logic Design Project:

- Start by creating a directory in your 'My Documents' folder, call the directory "Projects".
  - You can subdivide your directory structure any way you like; however, it is highly recommended that you create separate directories for each project.
  - For most CAD software packages, there are many file created for each project. Sometimes the files have the same names. Also, it is easier to move your project from computer to computer if you keep the projects organized and separate from one another.

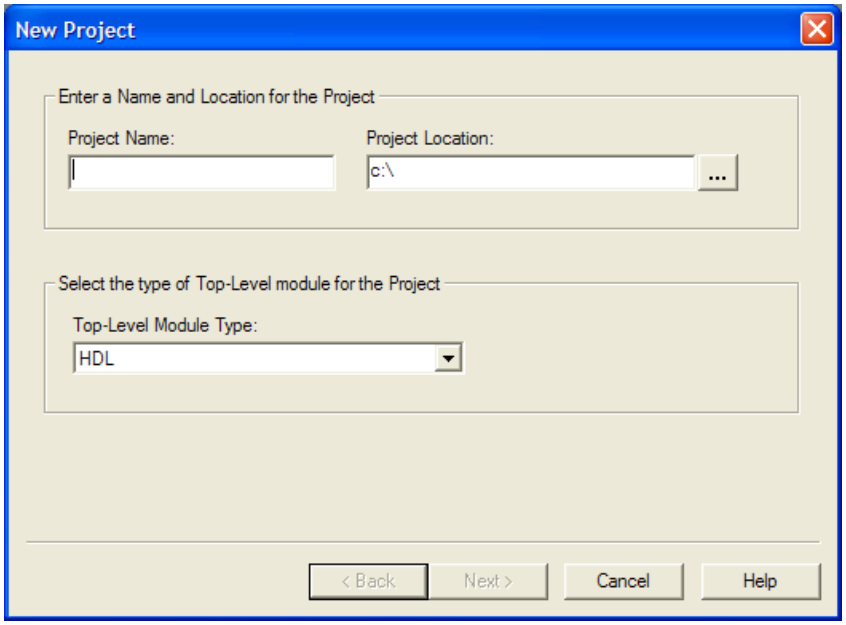
- Under the 'Projects' folder create a folder called "PrimeDetector". This directory is where the Xilinx CAD software will build your project.

- *Note:* you could also build your project on a UFD (USB Flash Drive). This is the method the author will be using during the tutorial.

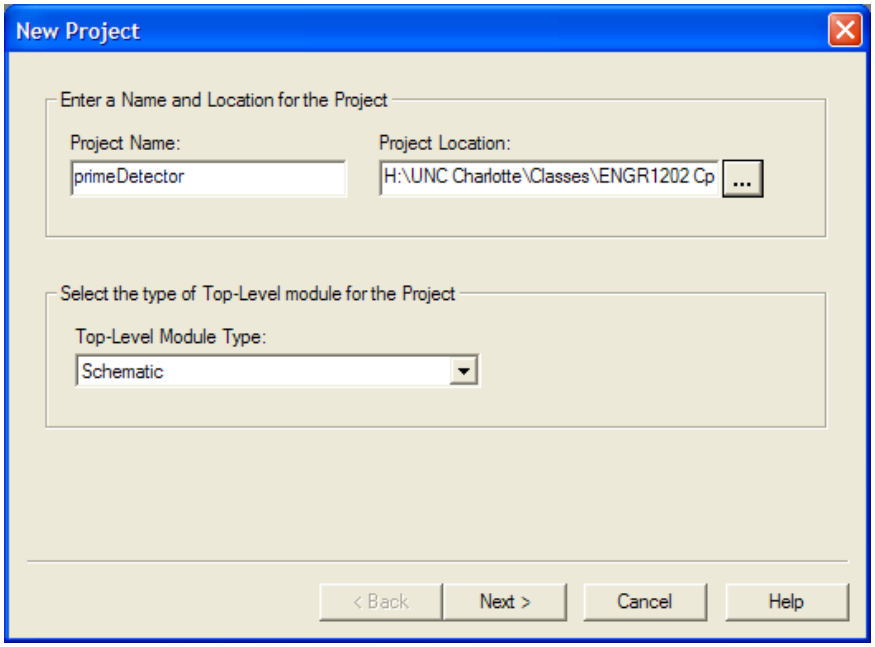
- Start the Xilinx Project Navigator.
  - "Start → All Programs → Mosaic XP → Engineering → Electrical → Xilinx Webpack → Xilinx ISE 7.1i"



- After the program opens start a new project … File → New Project. A project wizard will open as follows.



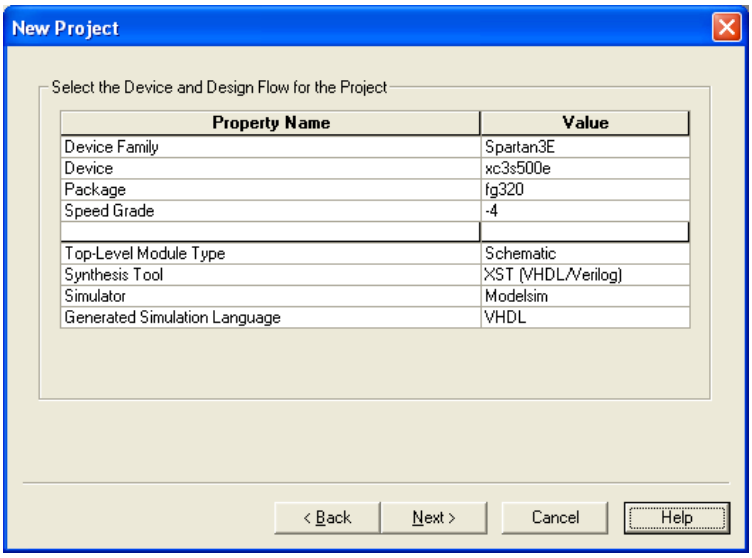
- Under Project Name type primeDetector, under Project Location click on the elipse and browse to the project directory "primeDetector" you created above. Finally, select Schematic from the drop down list under "Top-Level Module Type:"



- Click Next.

- This next page of the wizard will set up the chip information for the FPGA you are using. The FPGA information is printed in small text on the top of the FPGA.
    - For the Digilent, Inc. Nexys2 boards use the following settings:
        - *Device Family:* Spartan 3e
        - *Device:* xc3s500e0
        - *Package:* fg320
        - *Speed Grade:* -4
        -
    - For the Spartan3 Starter Kit board use the following settings:
        - *Device Family: Spartan 3*
        - *Device: xc3s200*
        - *Package: ft256*
        - *Speed Grade: -4*

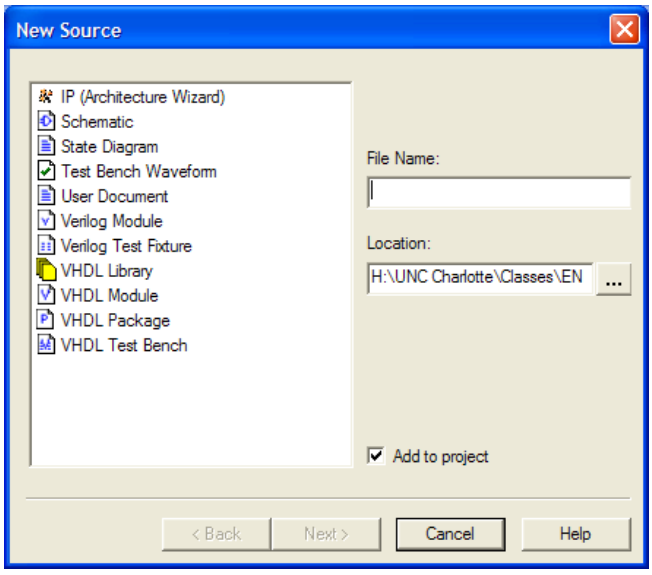
      Note: this information is printed on the top of the FPGA IC.



    - Do not change any of the settings in the bottom section. The default setting will suffice.

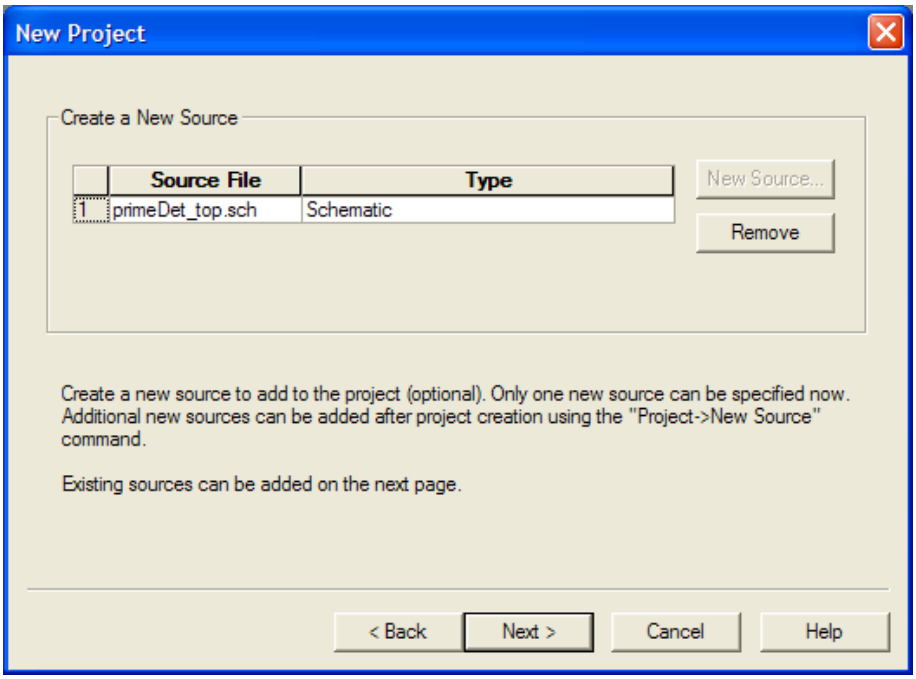
- Click Next.

- Click on the "New Source" button. The following window will open …



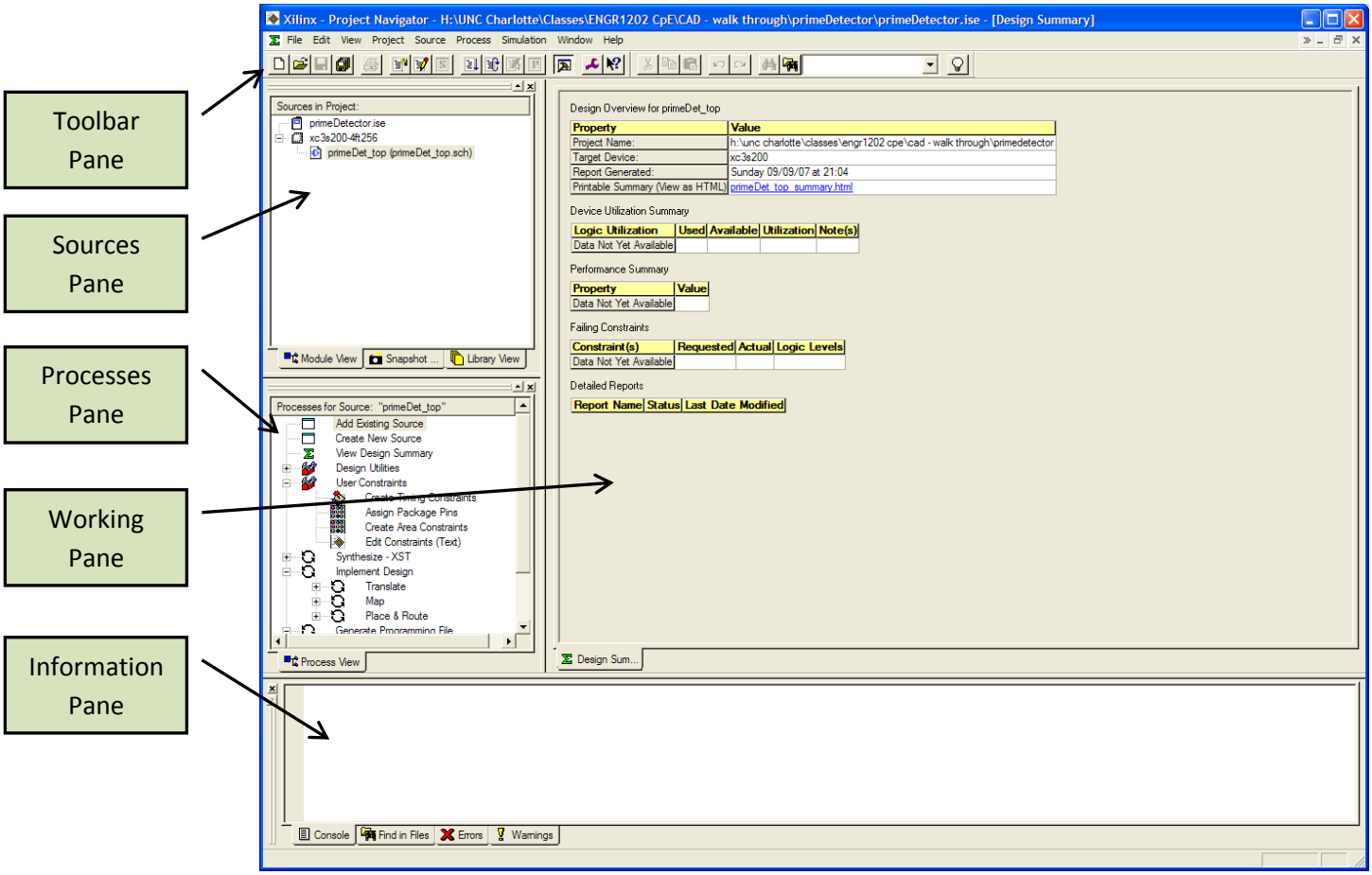
- In the left half of the window, click on Schematic. Then, type primeDet_top for the Filename. Do not change the location of the file. This is the top level schematic.

