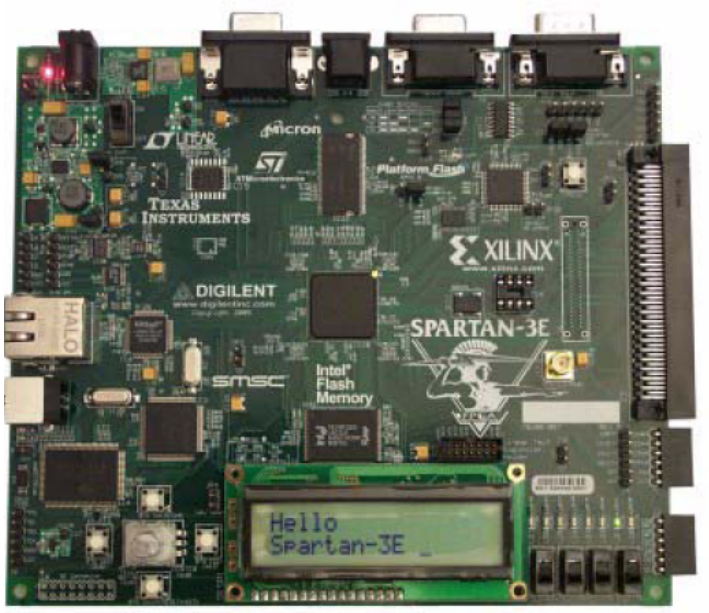
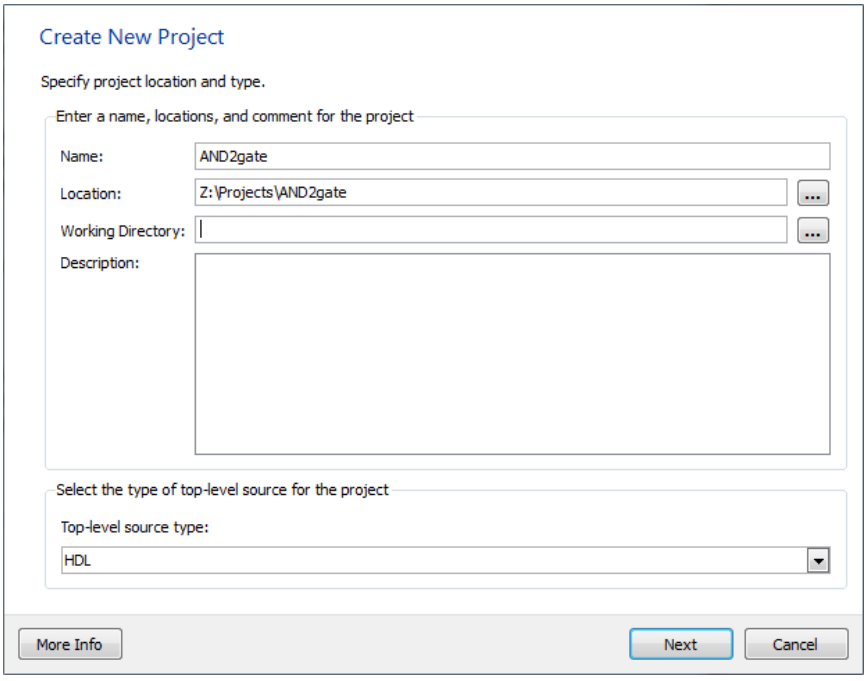
Xilinx ISE 13.2 Synthesis Tutorial

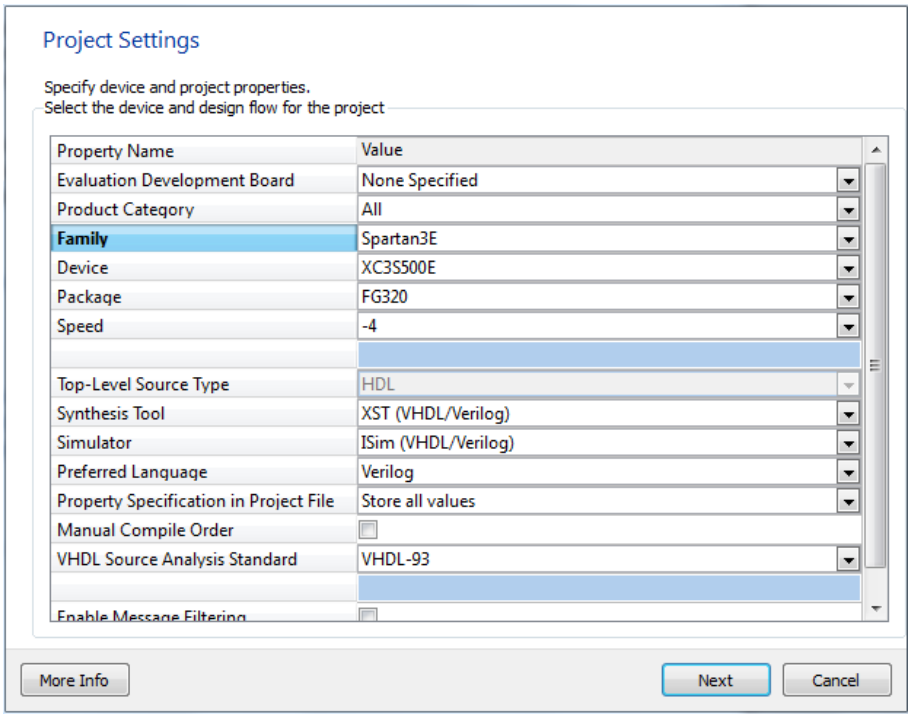
The following tutorial provides a basic description of how to use Xilinx ISE 13.2 to create a simple 2-input AND gate and synthesize the design onto the Spartan-3E Starter Board pictured below. This tutorial should also work with the Xilinx WebPACK that can be downloaded from Xilinx website.



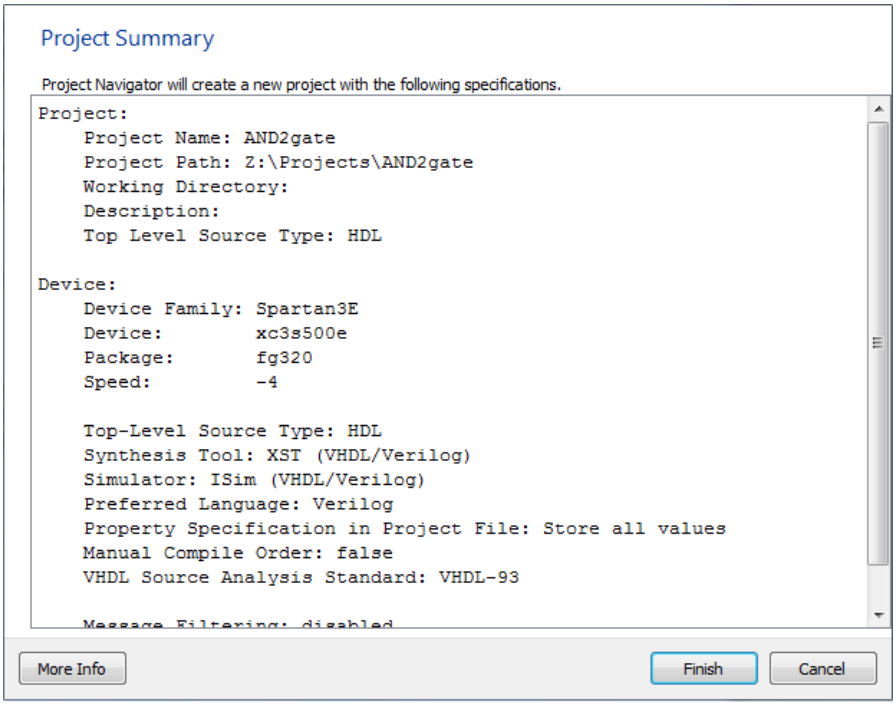
1. Start Xilinx ISE Project Navigator
2. Create a new project
   - Click on *File*, then choose *New Project* on the drop down menu
   - Enter your project name, in this case the project is called “AND2gate”
   - Choose your project location, this project is stored at “Z:\Projects\AND2gate”
   - Leave the working directory entry blank.
   - Choose *HDL* as the source type from the *Top-Level Source Type* menu.
   - Click *Next* button



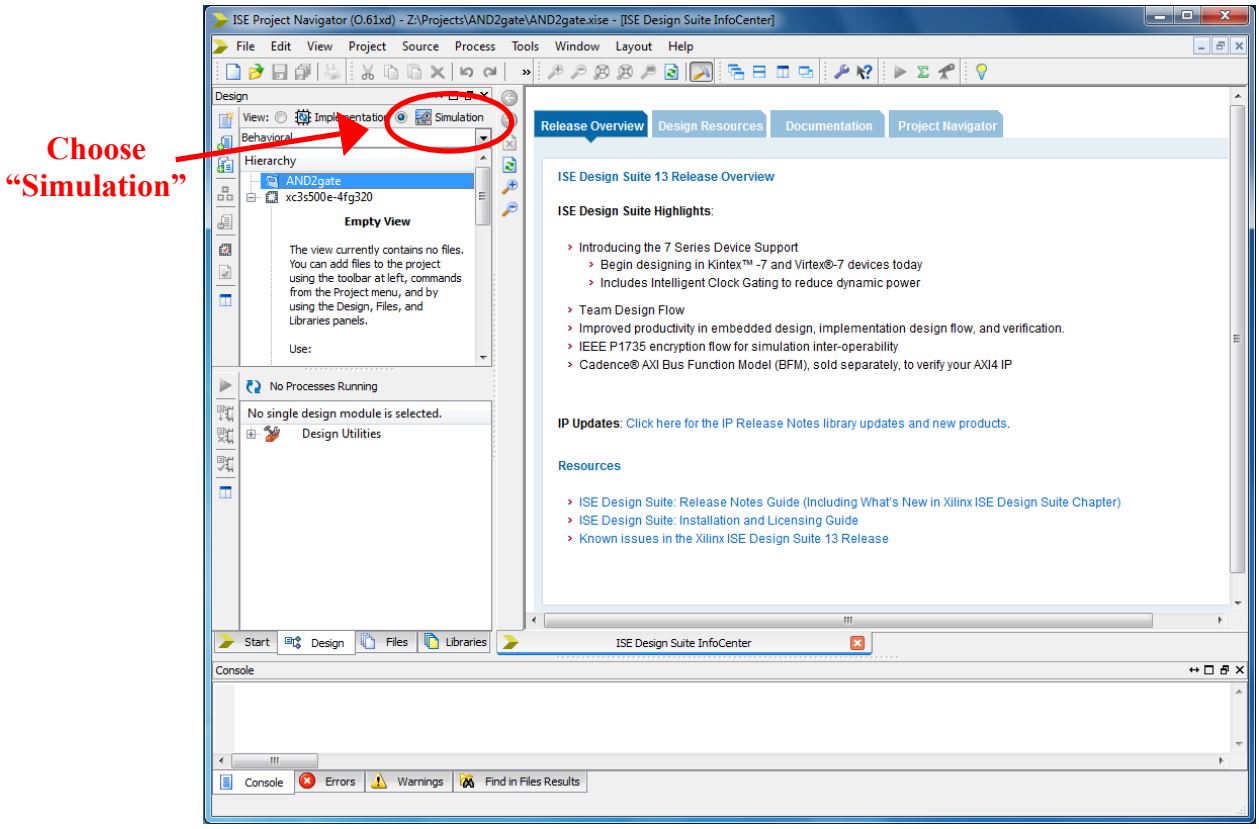
3. You will be asked to select the hardware and design flow for this project.
   - For *Family*, choose *Spartan3E*
   - For *Device*, choose *XC3S500E*
   - For *Package*, choose *FG320*
   - For *Speed,* choose *-4*
   - For *Simulator*, choose *ISim (VHDL/Verilog)*
   - Click *Next* button



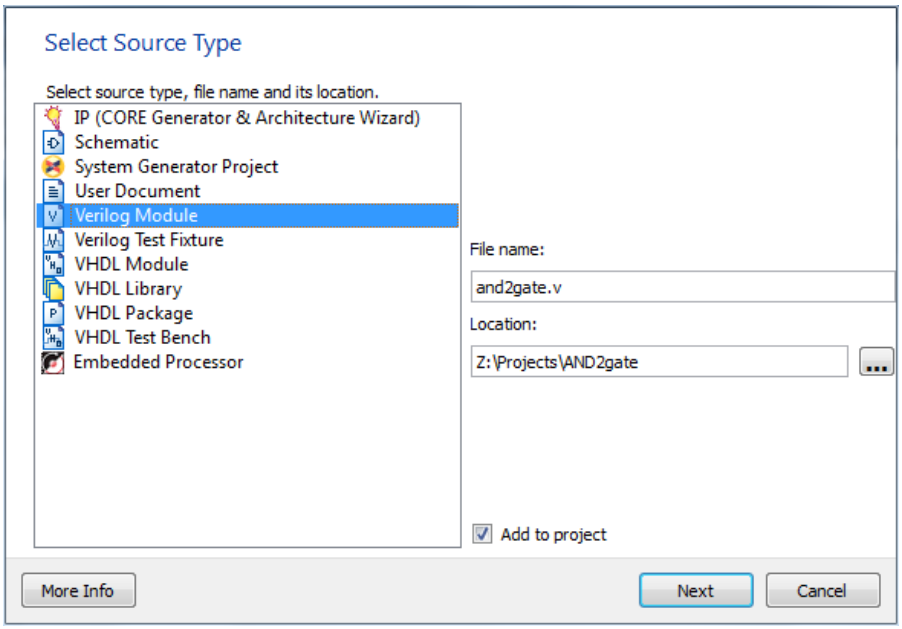
4. A project summary will appear. Click on the *Finish* button.