- Click Next and then click Finish. The following pane in the New Project wizard should be displayed.



- Click Next. Since we do not have any existing schematics to add to this project, click Next again. Finally, the project summary is shown. Click finish and your project will be set up.

- The Project Navigator window should look like the following:



- Double click on the primeDet_top.sch item in the Sources pane. An empty schematic sheet will open in the Working pane.

- You have now formed the basis structure of your project.

- To periodically save all your work, click on the Save All button. 

Step 1.1 – Schematic Capture (graphical model of circuit):

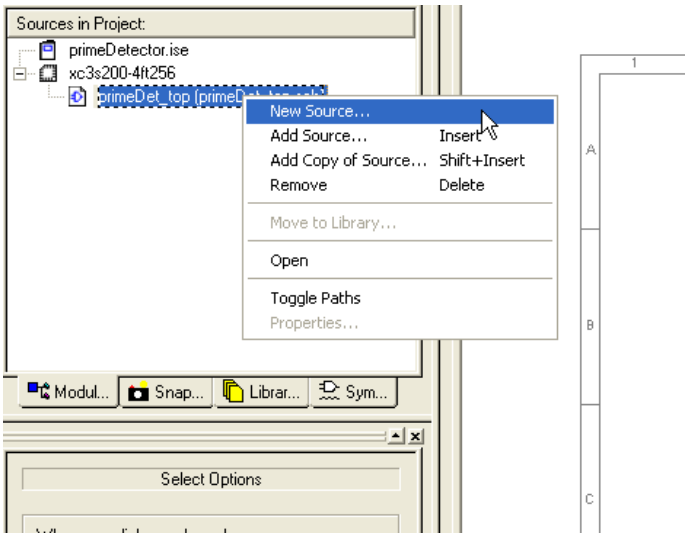
- The student will now create a schematic model of the Prime Number detector.

- In the Sources pane, right click on the top-level schematic and select 'New Source'.

- A new window appears … select schematic on the left and type in the name of the new schematic.

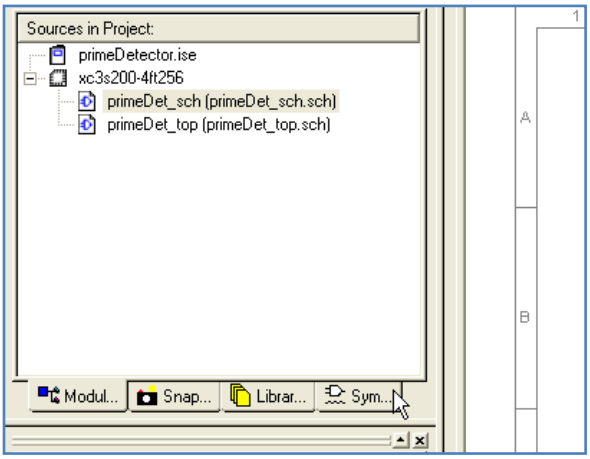
- Click Next, then Finish.

- Notice that the new schematic sheet is on the same level as the "top" schematic. This will change later when you add the schematic to the top-level design schematic.

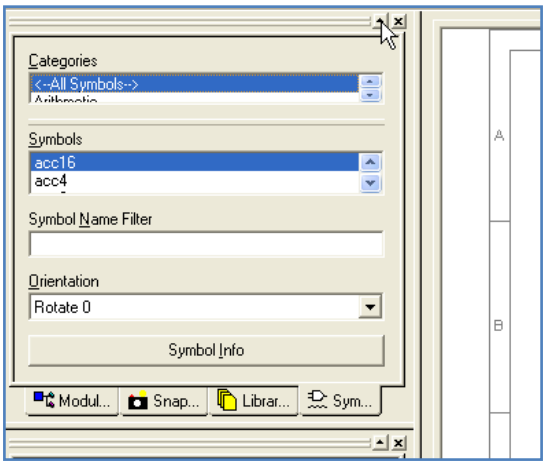


- Now the student will add the gate symbols to make up the circuit.

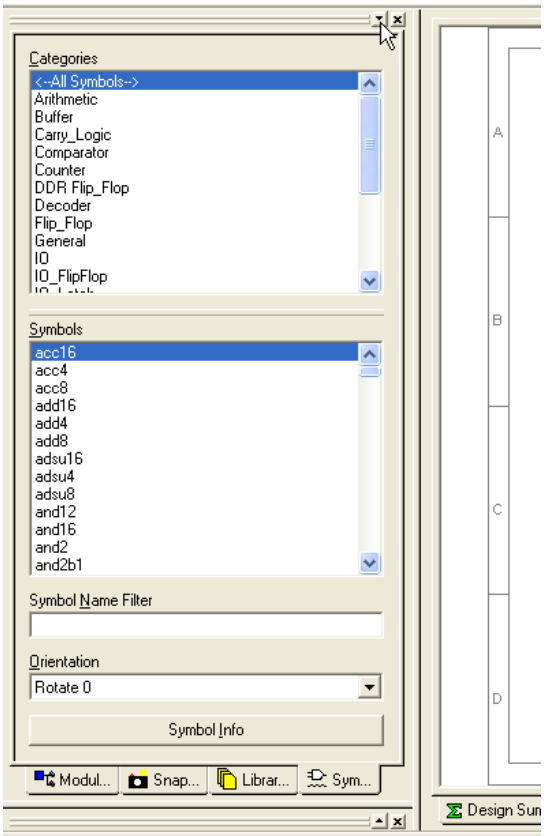
- To add symbols select the Symbols tab in the Sources pane.



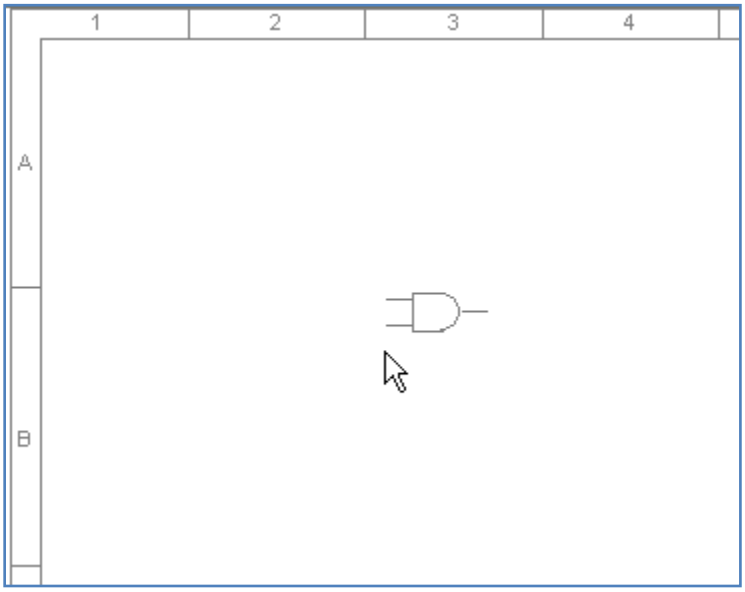
- Now click the up-arrow in the upper-right of the Sources pane to improve the visibility of the various fields within the Symbols tab of the Sources pane.



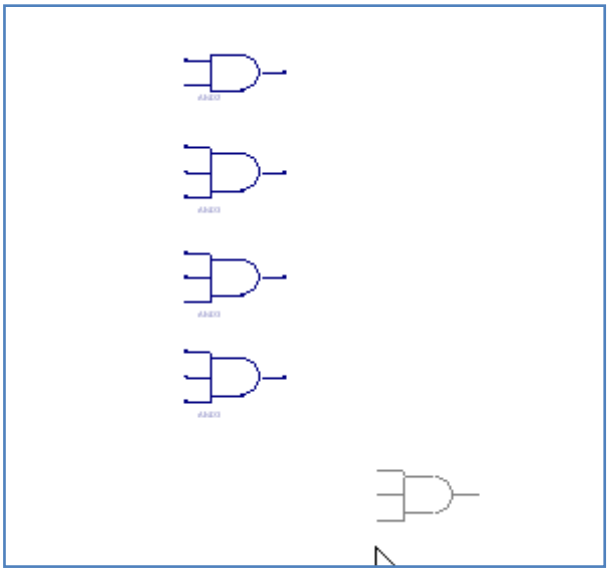
- A down arrow now shows up in place of the up arrow and the Sources pane is expanded. You can do this anytime you like to improve the visibility of the Sources pane.
- In the Categories box, scroll down until you see the Logic item; select Logic by left clicking on it.
- The Symbols box now shows only logic gates.
- To narrow the selection, left click on the 'Symbol Name Filter' box and type "and".
- Left click on "AND2", this is a two input AND gate.



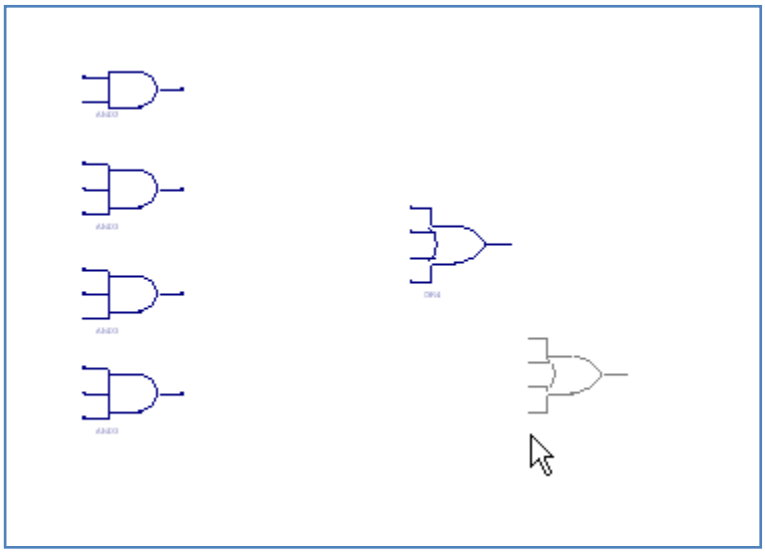
- Move the mouse cursor to the working window. The working window should be primeDet_sch.
- Left click in sector B3 of the primeDet_sch schematic sheet. This will place the component.



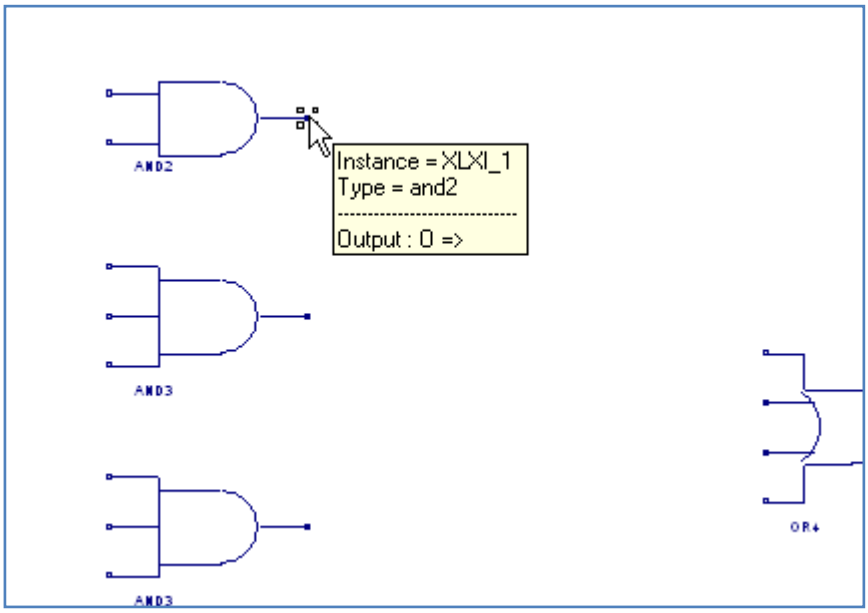
- Now select AND3 from the Symbols box and place that symbol by repeatedly left clicking in the positions shown in the figure to the right.
- NOTE: if you make a mistake press Ctrl-z to undo last action.

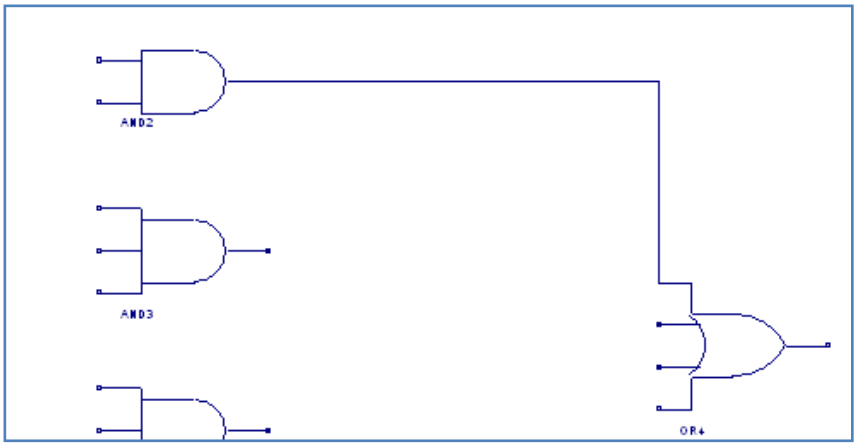


- Now find a 4 input OR gate … you must clear the 'Symbol Name Filter' box and type "or". Pick OR4 and place it in the schematic as shown.