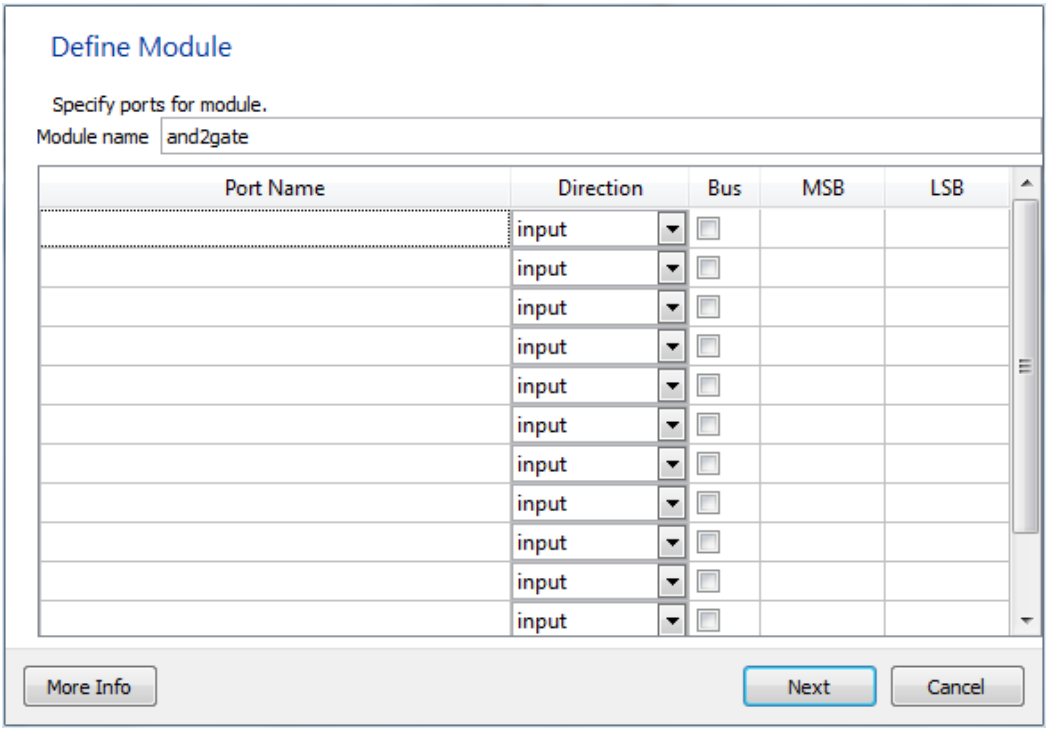
5. You now have a project by the name of "AND2gate". Next you want to specify the files in this project are for behavioral simulation.
   - Click on *Simulation*



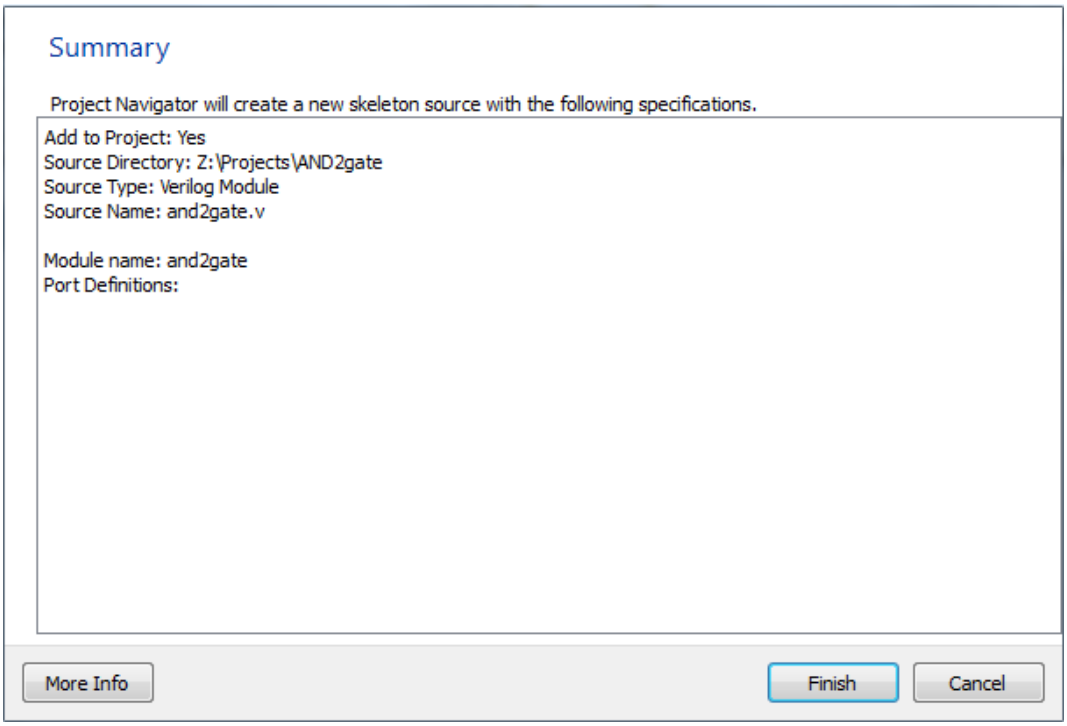
6. Now we want to add a new file to our project.
   - Click on *Project*, choose *New Source*
   - Choose *Verilog Module* as the file type
   - In the *File name:* box enter the desired file name, in this case the file is named "and2gate.v"
   - Click on the *Next* button



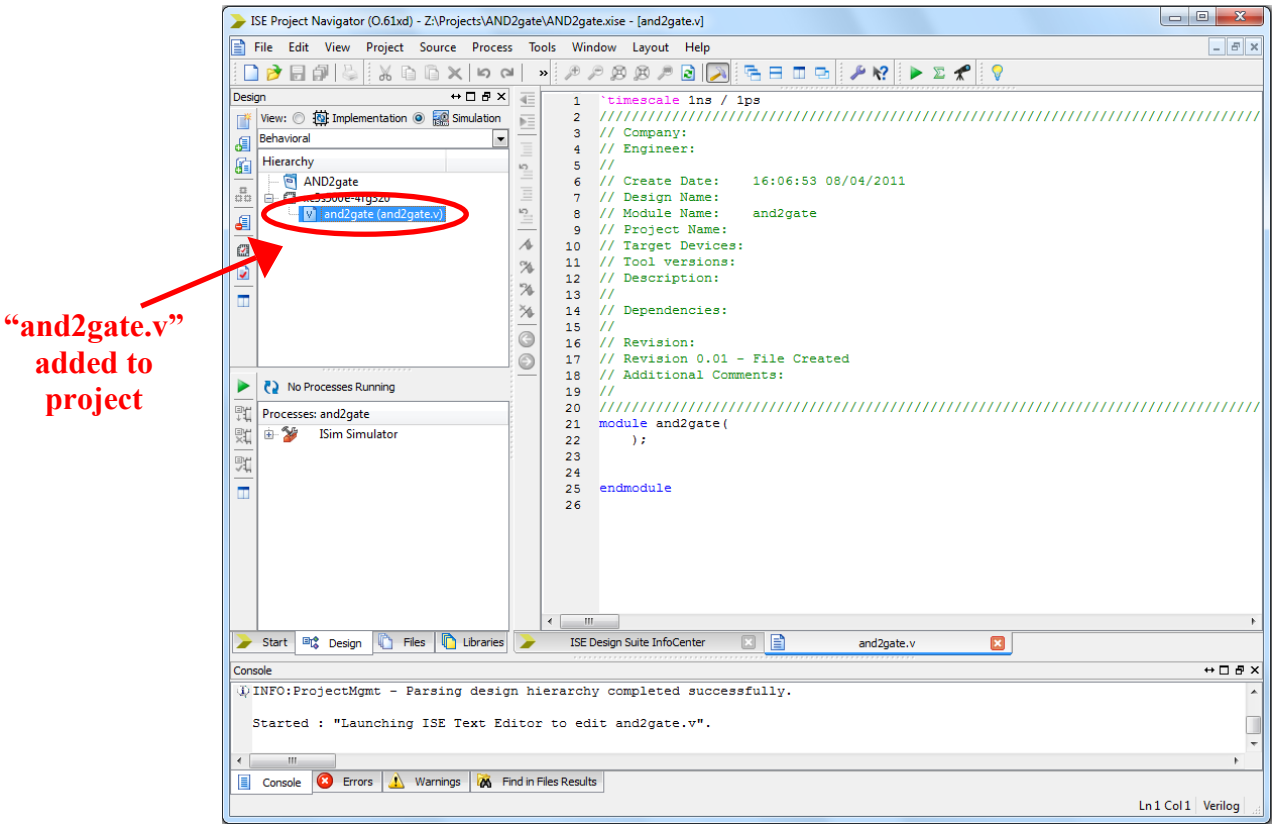
7. You will be asked for the module's port names/types. You can skip this step and click on the *Next* button.



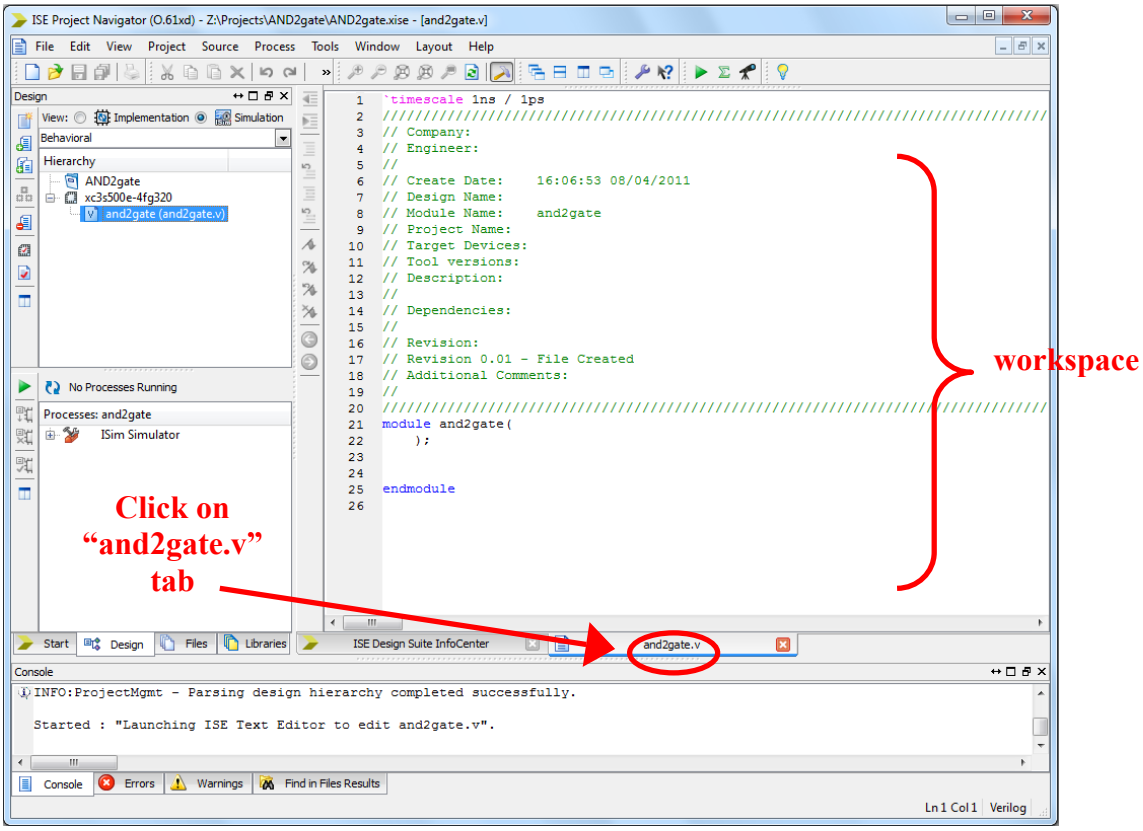
8. A project summary will appear. Click on the *Finish* button.