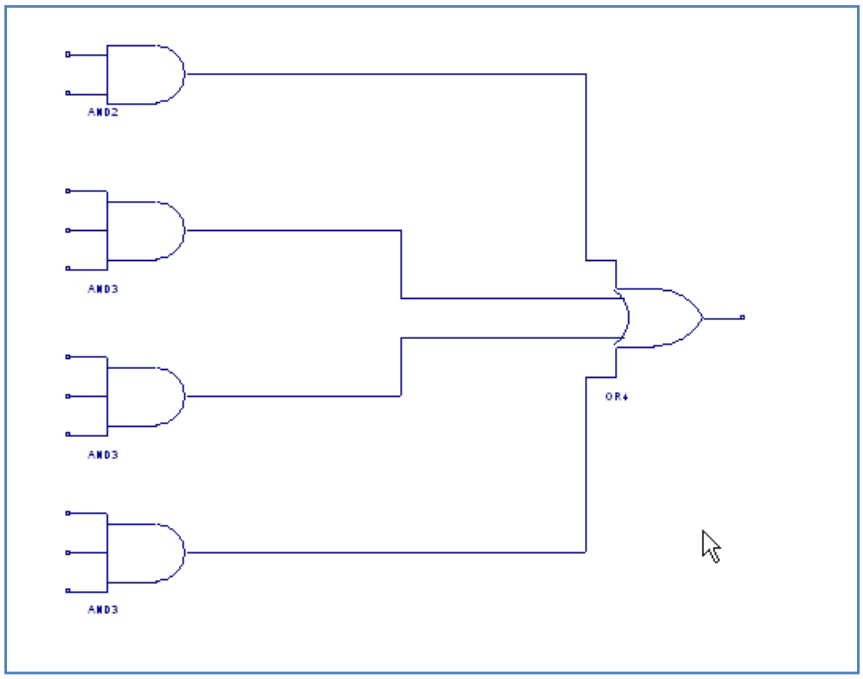


- You are now ready to begin connecting the AND gates to the OR gate.
- Left click on the Wire Drawing tool  in the toolbar (upper-left of {Proj Navigator window}
- Move the cursor to the output of the 2 input AND gate; you should see the connection marks indicating the gate pin where a connection will be made.
- Left click to attach the start of the wire.
- Move your cursor to the top-most input of the OR gate. Again you should see the connection marks that indicate this is where the wire will terminate.

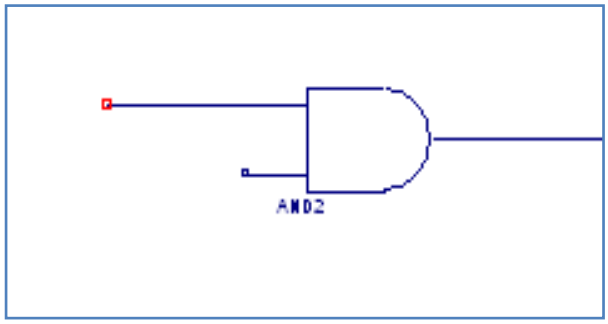




- Left click and the connection is made.
- Repeat until all the outputs of the AND gates are connected to their respective inputs of the OR gate.

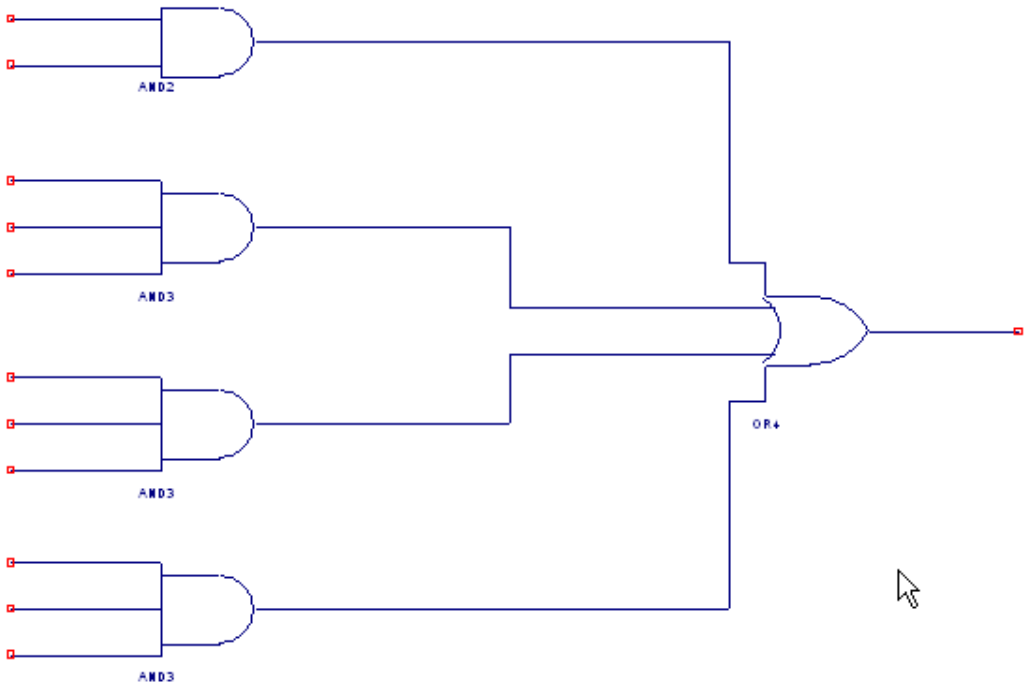


- You circuit should look like the one to the right.



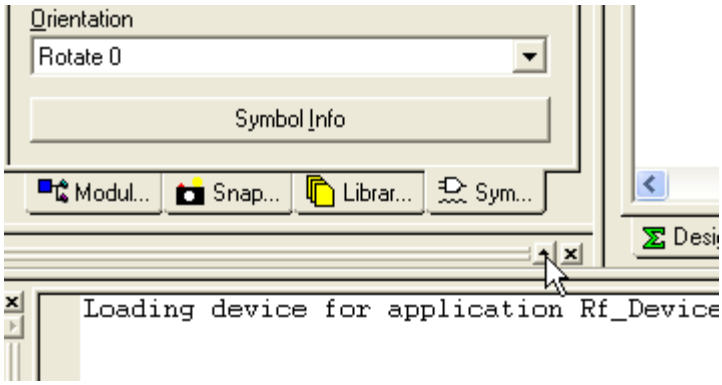
- Now connect a wire stub to each input of the AND gates and a stub to the output of the OR gate. The figure to the right shows the first stub.
- You can repeat this operation by selecting the Selection tool (left-most button) on the tool bar … it is an arrow.
- Select the wire stub you just created; it will turn red indicating what has been selected.
- Press Ctrl-C (copies wire to clipboard).

- Press Ctrl_V to paste the wire in the clipboard, then left-click to place the wire at each of the inputs to the AND gates in turn. Finally, place a wire by Ctrl_V and left-click on the output of the OR gate. Your circuit should now look like the following.

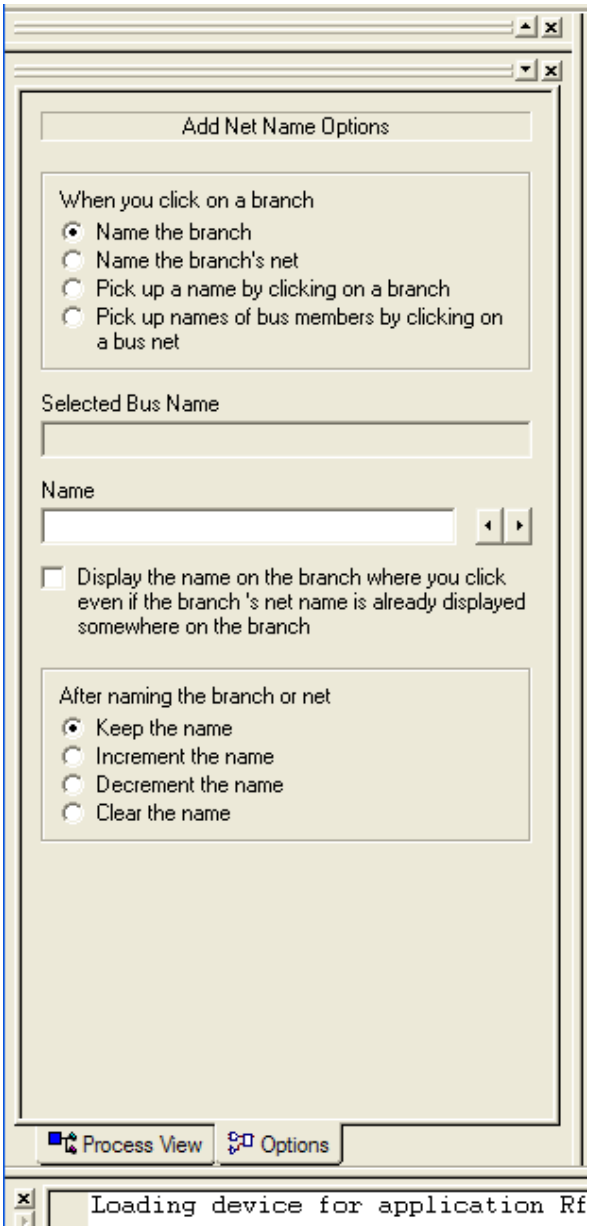


- We will now begin naming the wire stubs just created.
- Click on the Wire Naming tool in the toolbar … just right of the Wiring tool.

- You will need to increase the size of the Processes pane. The Processes pane is minimized below the Sources pane (enlarged). Click on the up-arrow just below the Sources window.

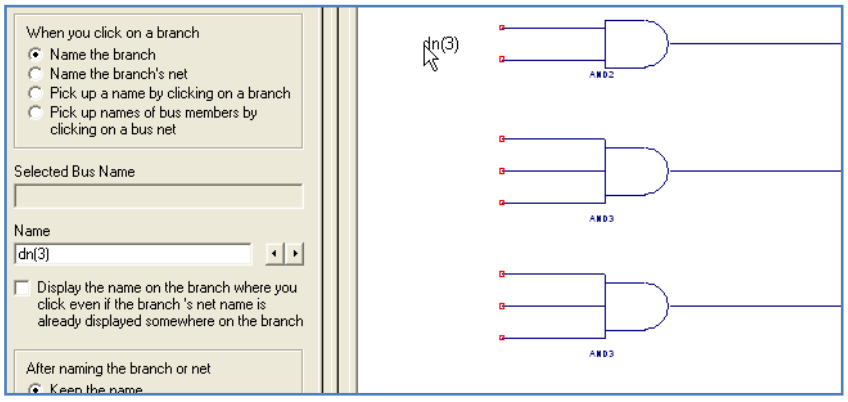


- The Wire Naming tool is manipulated in the Options tab of the Processes pane. See figure to the right.

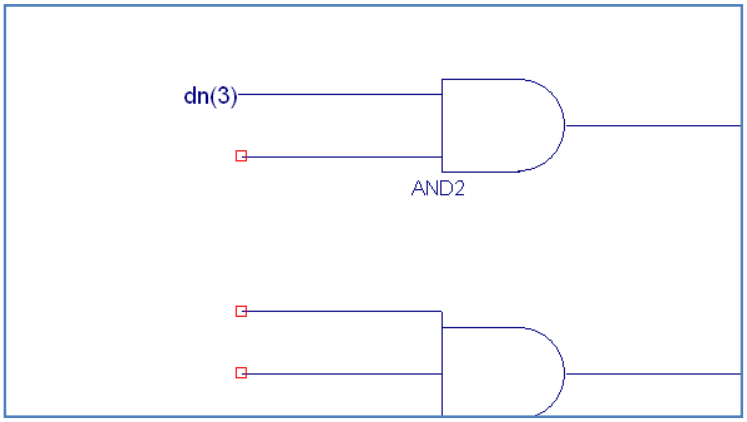


- Click in the Name box and type a name for the wire. The top-most input on the AND2 gate is “dn(3)”

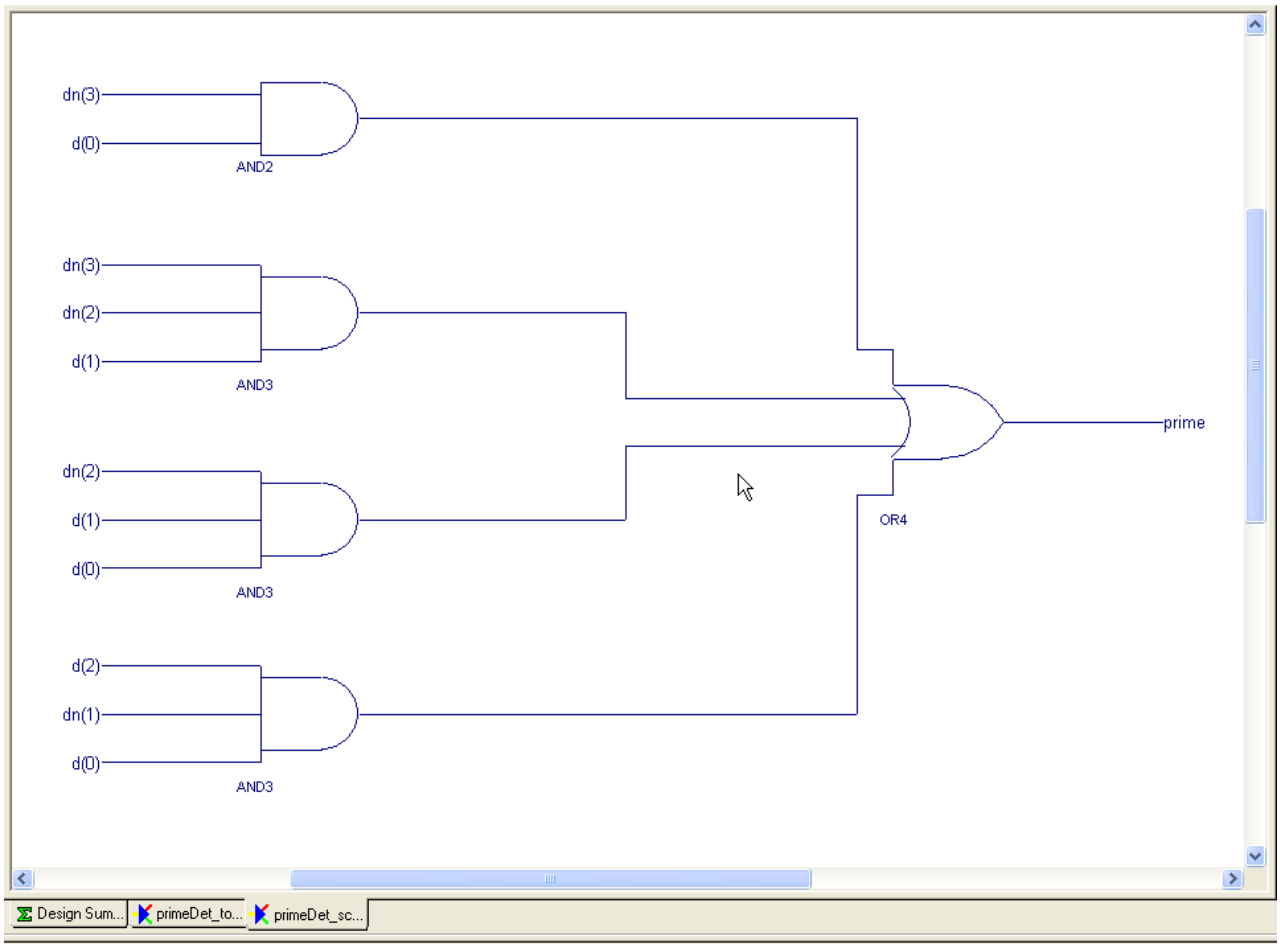
- The name follows the cursor.



- Click on the end of the wire you want to name. The result will be as shown in the figure to the right.

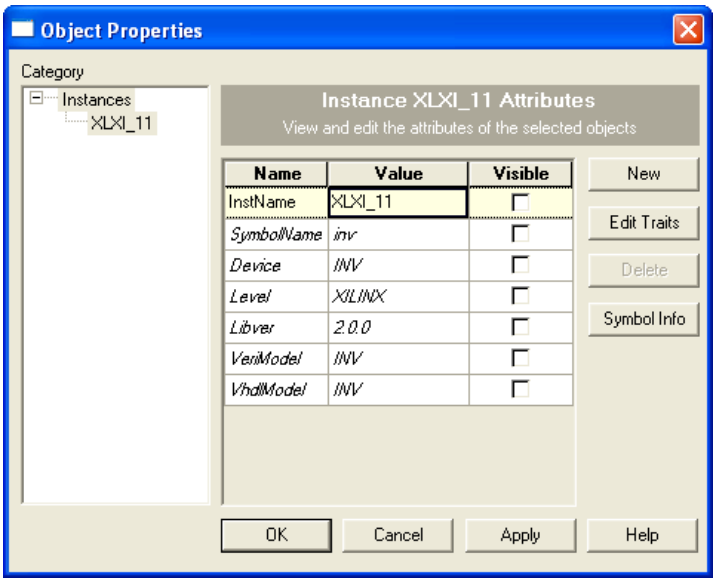


- Continue with this process, of naming wires, until all the wires are named as shown in the following figure. *Note:* there are d(x) and dn(x). Also, note that the output is named "prime".

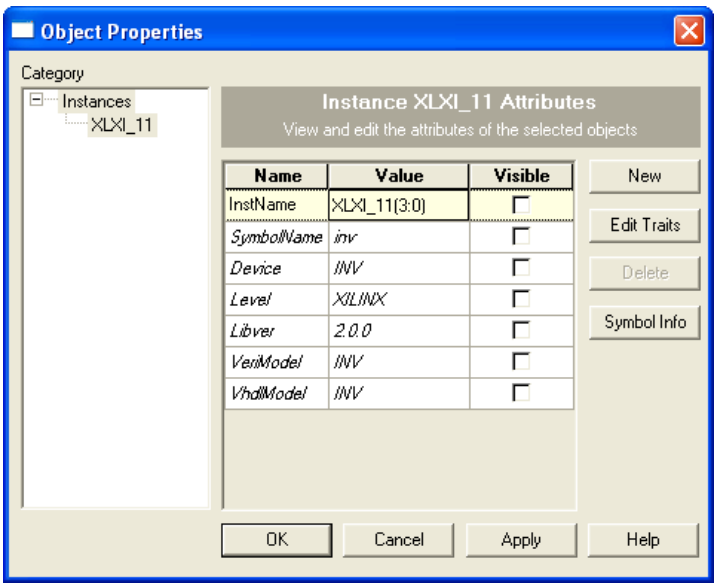


- The bit-vectors of which d(x) and dn(x) are elements need to be defined; i.e., they need a source. The bit-vector dn() equals NOT d(). The final components are inverters. We will be handling these components with vector notation, by making an array of inverters.

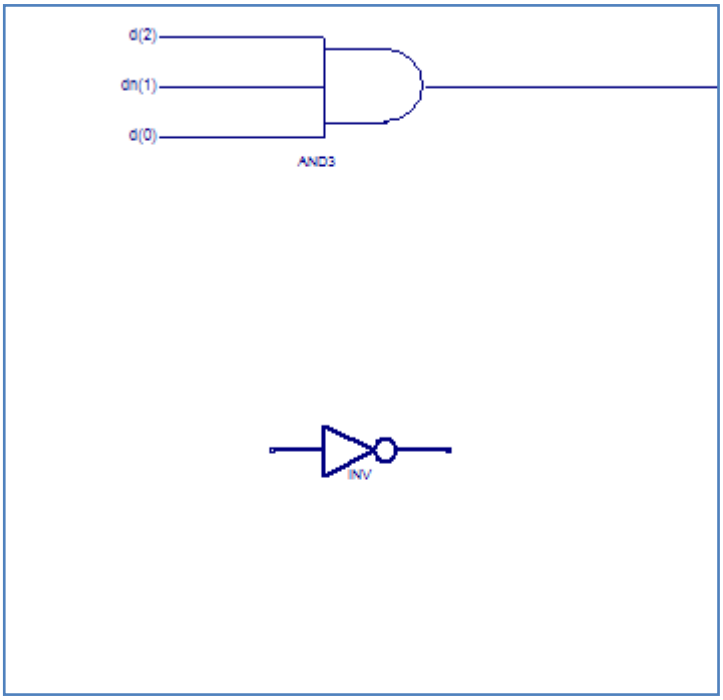
- Find and add a single inverter (inv) to the bottom of the schematic. Double-click on the inverter. The symbols properties dialog box opens, as shown on the right.



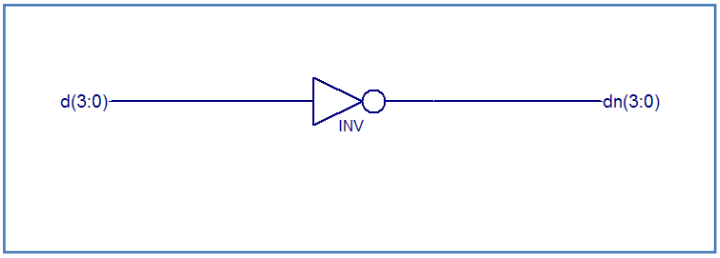
- Change the InstName property to XLXI_11(3:0) … i.e., just add the array notation to the end of the default InstName property. We have just made an array of inverters four wide. We use this for working with busses (bit-vectors). Note: the InstName for your inverter may be different, the important thing is to make it an array.



- The array of inverters will look like a regular inverter, except the lines in the graphic symbol will be thicker, as shown to the right.

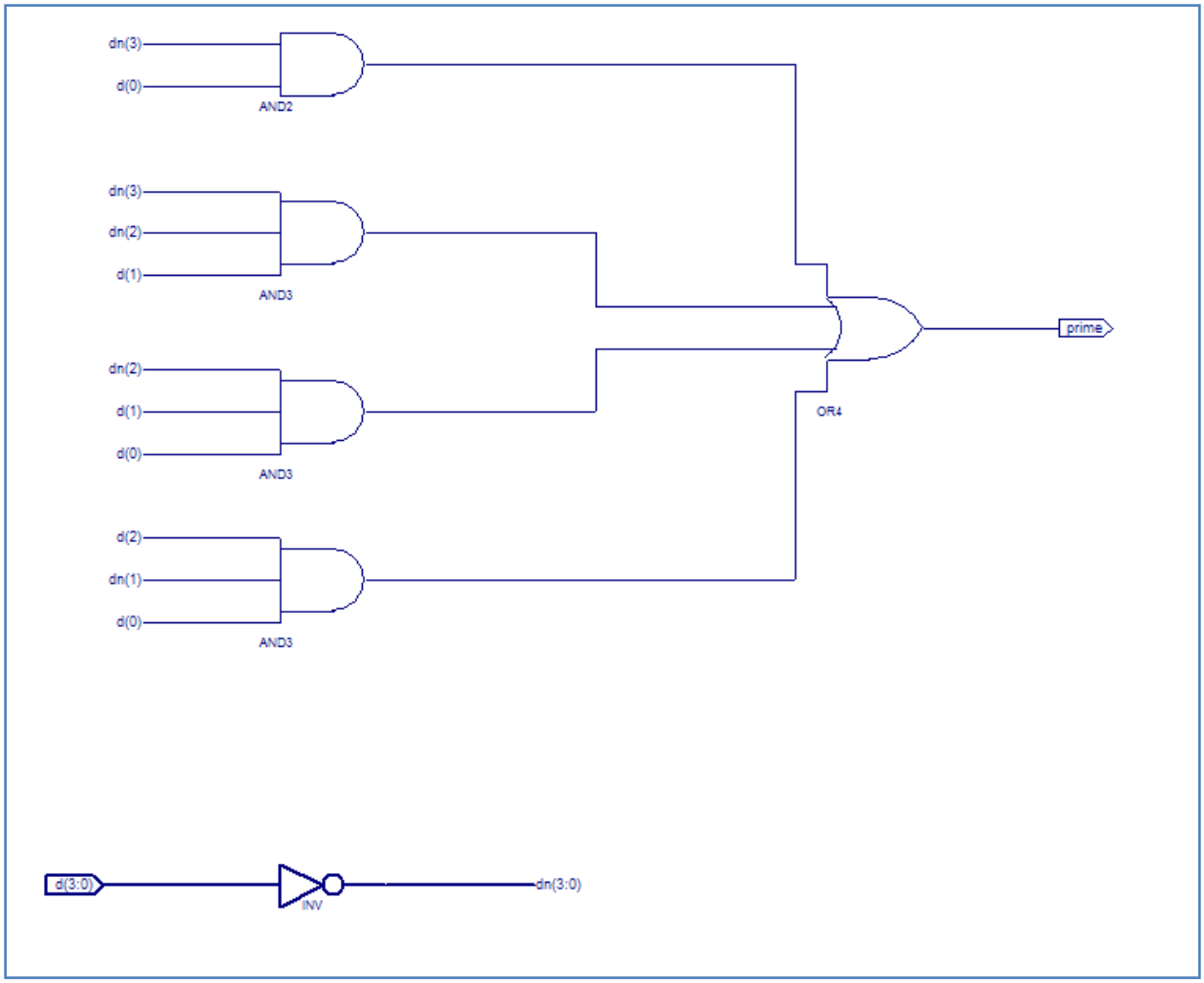


- Add wire stubs to the input and output of the array of inverters. *Note:* that the wires will be thicker in appearance as well. This indicates that the wire is a bus (or bit-vector or array of wires … however, you want to think of them).
- Name the input bus stub for the inverter "d(3:0)" and the output bus stub "dn(3:0)".
- The end result of the inverter circuit is shown in the lower figure to the right.



- Now, the finishing touches for the schematic model. We will add I/O Markers to the inputs and outputs we want to connect to from the outside.
- Click on the IOMarker tool on the toolbar. 
- Move your mouse cursor over the d(3:0) label you just created and click. An input marker will appear over the bus name.
- The IOMarker tool is still selected. So, move the cursor to the output 'prime' connected to the OR gate output and click on the end of the wire.

- The schematic should look like the figure below.