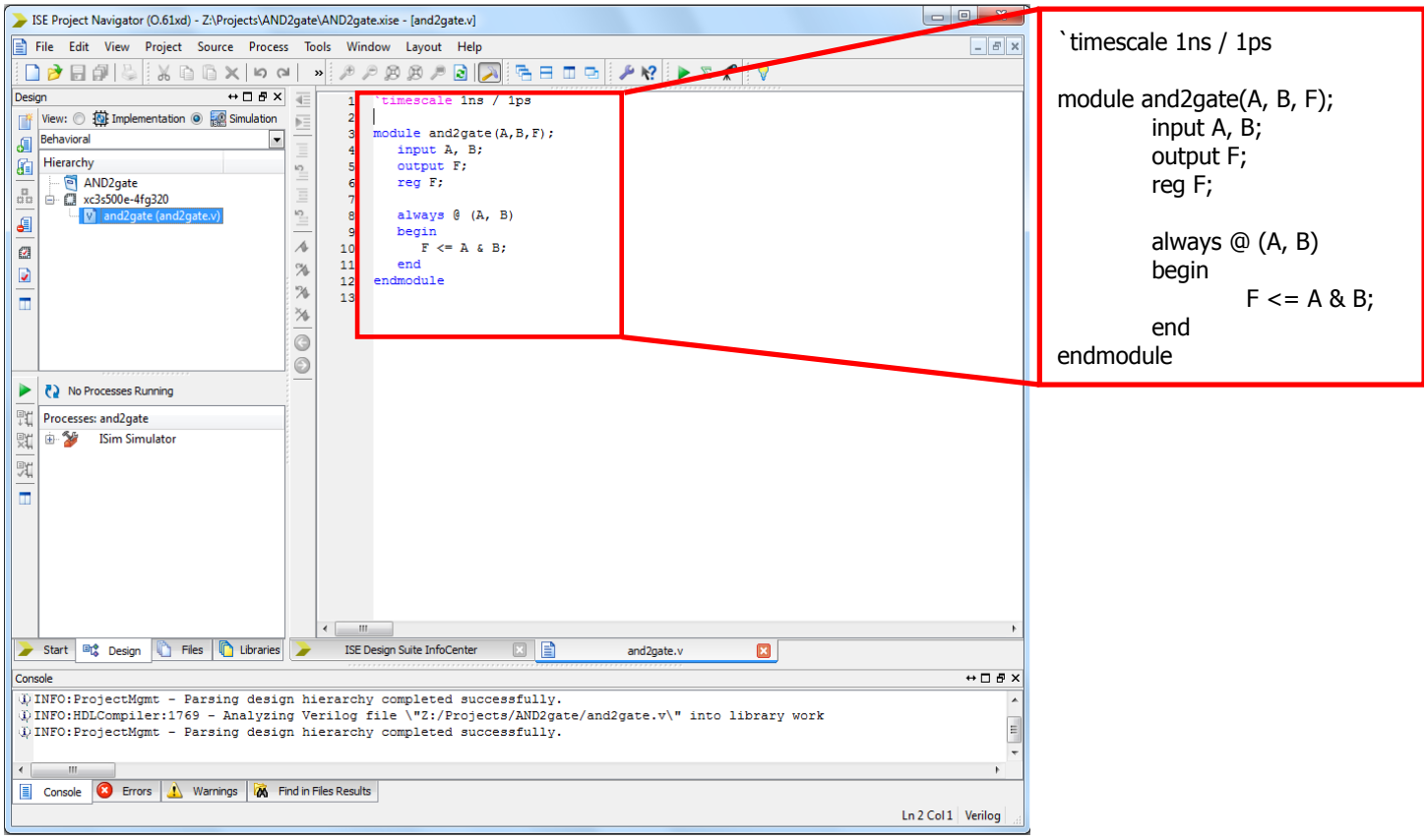
9. The “and2gate.v” file has been added to your project.



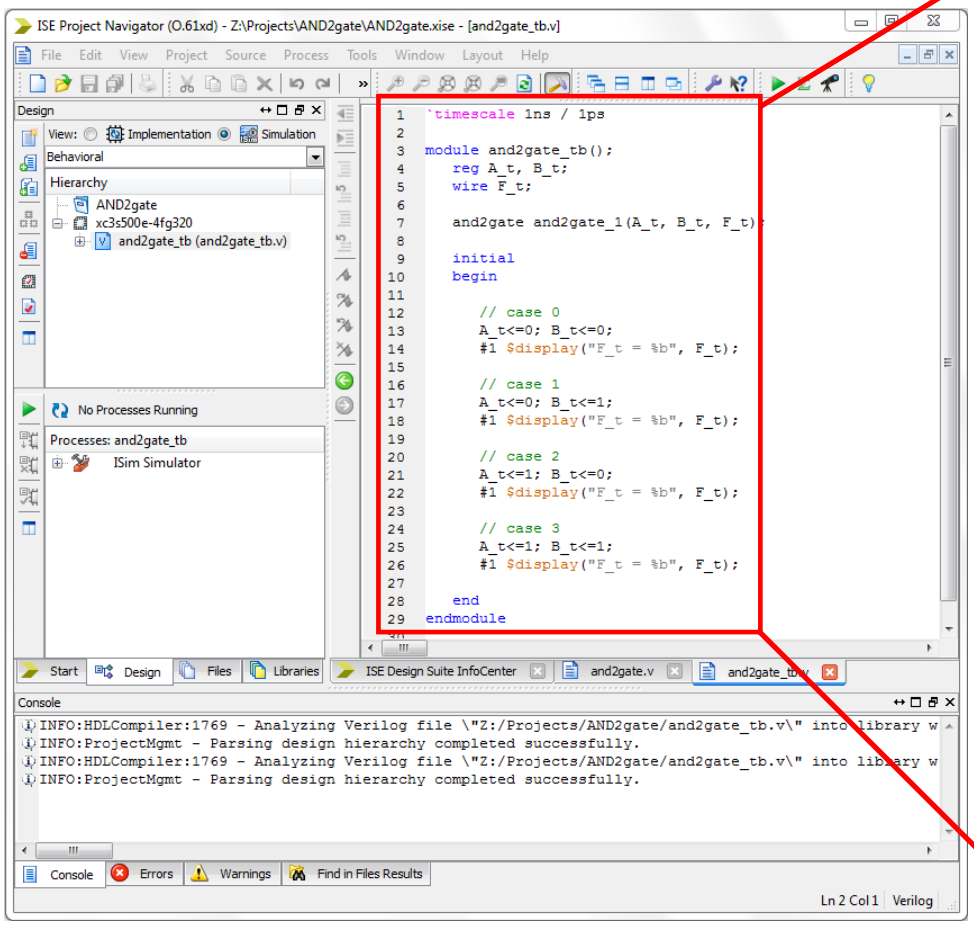
10. Click on the and2gate.v tab to show the file contents. You are now ready to specify the and2gate module’s functionality.



11. Notice that the ISE has already entered a comments sections along with a couple of lines of code for us.
    - The line "`timescale 1ns/ 1ps" is located at the top of the file. The Verilog language uses dimensionless time units, and these time units are mapped to "real" time units within the simulator. `timescale is used to map to the "real" time values using the statement `timescale <time1> / <time2>, where <time1> indicates the time units associated with the #delay values, and the <time2> indicates the minimum step time used by the simulator.
    - The and2gate module is also declared using "module and2gate();" and "endmodule", but the ports are left for us to define.
    - We finish specifying the functionality of the and2gate module as shown below.



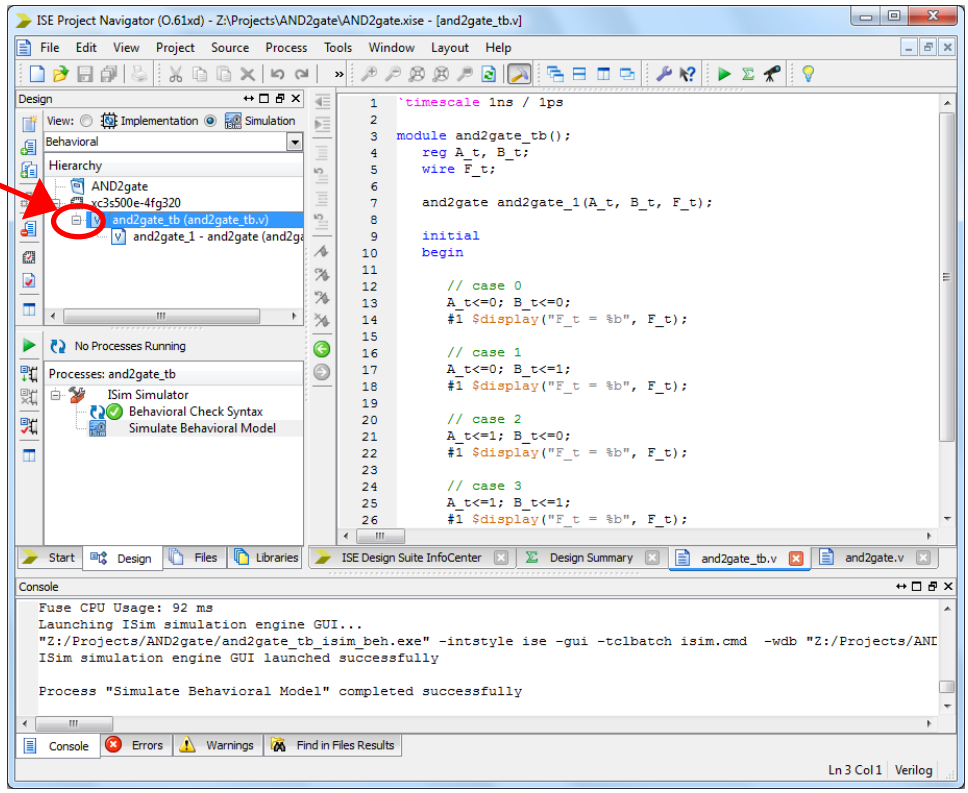
12. We also want to add a test bench and again follow Steps 8 – 11 to add “and2gate_tb.v”. Then we add the functionality of the testbench module as shown below.




```
`timescale 1ns / 1ps

module and2gate_tb();
   reg A_t, B_t;
   wire F_t;

   and2gate and2gate_1(A_t, B_t, F_t);

   initial
   begin

      // case 0
      A_t<=0; B_t<=0;
      #1 $display("F_t = %b", F_t);

      // case 1
      A_t<=0; B_t<=1;
      #1 $display("F_t = %b", F_t);

      // case 2
      A_t<=1; B_t<=0;
      #1 $display("F_t = %b", F_t);

      // case 3
      A_t<=1; B_t<=1;
      #1 $display("F_t = %b", F_t);

   end
endmodule
```

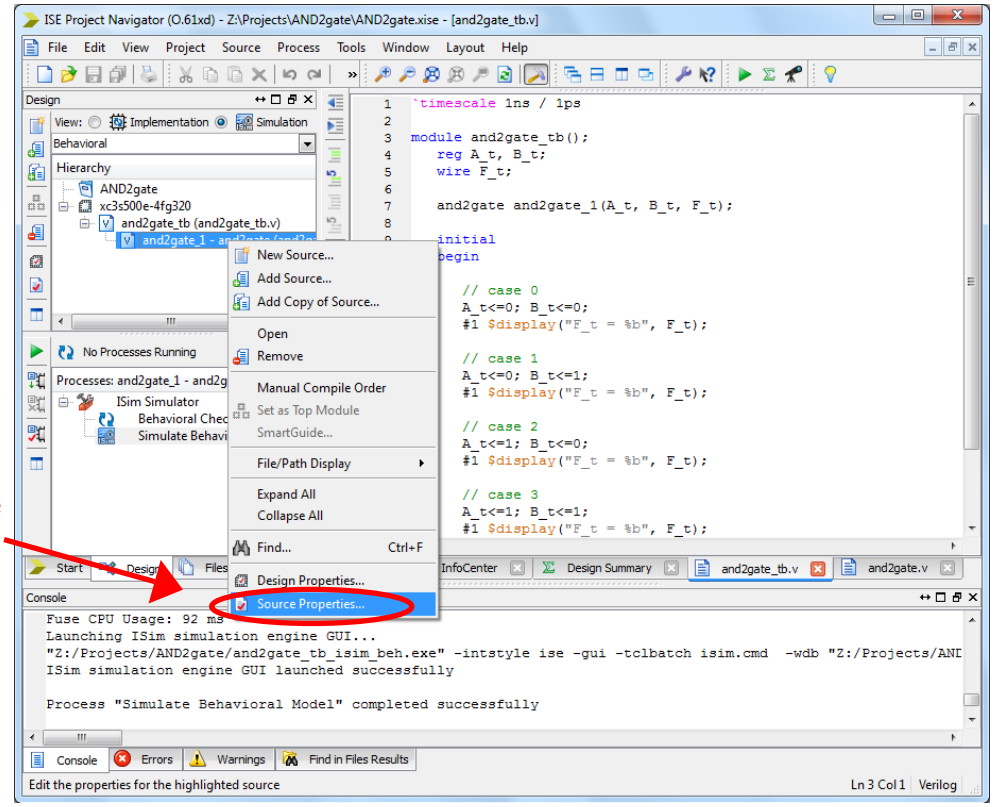
13. We now want to synthesize the AND2 gate circuit onto the Spartan3E Starter Board. While we will skip to synthesis in this tutorial, it is always a good idea to first simulate the design to ensure correctness.
    - Expand the source file listing by click on the + to left of and2gate (and2gate_tb.v) in the *Hierarchy* area.