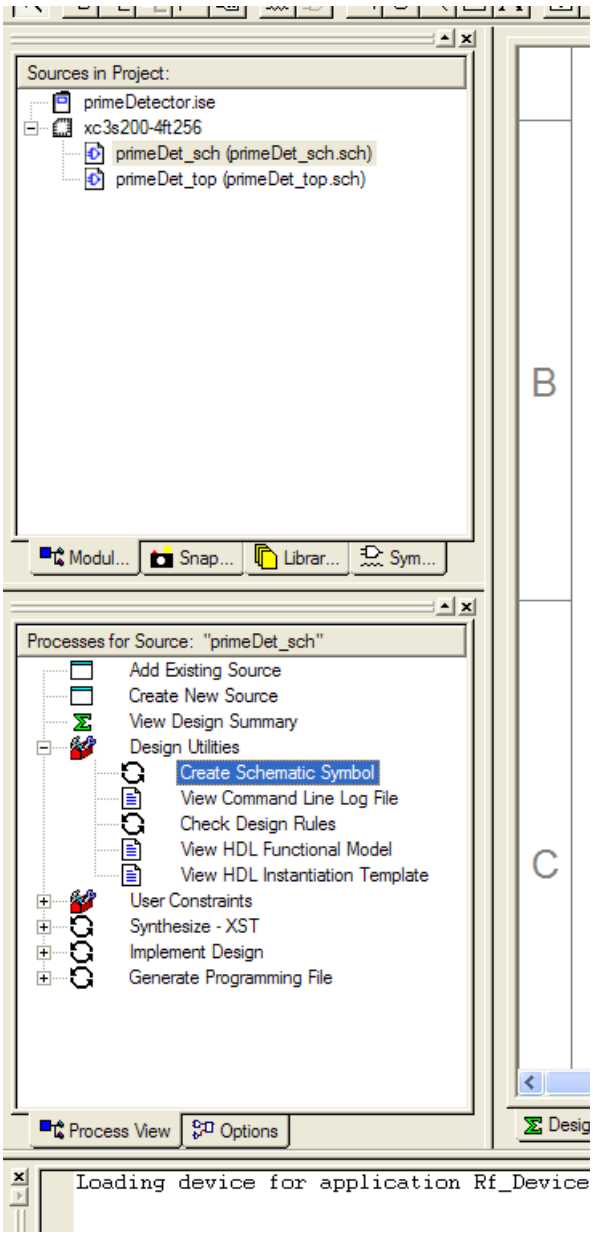


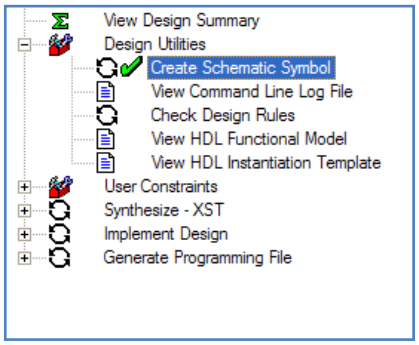
- In order to use this model in the top-level schematic we need to create a schematic symbol. The IO Markers will show up as inputs and outputs on the new symbol. This process is discussed in the next section.

- Click 'Save All' button on the toolbar.

Step 1.2 – Creating a Schematic Symbol for a Digital Logic Model

- Make the Sources pane and the Processes pane both visible.
- Select the schematic labeled primeDet_sch in the Sources pane.
- Click on the Process View tab. Make sure the Design Utilities item is expanded, as shown in the figure to the right.
- Double click on “Create Schematic Symbol”, under Design Utilities.
- A green check mark will appear next to the “Create Schematic Symbol” item in the Processes pane. See figure below. This indicates that the symbol file has been created. The filename should be “primeDet_sch.sym” and is located in the project directory.

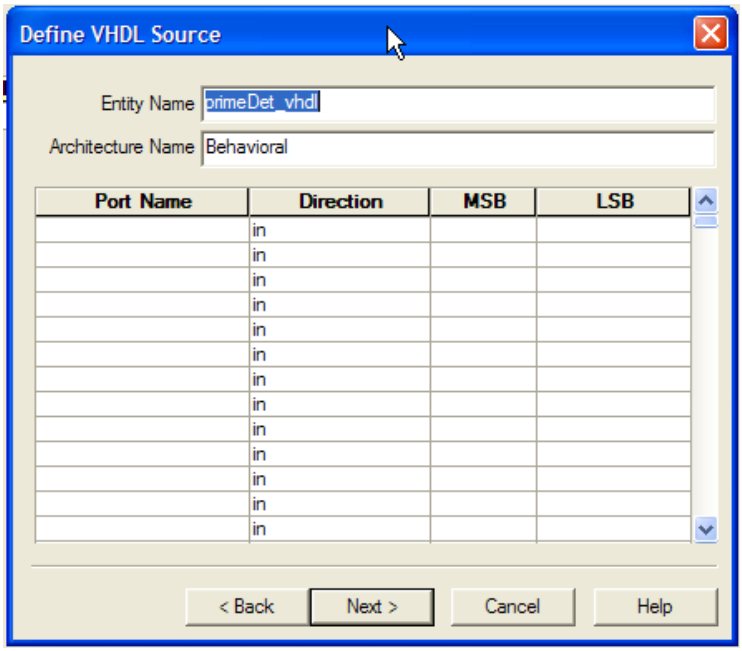




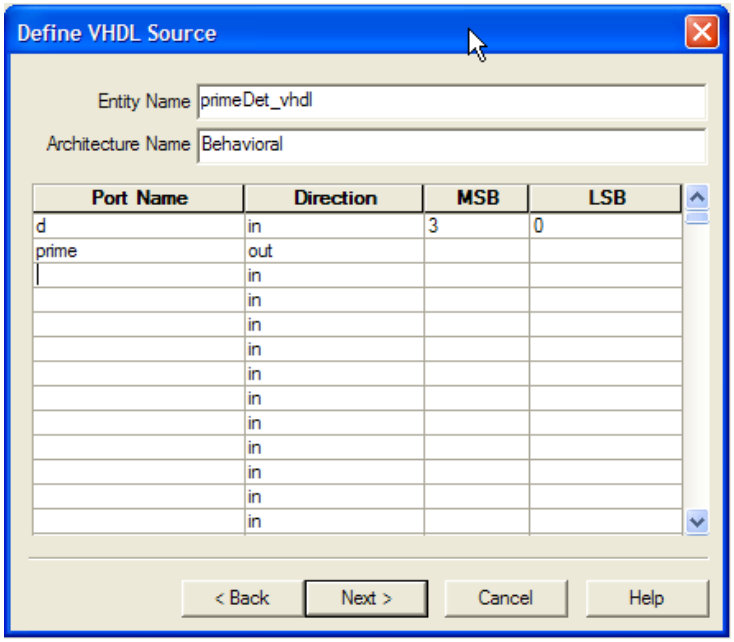
- This process must be repeated if changes to the schematic effect the IO of the model you created.
- You will use this same set of steps to create the schematic symbol for the VHDL model. Just select the source you want to make the symbol for and double-click “Create Schematic Symbol”.

Step 1.3 – Creating a VHDL Model with Xilinx Project Navigator

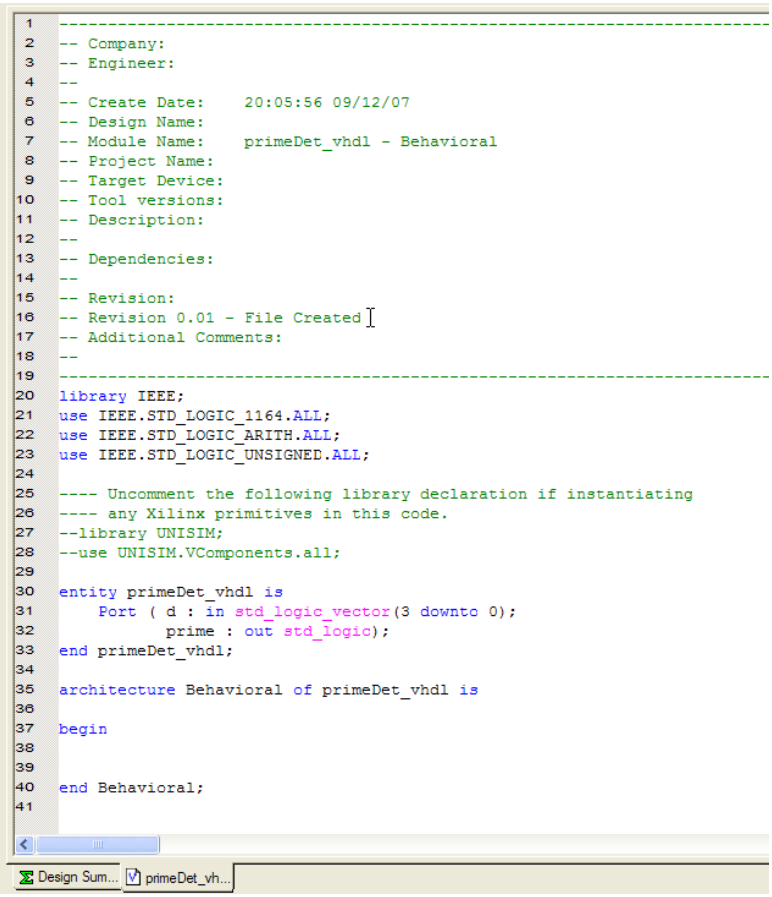
- Click the Project menu option and then select New Source.
- In the New Source window, click on VHDL Module.
- Give the model a filename. In this case use primeDet_vhdl.
- Click Next. The New Source wizard will look the figure on the right.
- Our model with have the same functionality as the schematic model. Thus, the IO is the same.
- The first row under the Port Name column type “d”. Then, tab over to MSB column and type “3”. Tab again to the LSB column and type “0”. This is how to define a bus (a 4-bit bus or bit vector with 4 elements.)
- On the next row type “prime” and select ‘Out’ option under the direction column for that row.



- Click Next, then click Finish.



- You will be presented with a skeleton VHDL model with your IO defined.



```
1  ----------------------------------------------------------------------------------
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    20:05:56 09/12/07
6  -- Design Name:
7  -- Module Name:    primeDet_vhdl - Behavioral
8  -- Project Name:
9  -- Target Device:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 ----------------------------------------------------------------------------------
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity primeDet_vhdl is
31     Port ( d : in std_logic_vector(3 downto 0);
32            prime : out std_logic);
33 end primeDet_vhdl;
34
35 architecture Behavioral of primeDet_vhdl is
36
37 begin
38
39
40 end Behavioral;
41
```

Design Sum... | primeDet_vh...

- Place the cursor on line 39 of the figure above and tab over once, then type the following text.

```
with d select
   prime <= '1' when "0001",
            '1' when "0010",
            '1' when "0011",
            '1' when "0101",
            '1' when "0111",
            '1' when "1011",
            '1' when "1101",
            '0' when others;
```

- The bottom of the VHDL model should look like the following figure.

```
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  ---- Uncomment the following library declaration if instantia
26  ---- any Xilinx primitives in this code.
27  --library UNISIM;
28  --use UNISIM.VComponents.all;
29
30  entity primeDet_vhdl is
31      Port ( d : in std_logic_vector(3 downto 0);
32             prime : out std_logic);
33  end primeDet_vhdl;
34
35  architecture Behavioral of primeDet_vhdl is
36
37  begin
38
39     with d select
40        prime <= '1' when "0001",
41                 '1' when "0010",
42                 '1' when "0011",
43                 '1' when "0101",
44                 '1' when "0111",
45                 '1' when "1011",
46                 '1' when "1101",
47                 '0' when others;
48
49
50  end Behavioral;
51
```

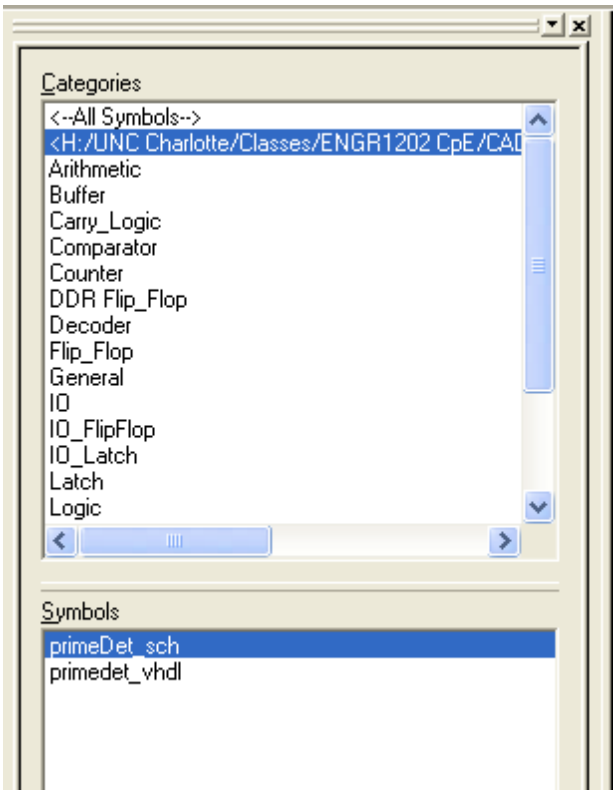
- Save all.

- Create the schematic symbol for primeDet_VHDL using the procedures described for the schematic above.

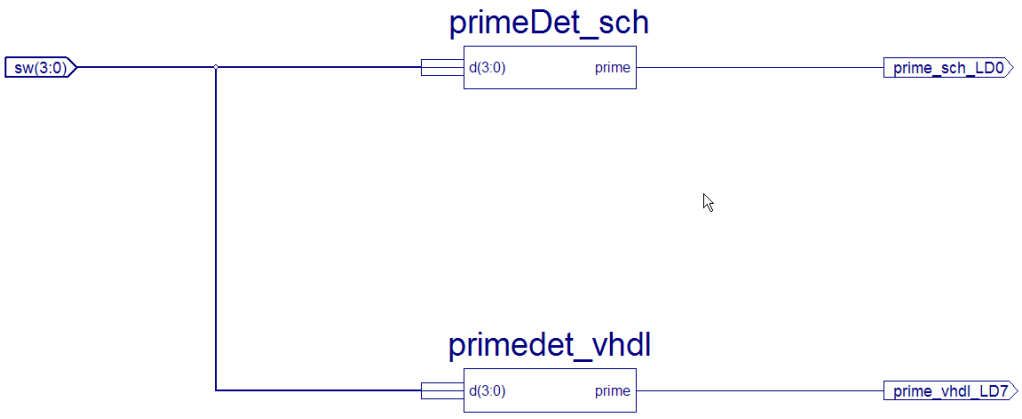
- Close all the tabs in the Working pane by clicking on the small x below the main window close button (in this case the red x). Repeat until all the tabs are closed.

Step 1.4 – Top-level Schematic Model

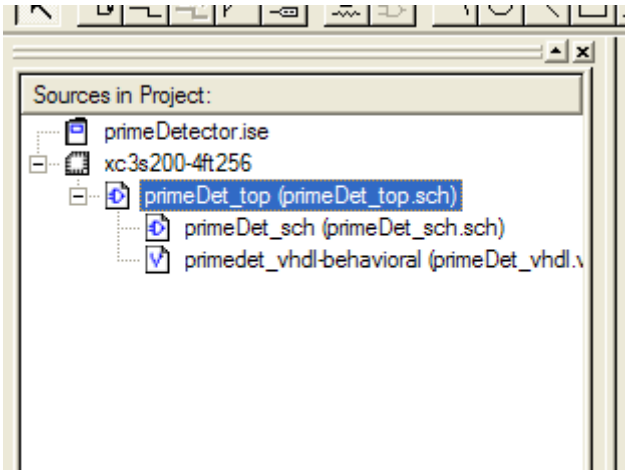
- Double-click on the primeDet_top item in the Sources pane to open the top-level schematic.
- Click on the Symbols tab in the sources window. Expand the size of the processes pane to get a better view of the contents.
- In the Categories box select the project working directory.
- You should now see the symbols you created in the Symbols box.



- Select each symbol in turn and place them in your schematic. See the figure below.
- Wire the components up as shown in the figure below.
- Give the wires and busses the names shown.
- Add the IO Markers as shown.



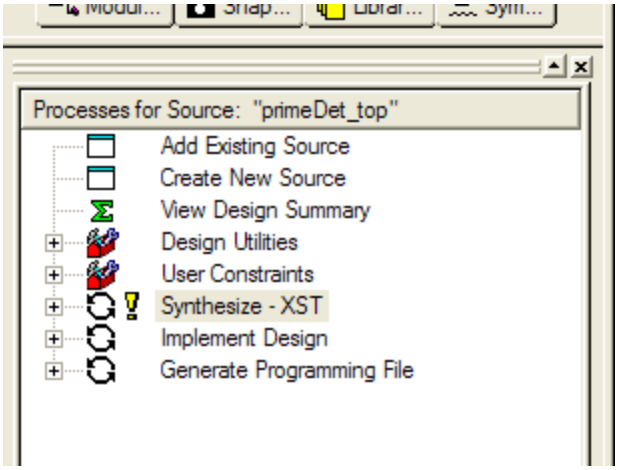
- Save All.
- There is no need to make a symbol for the top-level schematic.
- After you have saved the project the two models you created will show up as levels below the top-level schematic in the Sources pane.



- Making a model then placing it into a schematic is referred to as a Hierarchical structure. You can nest models as deep as you please. The idea is to break the design into manageable pieces.

Step 2 – Synthesize the Circuit.

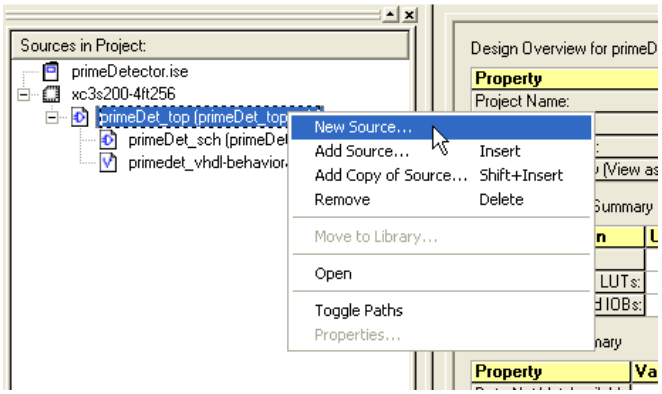
- Select the top-level schematic in the Sources pane.
- Select the Process View tab in the Processes pane.
- Double click on the "Synthesize – XST" item in the Processes pane. Text will scroll by in the Console pane at the bottom of the Project Navigator window.
- If the synthesis went ok, you should see a green check mark next to the "Synthesize – SXT" item in the Processes pane.
    - A yellow exclamation point means a warning was given. The models will work; however, the CAD software is trying to warn you about a potential hazard.
    - In this design we get a yellow exclamation point. Check the warning to make sure it is not hazardous to the functionality of the design.
    - To look at the warnings, click on the Warnings tab in the Information pane. The warnings will be displayed.



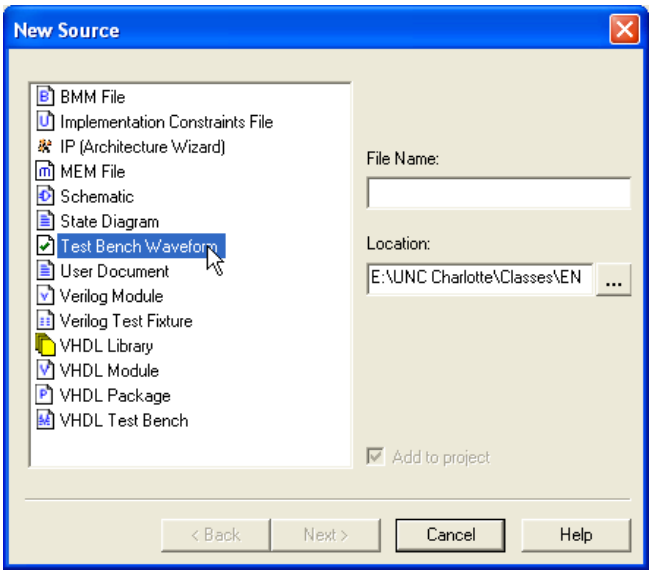
    - In this case we have a warning that is not hazardous to the model functionality. An unconnected source pen. The software is thinking (rather the software designers) that you went to the trouble of creating a source and you are not using it … must be a problem. Well it's not this sometimes happens when using busses. Be assured that this will not affect your circuits' operation.
    - If a red X appears then the synthesizer could not complete the task. Errors can be viewed in the Errors tab in the Information pane.
    - If the information in the Warning or Error tabs is not sufficient to determine the corrective action (if one is required), then click on the Console tab and scroll through the text that was generated during the synthesis process.
- Once the circuit synthesizes, we will move on to the Simulation (or test) phase of the process.

Step 3 – Simulation (Test Phase)

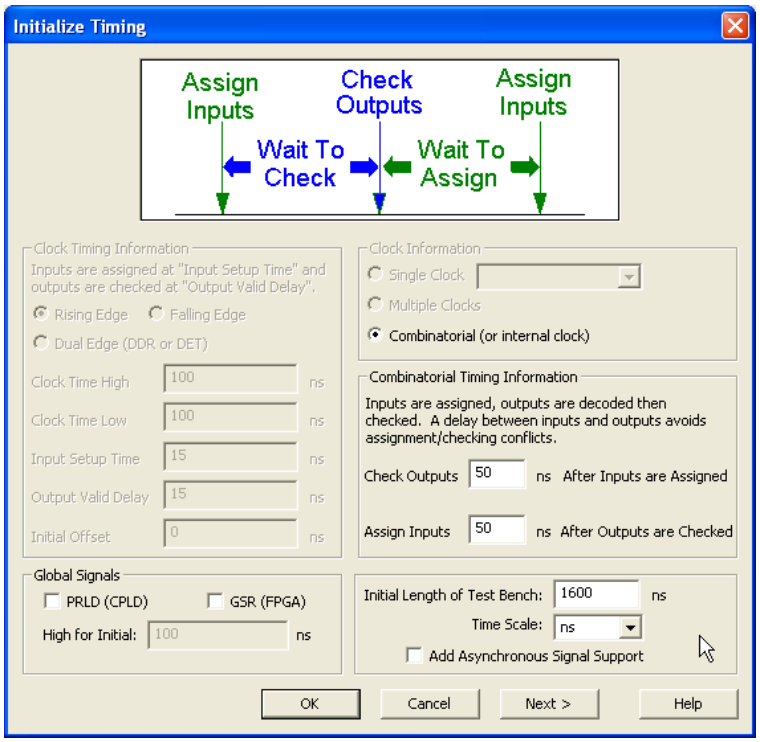
- *Step 3a* – Stimulus Setup (Test Bench creation)
- Select the top-level schematic in the Sources pane.
- Right click on the top-level schematic and select 'New Source'
- The New Sources wizard appears.



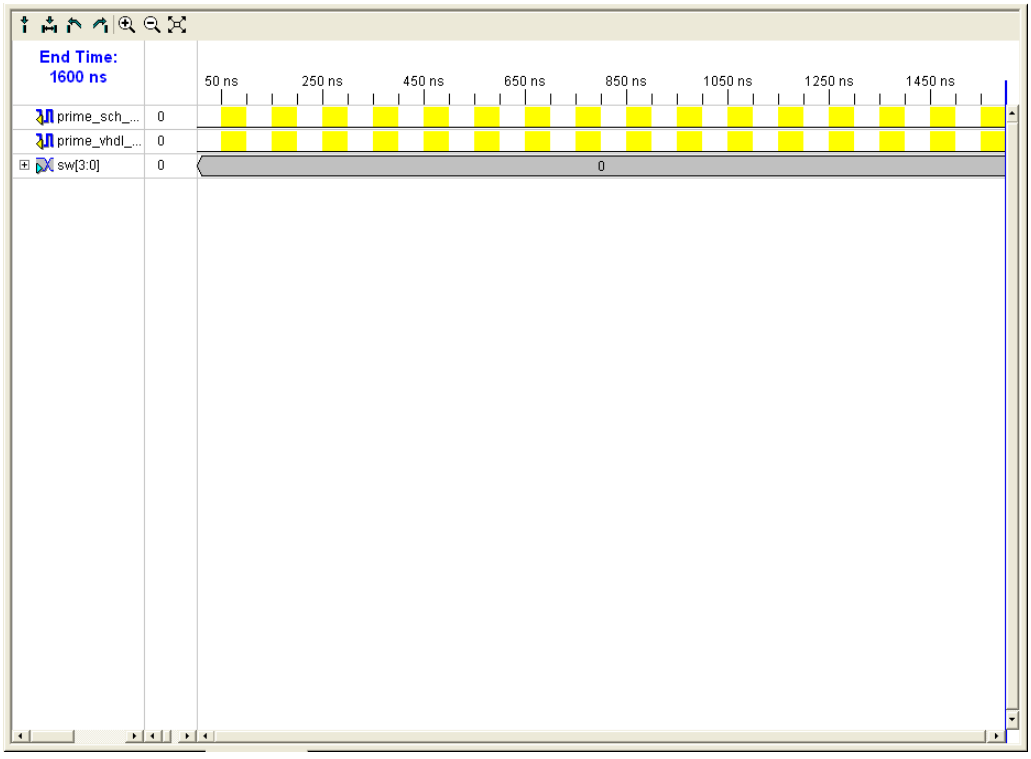
- Select Test Bench Waveform in the list box on the left.
- Type the file name → primeDet_top_tb
- Click Next.
- A list of the possible sources' for the project are displayed. Select the top-level schematic ("primeDet_top").
- Click Next
- Click Finish



- A new window appears. This window is for setting up when inputs and outputs will be changing.
- Change the 'Initial Length of Test Bench:' setting to 1600 ns.
- By default 'Check Outputs' and 'Assign Inputs' fields should be set to 50 ns.
- All check boxes should be unchecked.
- Click Ok.
    - ***Caution:*** If you click Next here, you will be given a window with a question that cannot be answered; i.e., you will get stuck.



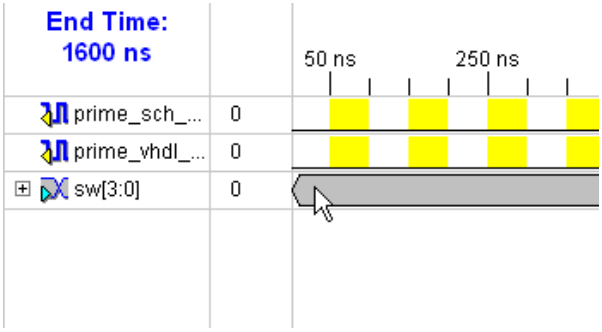
- If all goes well you should have a new source that will automatically open in the Working pane. See figure below.



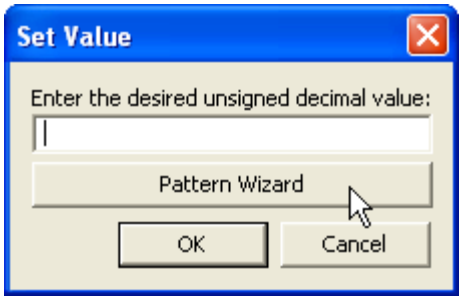
- The first two lines are the outputs (denoted by the yellow arrows pointing left) and the third line is the input bus (denoted with a blue arrow pointing right).

- Setup the inputs to include all of the possible input combinations. This is done by counting from 0 to 15 in binary.

- Click on the gray area of the sw[3:0] input bus, just to the left of the 50ns mark.



- The 'Set Value' box is now presented. Click on the Pattern Wizard button

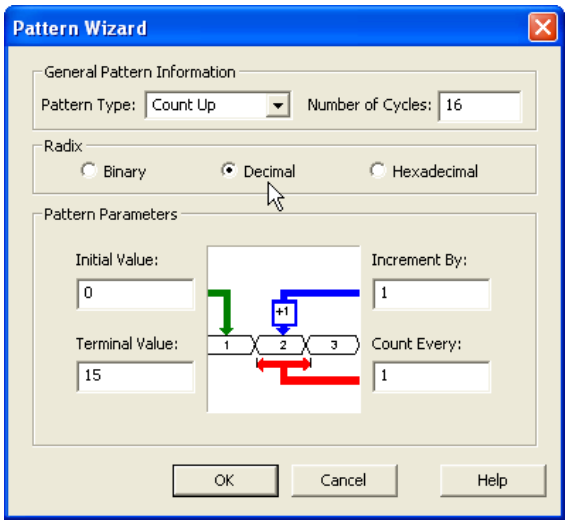


- In the Pattern Wizard window, select the 'Count Up' option from the Patten Type drop down list.