Expand the and2gate_tb by clicking on the +

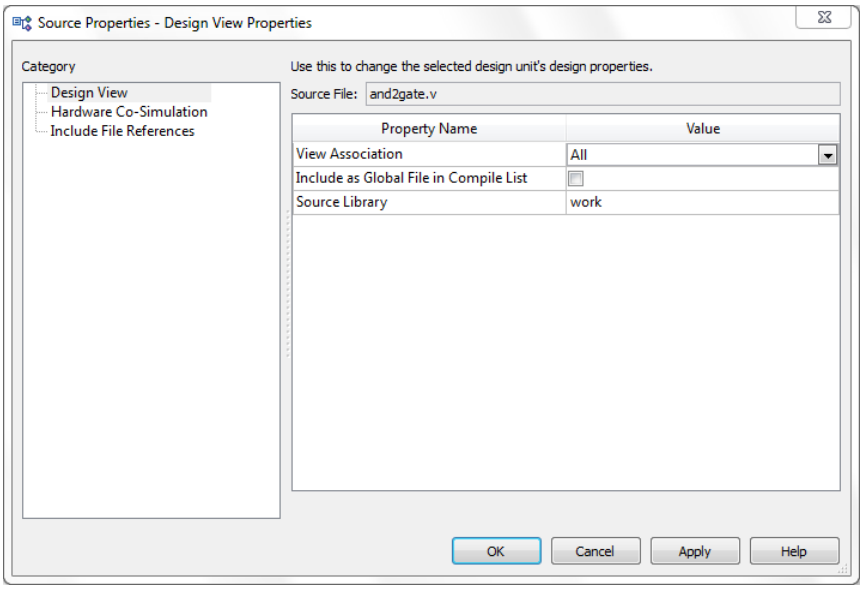


- Right click on the and2gate (and2gate.v) in the *Hierarchy* area and select *Source Properties*.

Right click on and2gate and select "Source Properties"

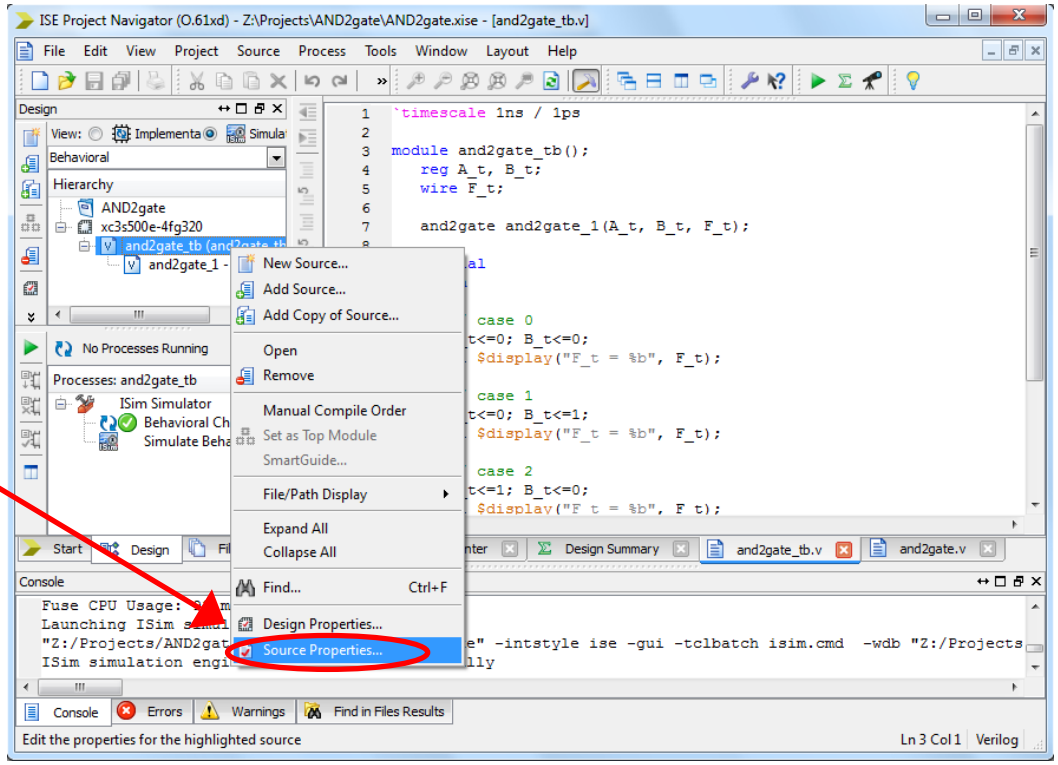


- Choose *All* as the *View Association*

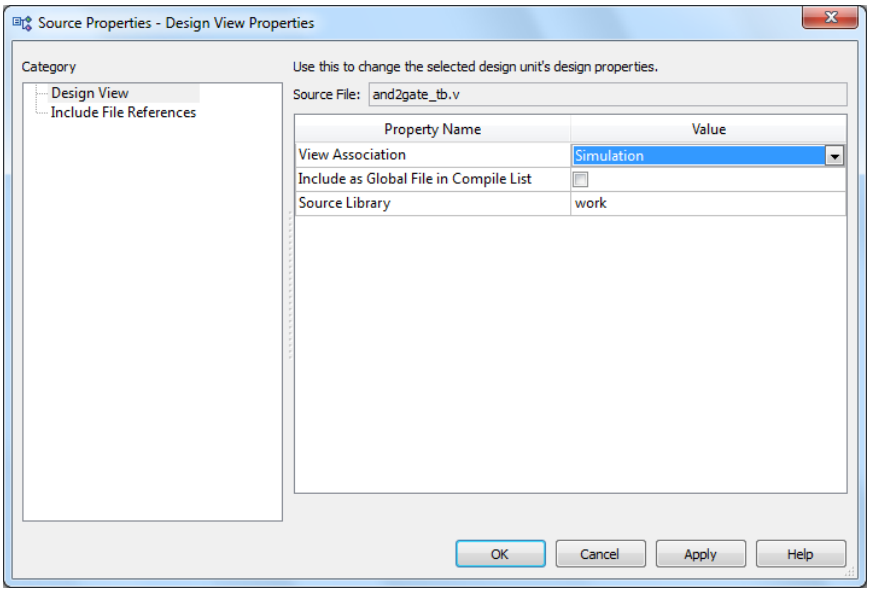


- Right click on the and2gate_tb (and2gate_tb.v) in the *Hierarchy* area and select *Source Properties*.

Right click on and2gate_tb and select "Source Properties"

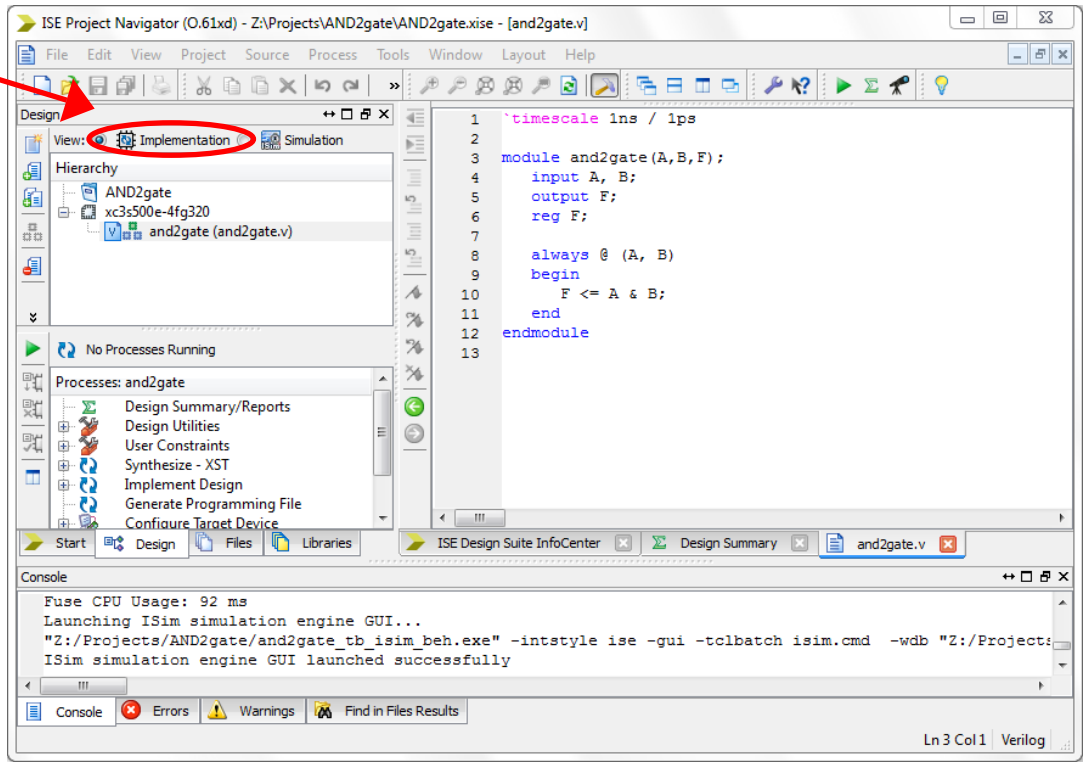


- Choose *Simulation* as the *View Association*

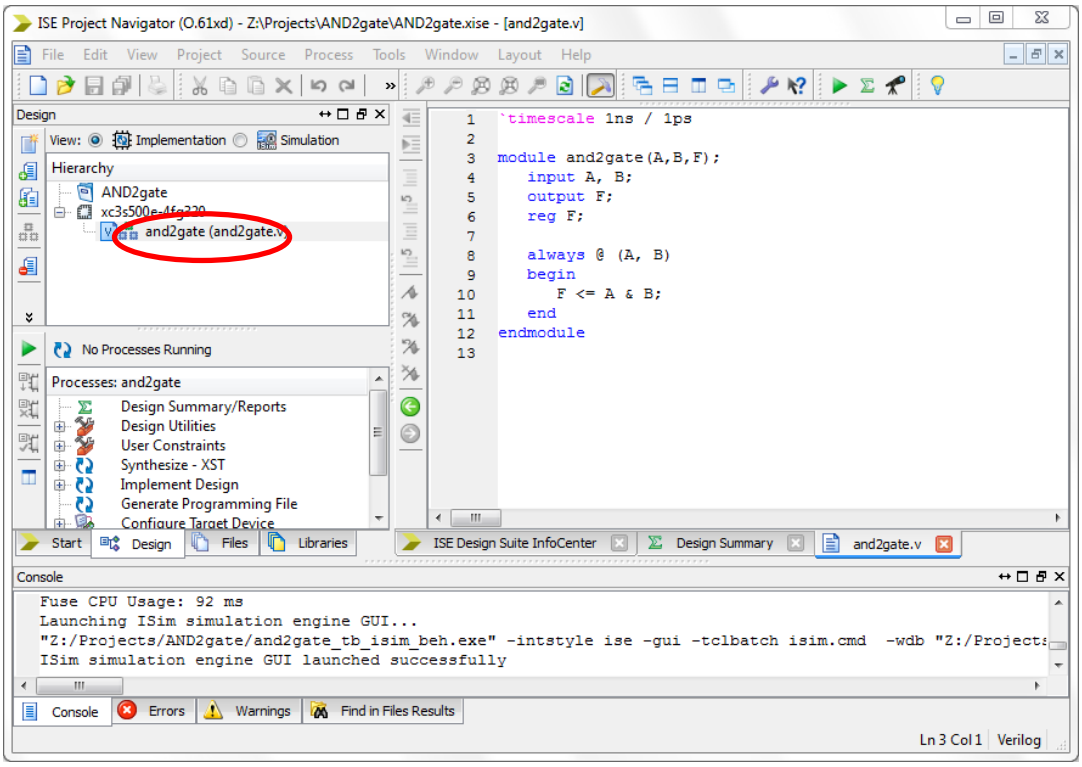


- Choose "Implementation" as the view.