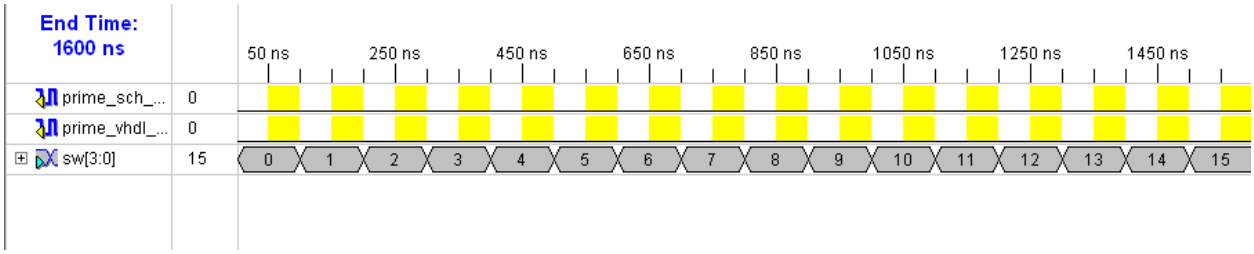
- Set 'Number of Cycles:' to 16.

- Leave the remaining options as default.

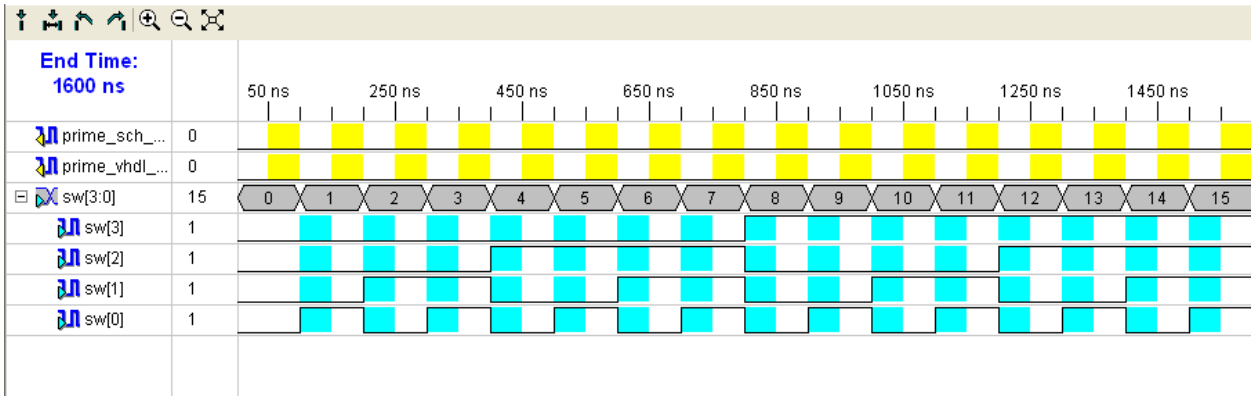
- Click Ok



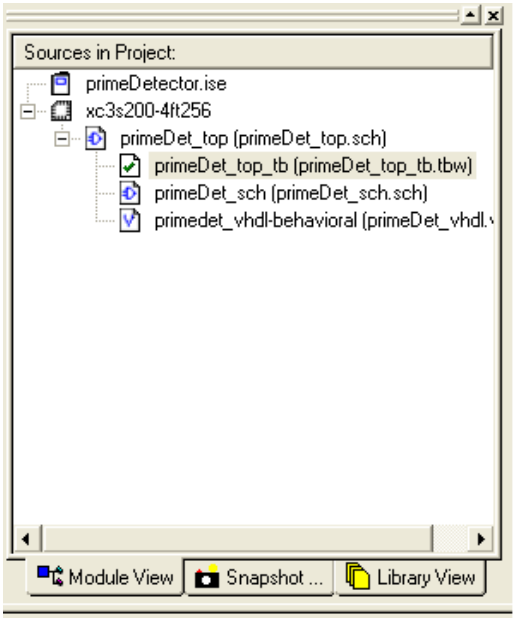
- The Test Bench waveform file will now appear as shown below.

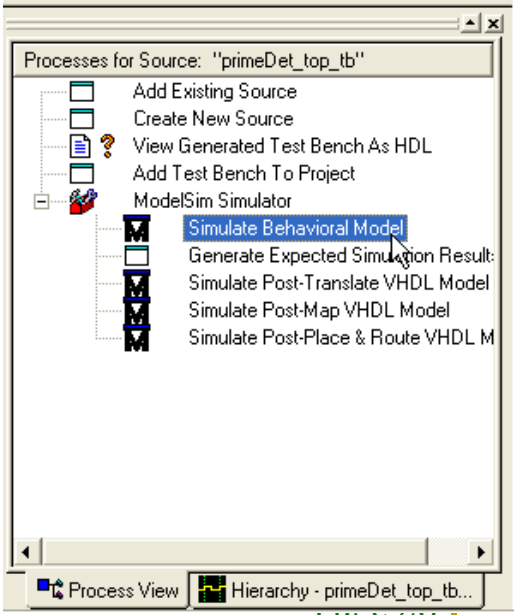


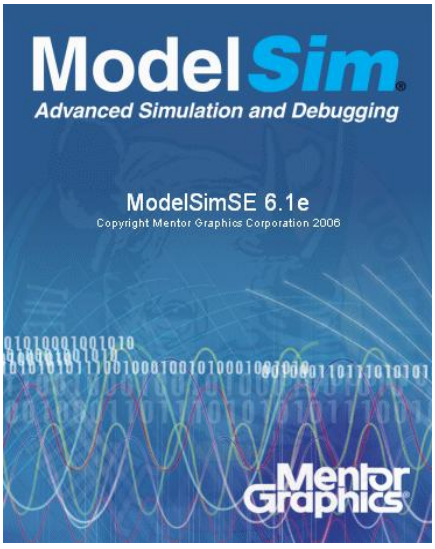
- If you expand the input sw[3:0] (click on the + symbol next to the input) the input bus will expand to show the binary counting for each signal in the bus.



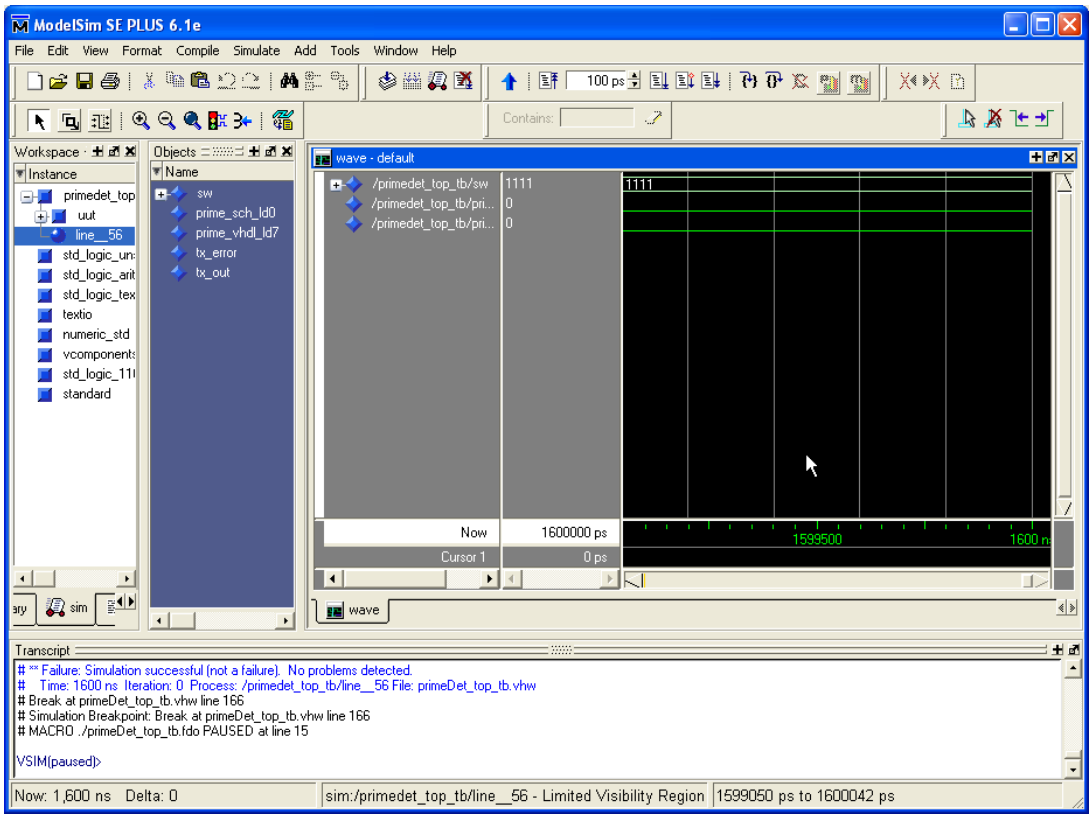
- Note: changing the values of the output signals will not have an effect on the test. The simulator will be calculating these values.
- Click the Save All button on the toolbar.
- Click on the 'Process View' tab in the Processes pane.
- Select the test bench waveform file in the Sources pane.
- Double click on the Simulate Behavior Model item in the Processes pane.
- The simulator application (ModelSim) will start up.





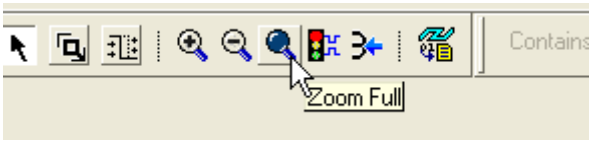


- The Modelsim window will look like the following:



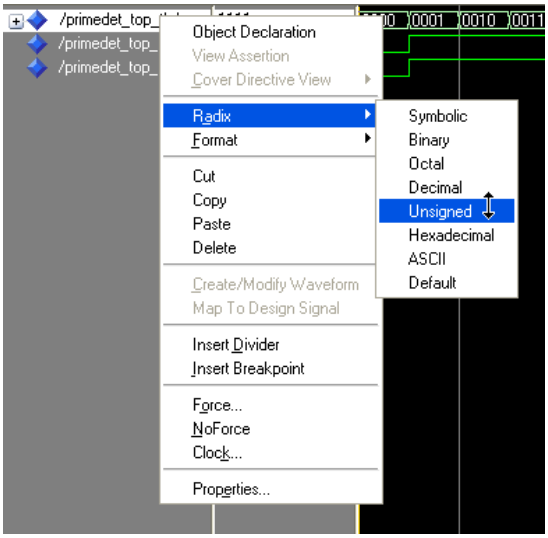
- Notice the time scale at the bottom. The view defaults to a very small slice of the end of the simulation results.

- Click on the 'Zoom Full' button on the toolbar to see all the simulation results.

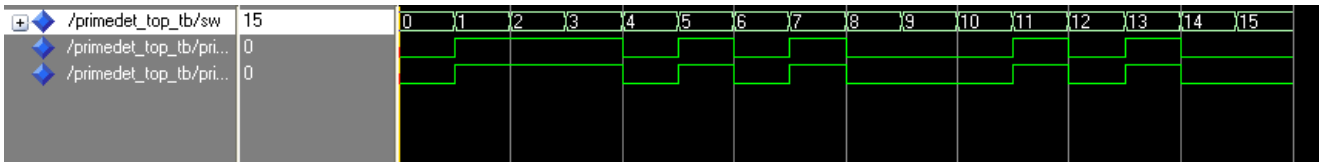


- The results should now appear as follows:

- To make these results easier to read right click on the input name on the left and select Radix → Unsigned. See figure for details.



- The following figure should be seen.



- Verify your circuit is working by observing the outputs for each of the input combinations.
    - The outputs should be high for the prime numbers 1, 2, 3, 5, 7, 11 & 13.
    - The outputs should be low for non-prime numbers 0, 4, 6, 8, 9, 10, 12, 14 & 15.

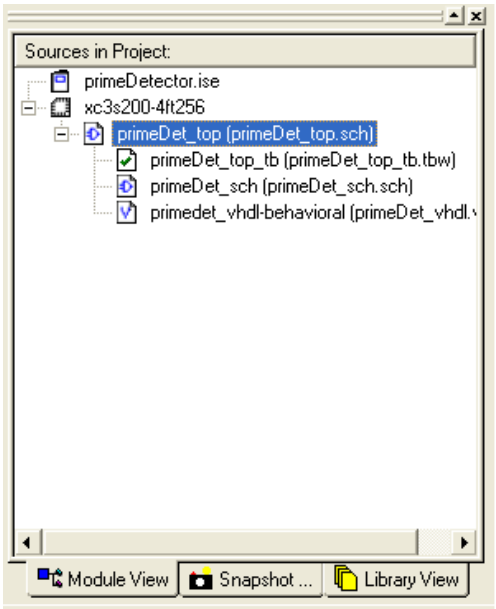
- If errors are encountered, then close Modelsim and trace through the schematics. Trace through the blocks associated with the errors. Re-Synthesize the whole circuit and start the simulator up again, check your outputs as indicated above. Repeat if necessary until the correct outputs are obtained.

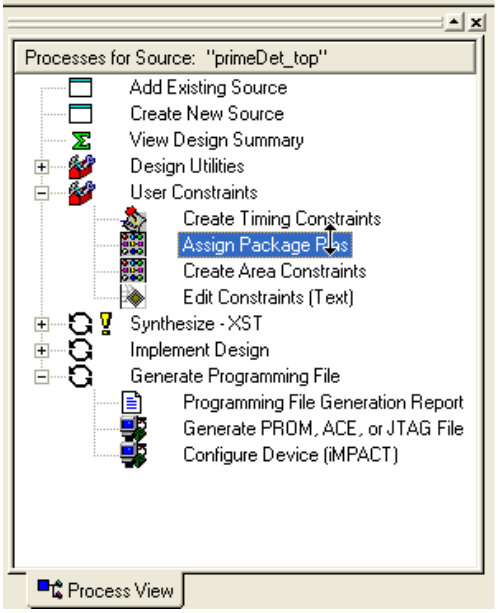
Step 4 - Loading the Models into the FPGA

- The FPGA board has the FPGA pins assigned when the board was designed. We must setup our project to use the same pins as the board.

- For the Spartan 3 Starter Kit and the Nexys2 boards, the pin numbers of the FPGA are silkscreened on the board near the I/O device (i.e., switches (SWx), LEDs (LDx), etc.)

    - If a different board is used you must read through the documentation to determine where things are connected.





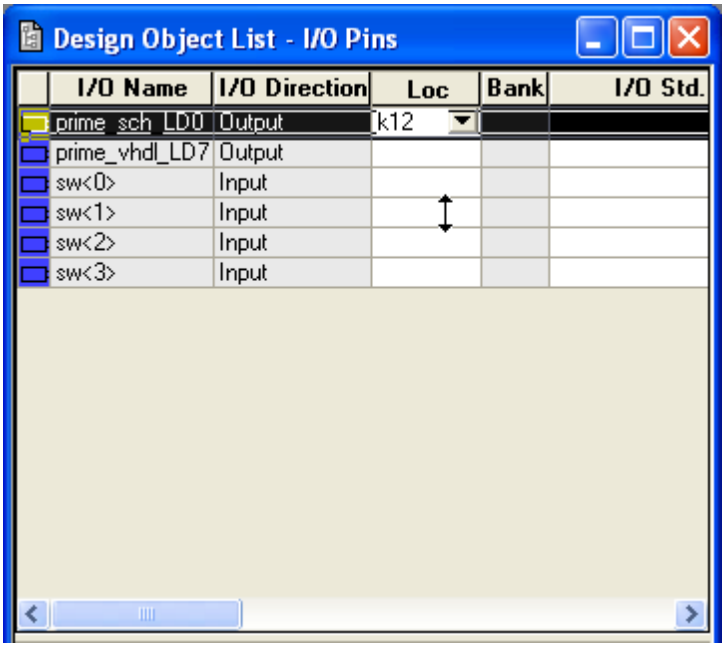
- ***Step 4a*** - To assign pins within the project select the top-level schematic in the Sources pane.

- Double click on the 'Assign Package Pins' item (under User Constraints item) in the Processes pane.

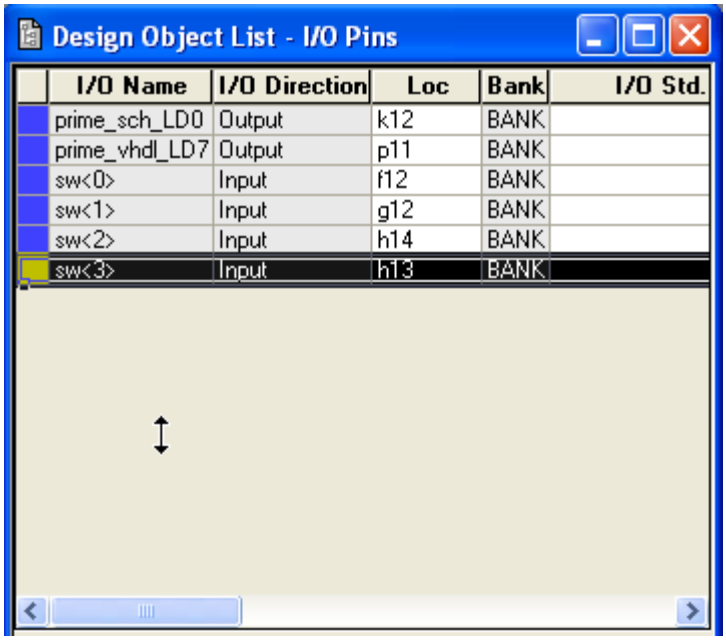
- If this is the first time you selected 'Assign Package Pins' you will see a dialog box that ask if you want to add the file to the project … paraphrased. Click on Yes.

- Another application will open called "Xilinx PACE". This is a constraints editor program. In this case, we are assigning pins.

- The pin assignments are:
    - Spartan3 Starter Kit:
        - Prime_sch_LD0 → K12
        - Prime_vhdl_LD7 → P11
        - sw<0> → F12
        - sw<1> → G12
        - sw<2> → H14
        - sw<3> → H13
    - Nexys2:
        - Prime_sch_LD0 → J14
        - Prime_vhdl_LD7 → R4
        - sw<0> → G18
        - sw<1> → H18
        - sw<2> → K18
        - sw<3> → K17



- In the 'Design Object List – I/O Pins' pane, (setup like a spreadsheet) click on the prime_sch_LD0 under the Loc field and type K12.
- The assignment of the prime_sch_LD0 pin is made. (Shown for the Spartan3 Starter Kit boards.)
- Continue until all pins are assigned as shown in the list above.



- The assignments are as shown to the right. (Shown for the Spartan3 Starter Kit)

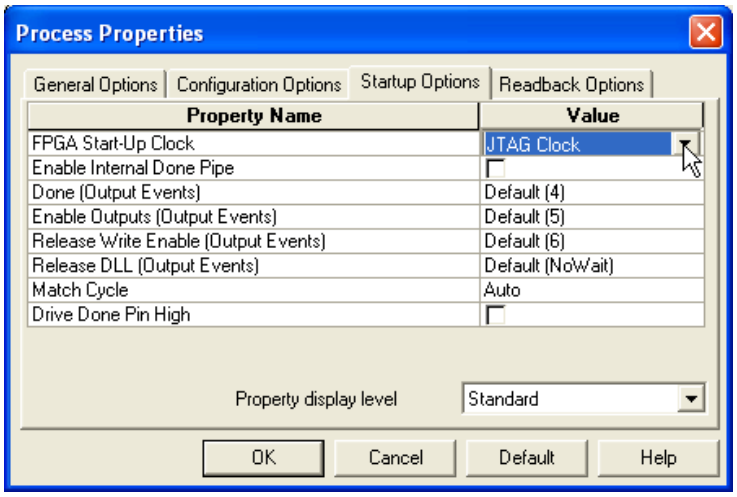
- Click on the Save button on the toolbar of XilinxPACE.
- A dialog will appear. Use the defaults by clicking Ok.
- Now close the XilinxPACE application.

- The .ucf file is added to the project under the top-level schematic.

- ***Step 4b*** – Creating the .bit file. The .bit file will be loaded into the FPGA to achieve the desired operation.

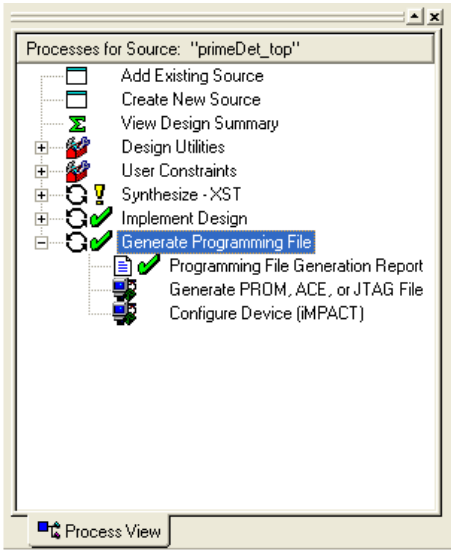
- Select the top-level file in the Sources pane.

- Right click on the 'Generate Programming File' item in the Processes pane and select 'Properties'.
- Select the 'Startup Options' tab.
- Change the 'FPGA Startup Clock' option to 'JTAG Clock'.
- Click Ok.



- Double-click on the 'Generate Programming File' item in the Processes pane.

  - This process takes a while.
  - The 'Implement Design' and 'Generate Programming File' items in the Processes pane should receive a Green checkmarks; as shown in figure to the right.

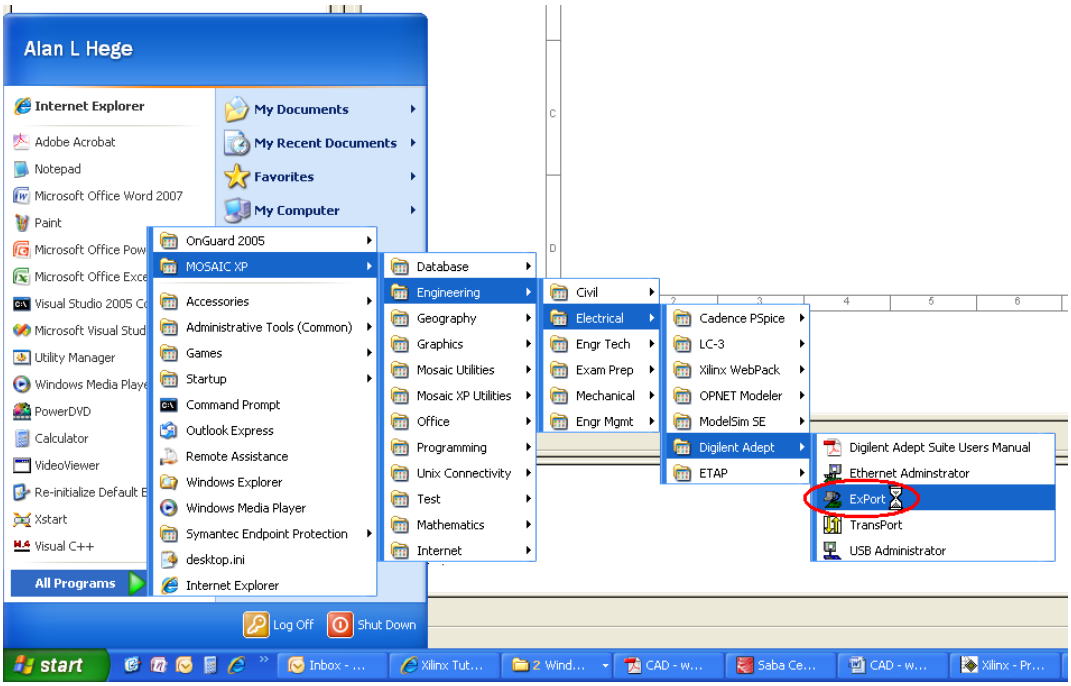


- ***Step 4c*** – Configuring the FPGA Nexys2 boards shown here (Spartan3 Starter Kits – see tutorial list for instructions)

- Connect the USB cable to the Nexys2 board.

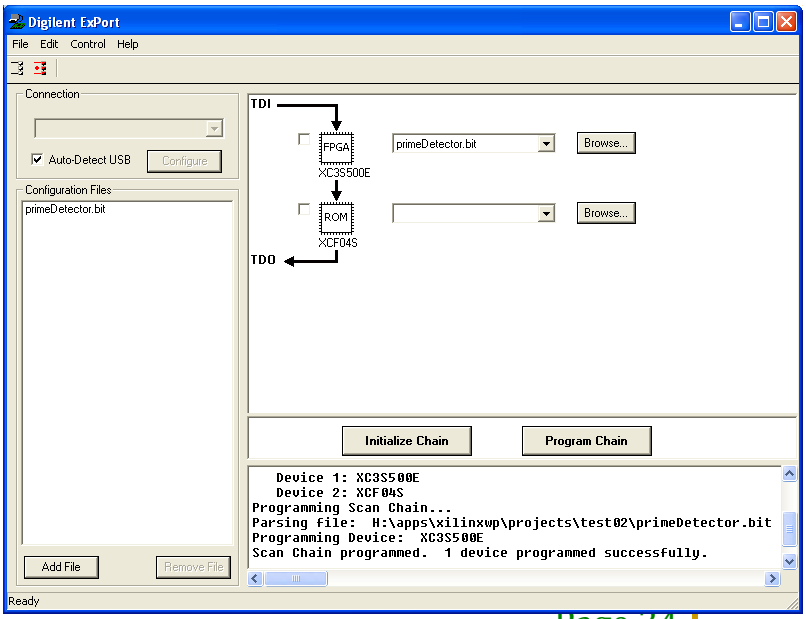
- Turn on the board with the power switch (upper left corner of board.) Switch in the up position. Power LED will illuminate.

- Open up an application called Export (Digilent, Inc.).



- Click on the *Initialize Chain* button within the application (center toward bottom.)

- Click on browse (next to FPGA) to find your .bit file. It will be located in your project directory.
(click yes on the pop-up if it occurs)



- Click on Program Chain to configure the FPGA.
- Once the Done light comes on the FPGA is configured to implement your digital circuit.
- Once you have finished the verifying your design, close the ExPort application.
- The process is complete and the student should be ready for the operational testing.

Step 5 – Operational Testing

- The right-most four switches are assigned as the inputs to the Prime Number Detector circuit.
- The outputs are two LEDs labeled LD0 for the schematic model and LD7 for the VHDL model.
- Both outputs should agree.
- For the switches down is a '0' and up is a '1'.
- Go through all the binary combinations checking that each matches the expected values given below.

Value (decimal)	d(3:0) (binary)	Expected LD0	Actual LD0	Expected LD7	Actual LD7
0	0000	Off		Off	
1	0001	On		On	
2	0010	On		On	
3	0011	On		On	
4	0100	Off		Off	
5	0101	On		On	
6	0110	Off		Off	
7	0111	On		On	
8	1000	Off		Off	
9	1001	Off		Off	
10	1010	Off		Off	
11	1011	On		On	
12	1100	Off		Off	
13	1101	On		On	
14	1110	Off		Off	
15	1111	Off		Off	

- Example for the value of 13 the switches should be set as SW0 → UP; SW1 → DOWN; SW2 → UP; & SW3 → UP. Both LD0 and LD7 will be illuminated.