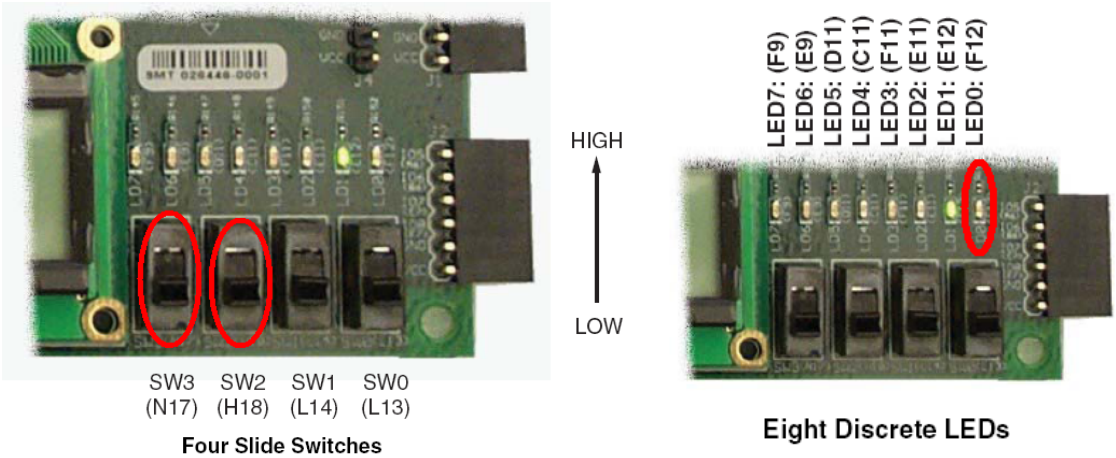
Choose "Implementation"



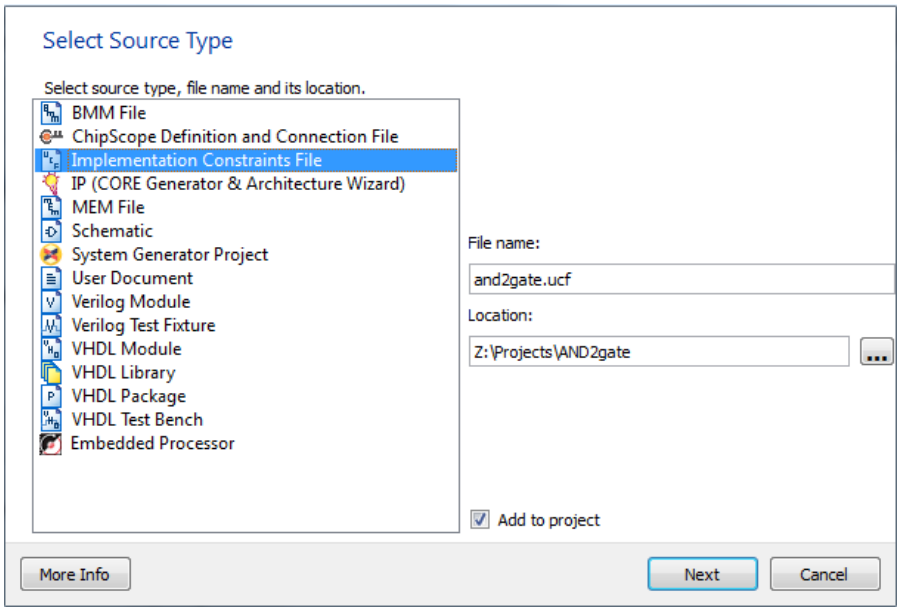
- Double-click on the and2gate (and2gate.v) in the *Hierarchy* area.



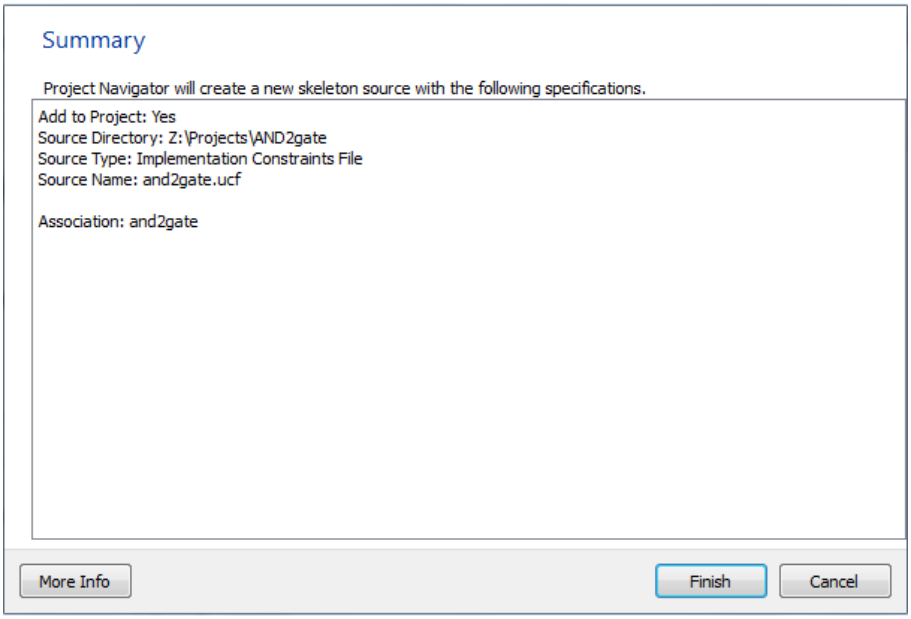
14. Before we synthesize our design, we need to map the and2gate's inputs and outputs to the pins of the FPGA that we want connected to our design using a User Constraint File (UCF). For this circuit we will use the two of the switches on the Spartan3E Starter Board as the inputs (A, B) and one of the LEDs as the output (F). The following picture show which switches and LED we will be using. Printed on the board next to each of the components is the pin number associated with that component.



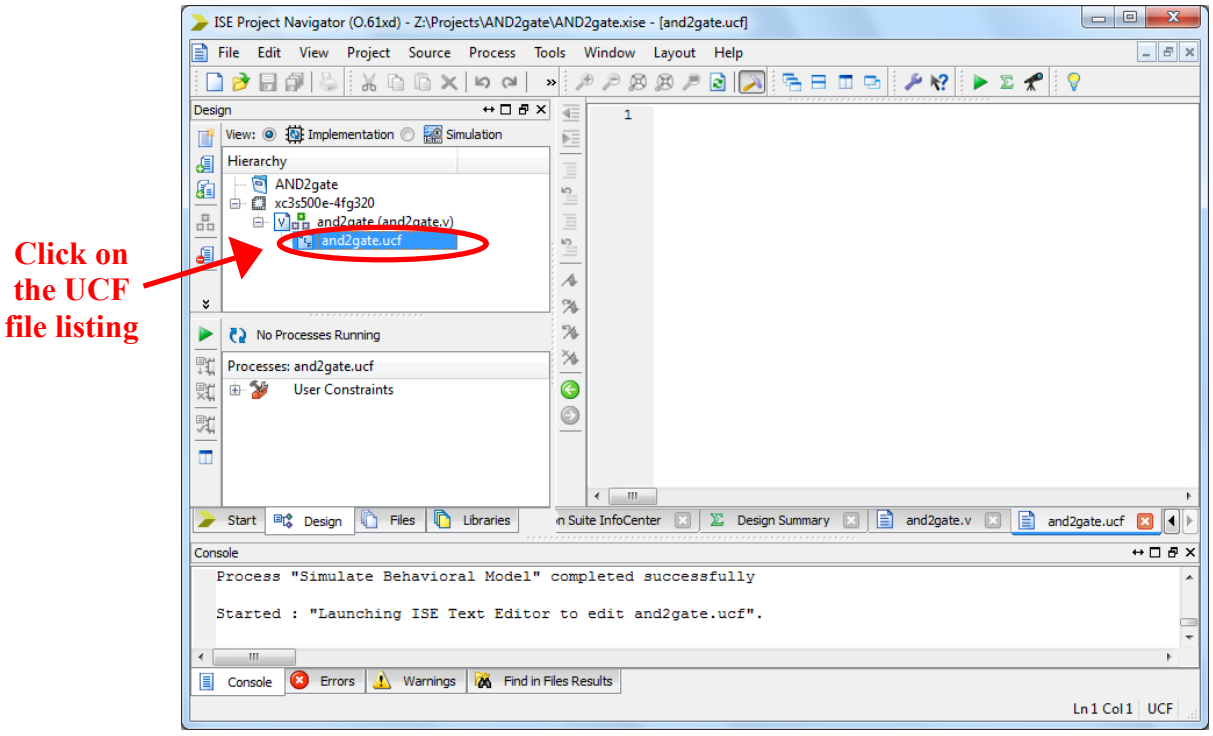
- Click on *Project*, choose *New Source*
- Choose *Implementation Constraints File* as the file type
- In the *File name:* box enter the desired file name, in this case the file is named “and2gate.ucf”
- Click on the *Next* button



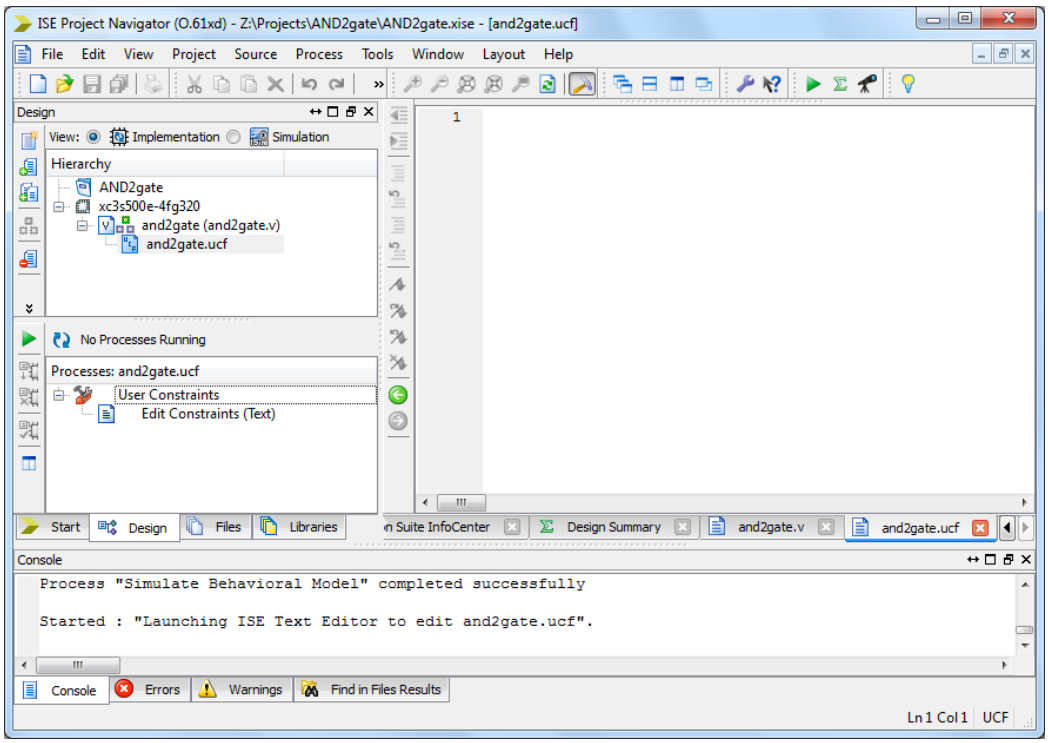
- A summary will appear. Click on the *Finish* button.



- Click on the UCF listing under and2gate (and2gate.v) in the *Hierarchy* area.

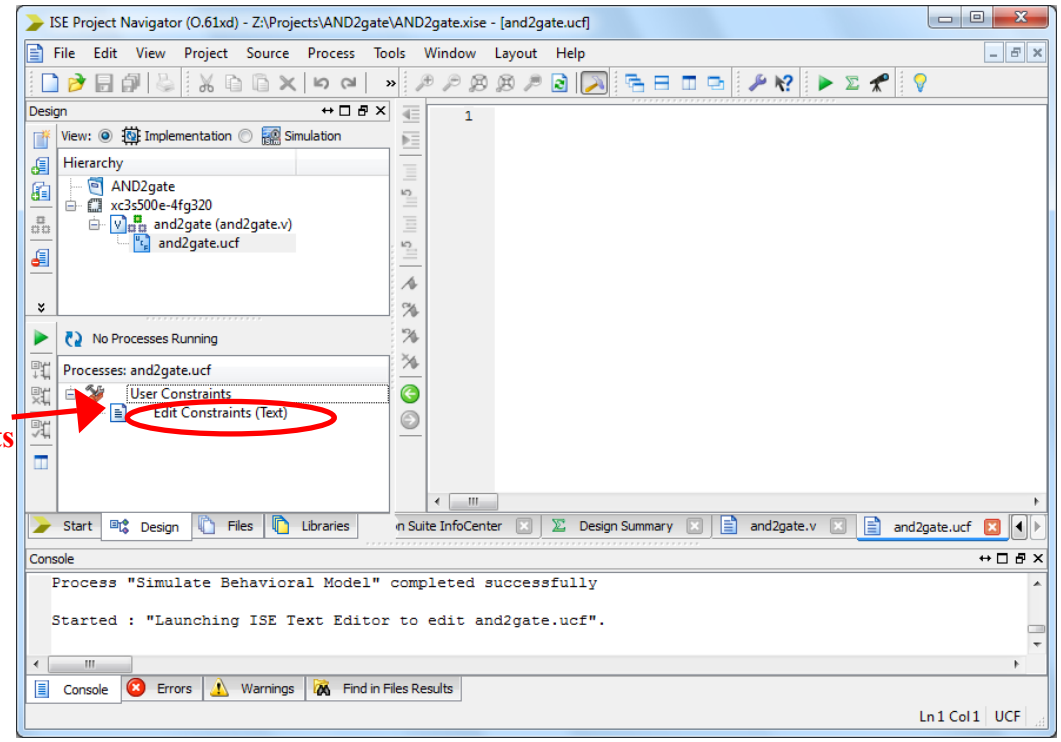


- Expand the *User Constraints* option by clicking on the + symbol located to the left.

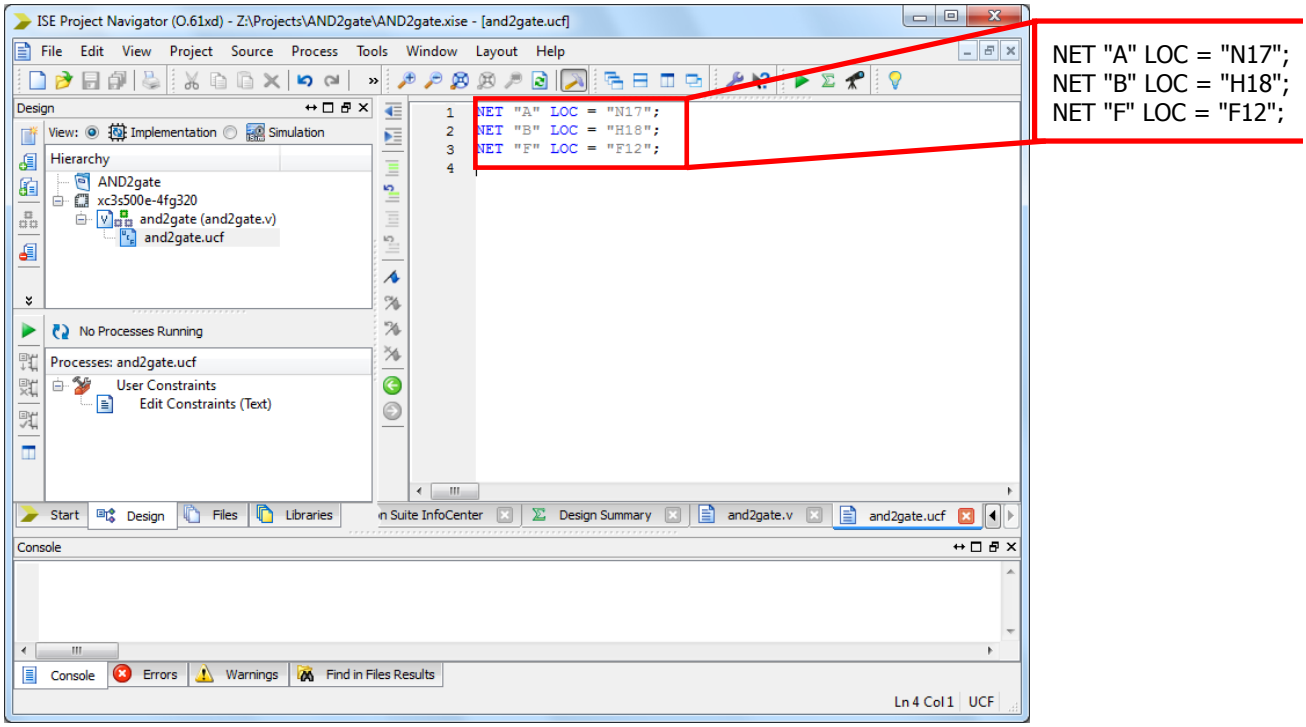


- Double-click on *Edit Constraints.*

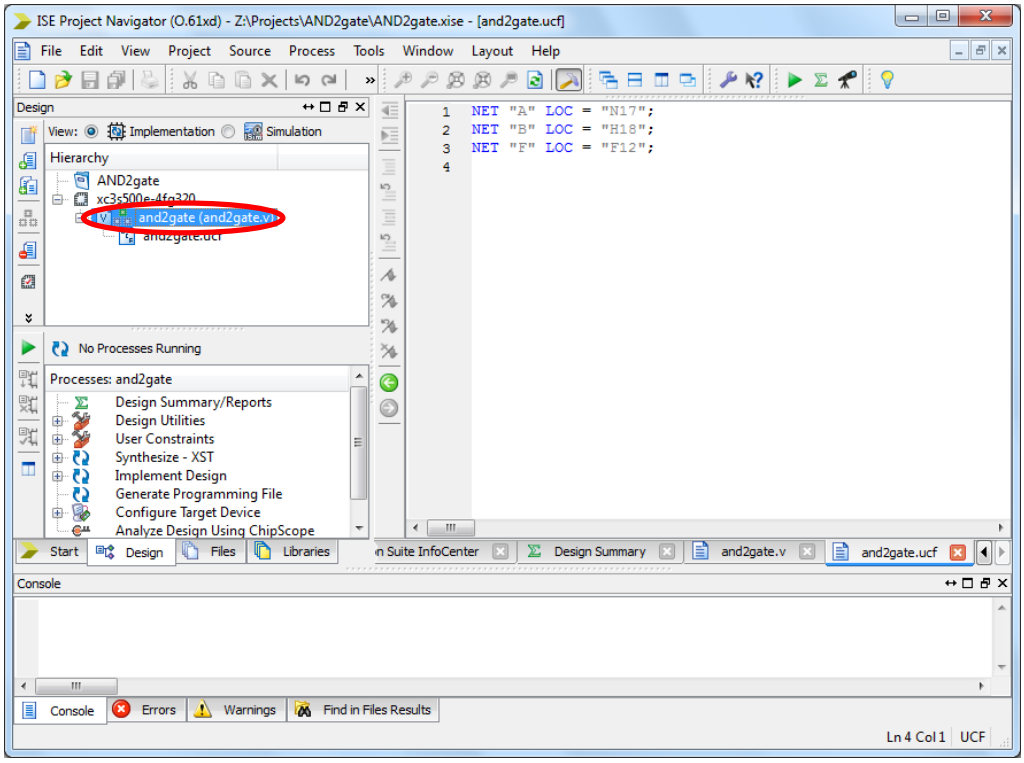
Double-click on "Edit Constraints (Text)"



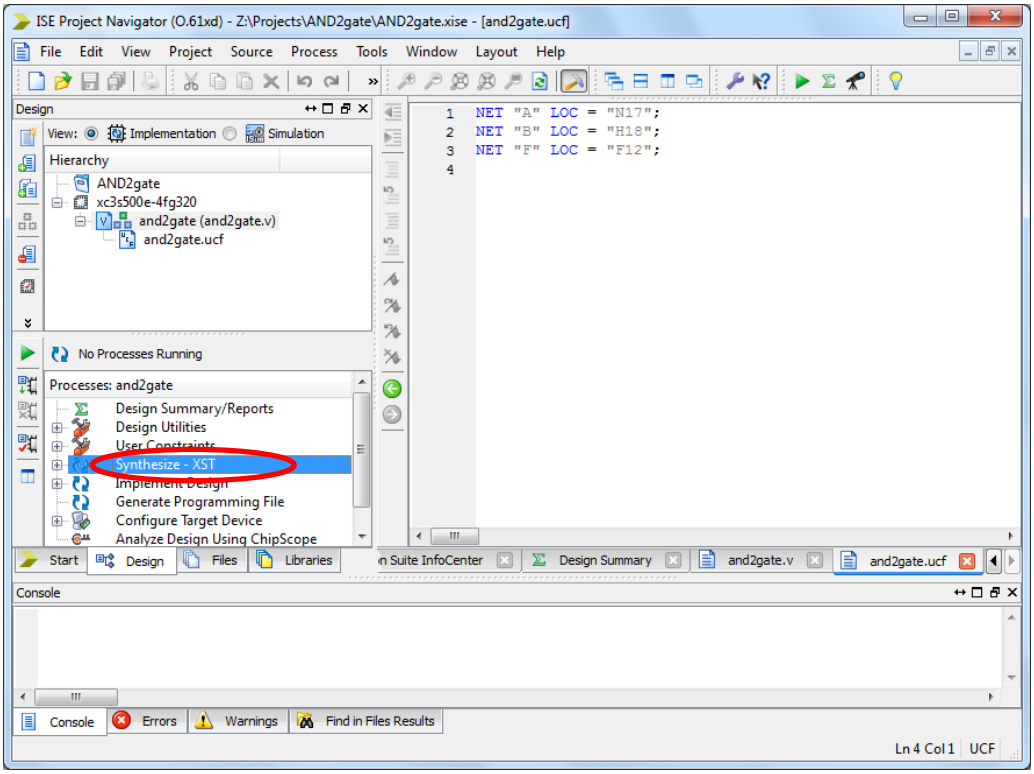
- We specify the connection between our and2gate design and the FPGA's pins as shown below. Be sure to save the UCF file before proceeding.



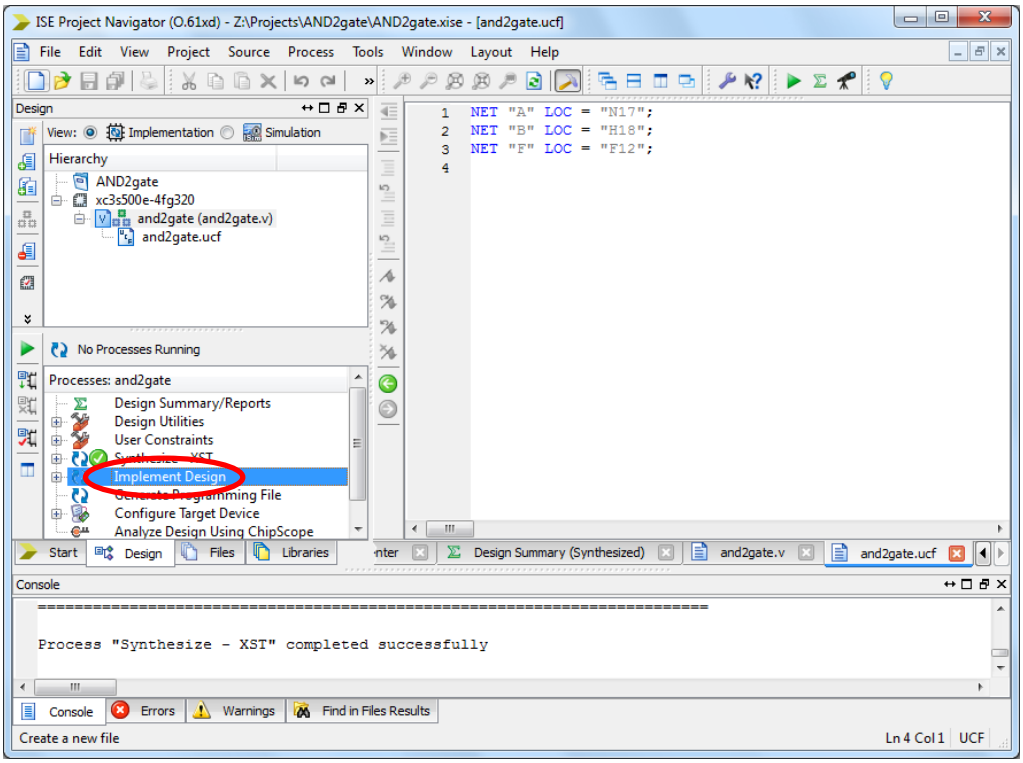
15. We can now synthesize our design.
    - Click on the and2gate (and2gate.v) in the *Hierarchy* area.



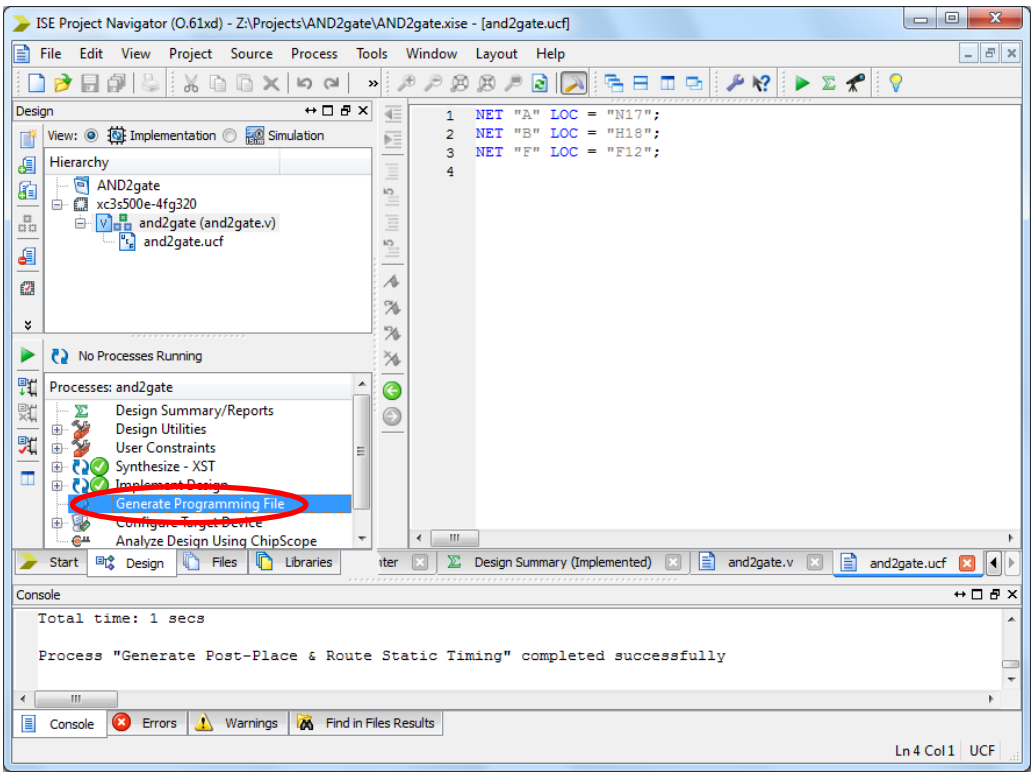
    - Double-click on *Synthesize – XST*. This step will synthesize your design to the basic logic structures of the FPGA (LUTs). When completed, you should hopefully see a message *Process "Synthesize" completed successfully*. If not, please go back and make sure you followed the previous steps correctly.



- Double-click on *Implement Design.* This step will create a final implementation for your design. When completed, you should see a message *Process "Generate Post-Place & Route Static Timing" completed successfully*. If not, please go back and make sure you followed the previous steps correctly.

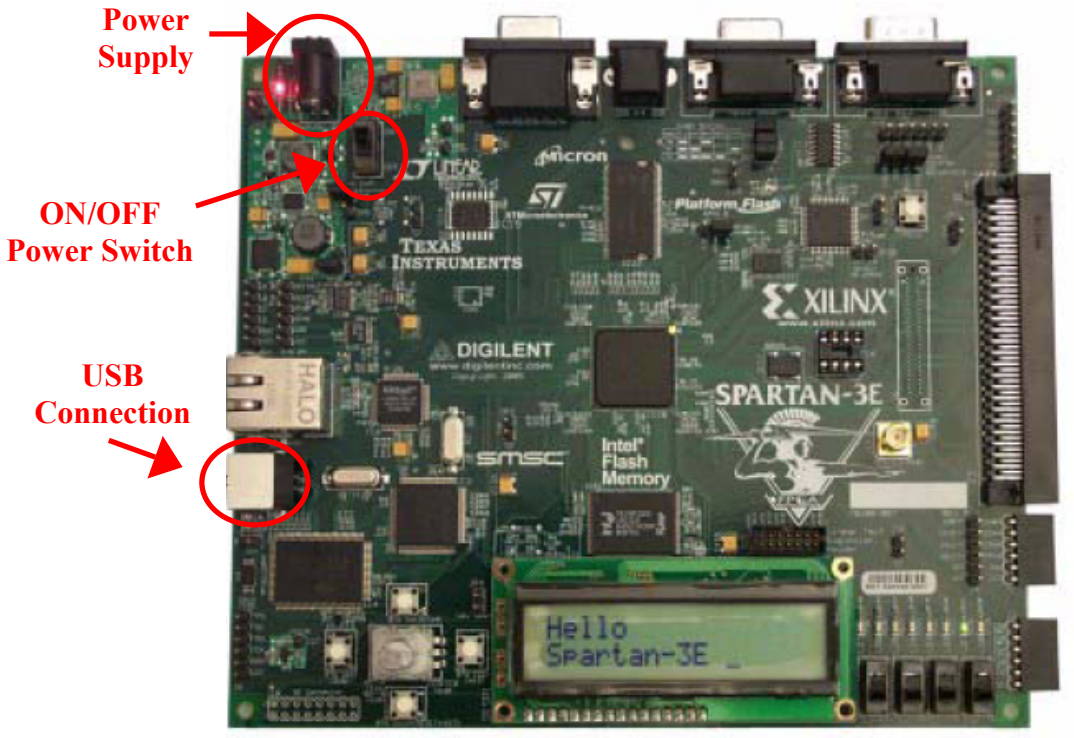


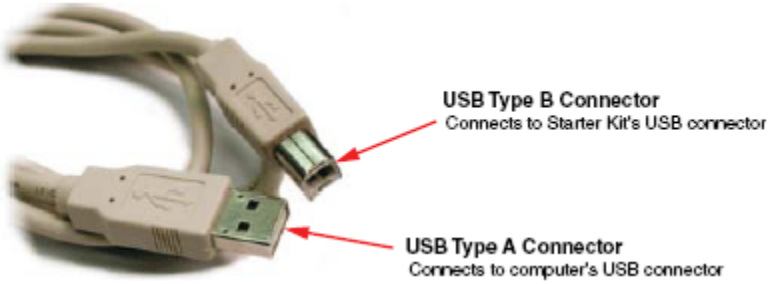
- Double-click on *Generate Programming File.* This step will create the bitstream needed to program the FPGA on the Spartan3E Starter Board. When completed, you should hopefully see a message *"Generate Programming File" completed successfully*. If not, please go back and make sure you followed the previous steps correctly.

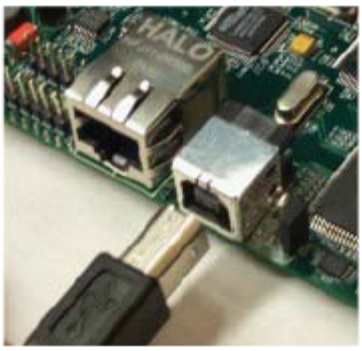


16. We are now ready to program the Spartan3E Starter Board. You should be as **careful as possible** when using these FPGA development boards. If you need to unplug the power, please **wait 30 seconds** before plugging the power supply in again. This will help to ensure a long life for the board.

- Plug the power supply into an appropriate wall socket.
- Plug the power supply into the Spartan3 starter board.
- Connect the Spartan3 Starter Board to the USB port of your computer using the supplied programming cable.
- Connect the other end of the supplied cable to Spartan3 Starter Board at the location shown below.
- Make sure your board is "ON", a light should illuminate by the power supply if the board is on.

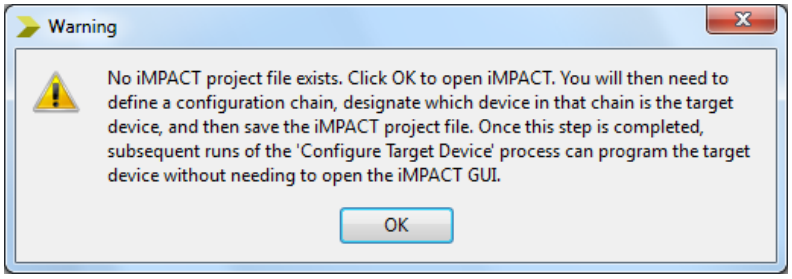






17. We will run the Xilinx iMPACT tool utilized to program the FPGA.

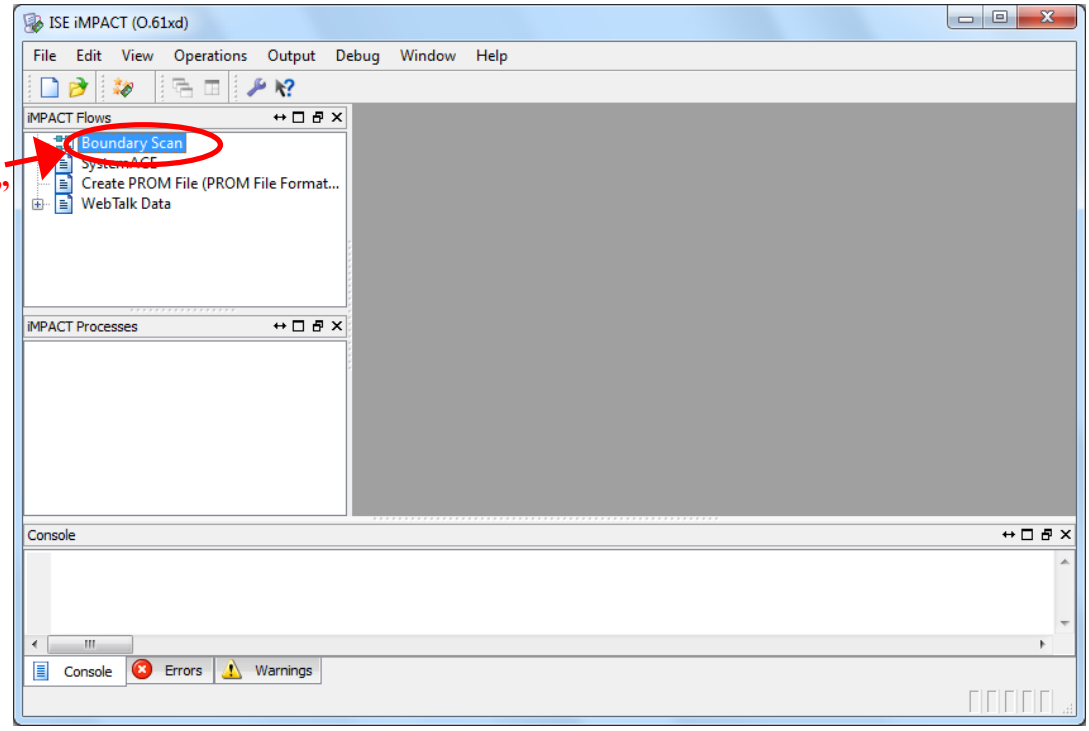
- Double-click on *Configure Target Device*. This will launch the iMPACT tool in a separate window.
- You may be presented with a Warning window. Click OK.



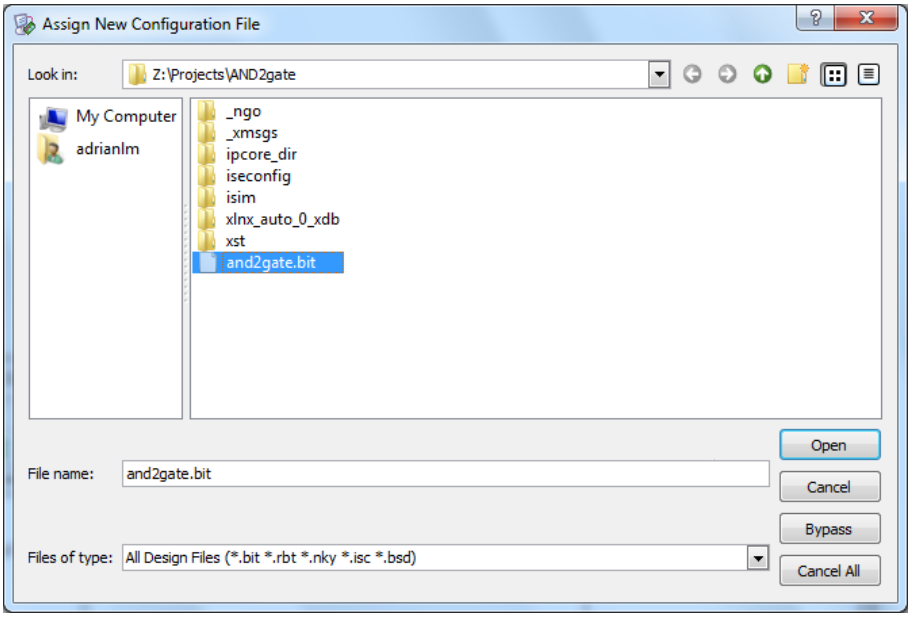
- ***Note: It may take a few minutes for the program to launch the first time. Please be patient.***

- Double-click on *Boundary-Scan* in the *iMPACT Flows* area.

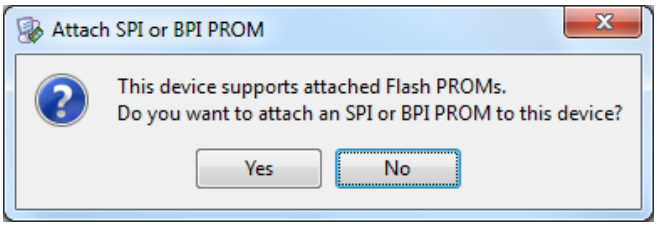
Double-click on "Boundary Scan"



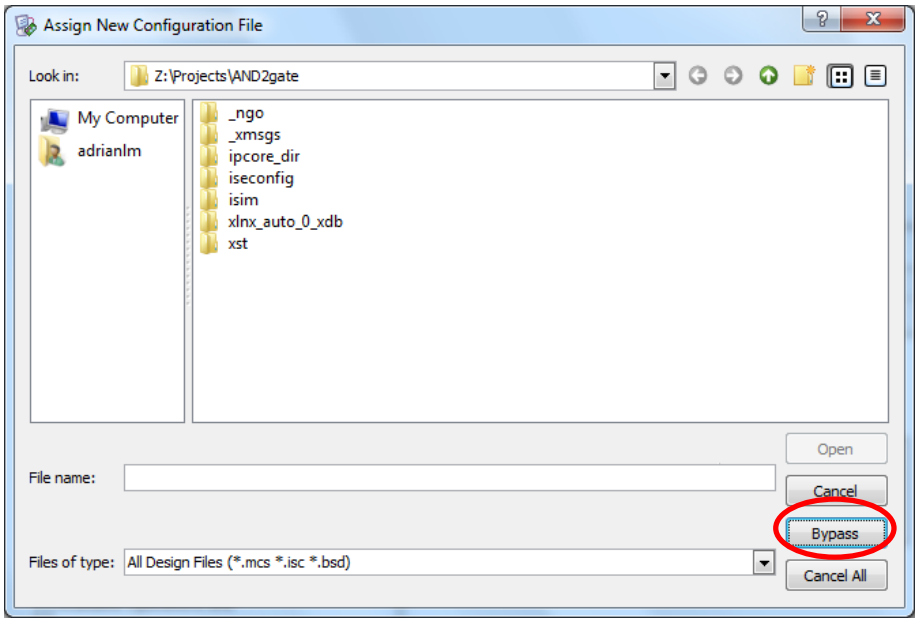
- Select *Initialize Chain* from the *File* menu. Click *Yes* if prompted with an *Auto Assign Configuration Files Query Dialog* window
- You will be prompted to Assign a New Configuration File. This file is for the Spartan3E FPGA we are configuring. Choose the and2gate.bit file in the Z:\Projects\AND2gate directory. Click *Open*.



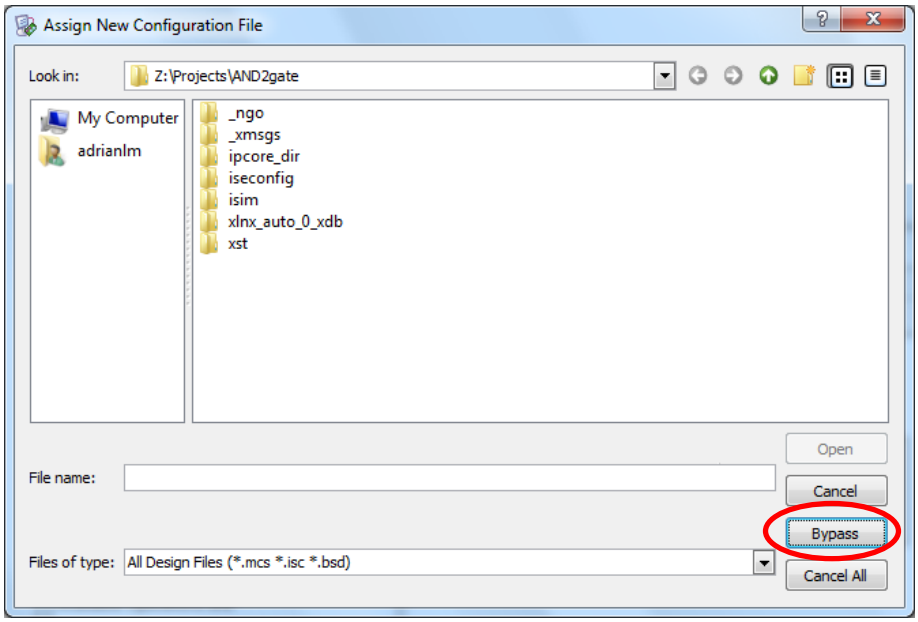
- You will be asked to attach an SPI or BPI PROM. Click *No*.



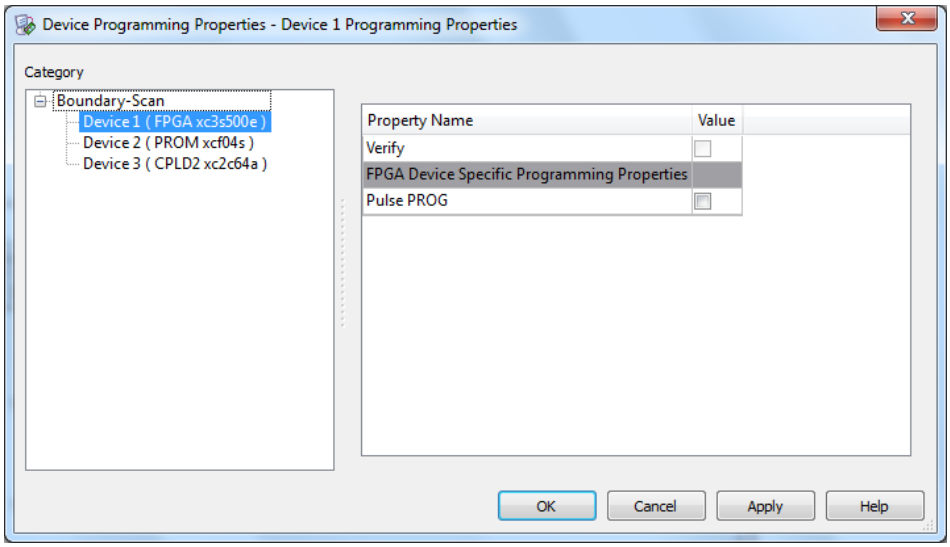
- You will again be prompted to Assign a New Configuration File. This file is for the FLASH memory that can be used to store the FPGA configuration. We will not be using the FLASH at this time. Click *Bypass*.



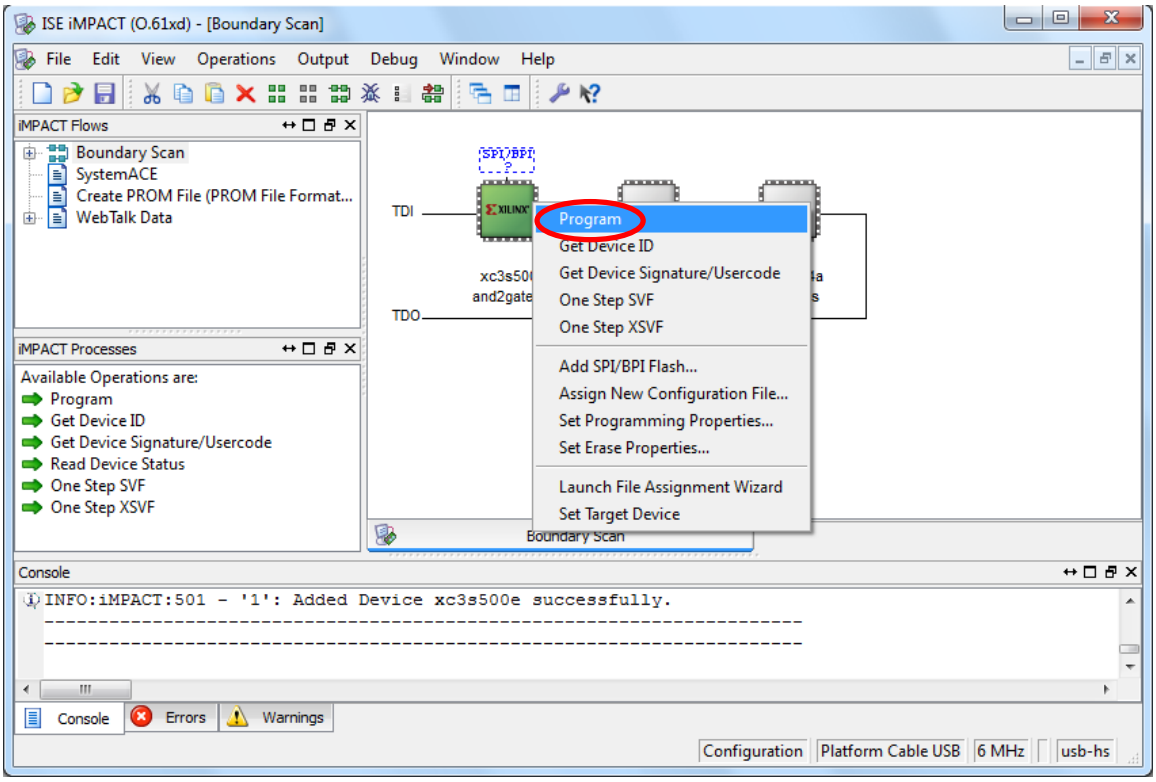
- You will again be prompted to Assign a New Configuration File. This file is for the CPLD. We will not be using the CPLD at this time. Click *Bypass*.



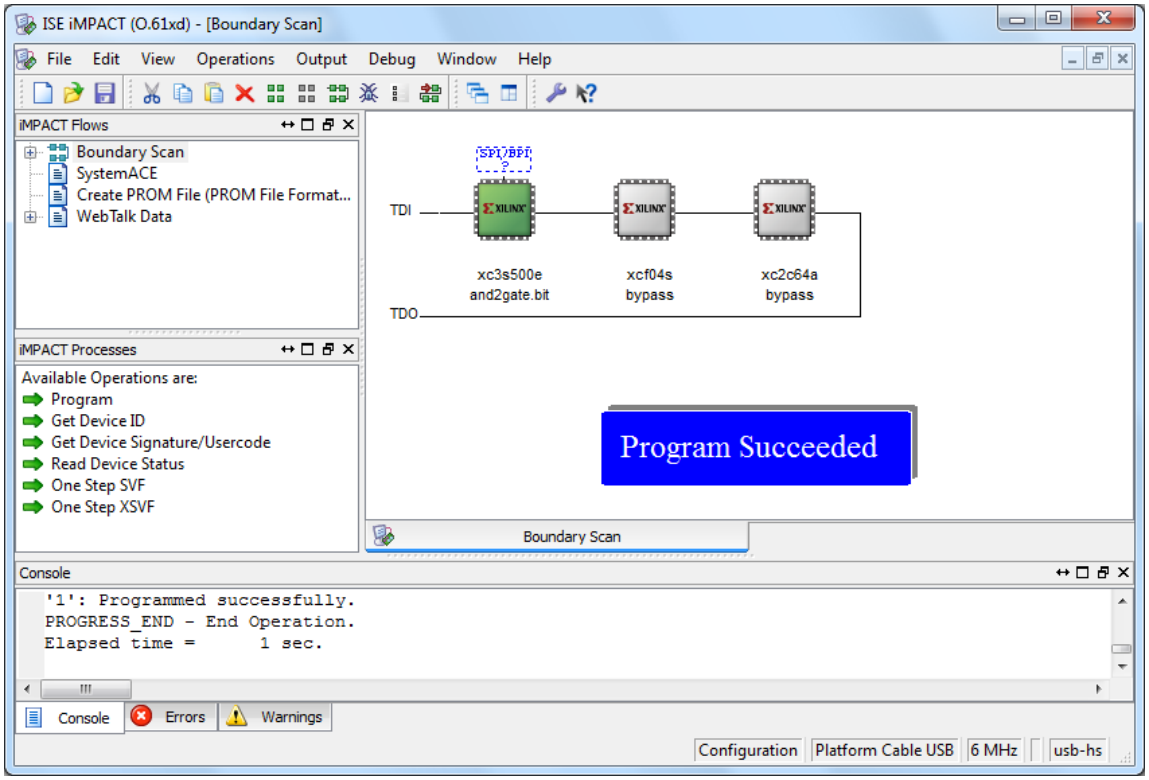
- A Device Programming Properties Dialog will open. Click *OK*.



- Right click on the device labeled xc3s500e and choose *Program...*



- Upon programming the FPGA you should hopefully see the *Program Successful* message as shown below.



- If instead you receive a *Program Failed* message, try the following steps to correct the problem.
  - Remove power from the Spartan3 Starter Board by carefully unplug the power supply from the board.
  - ***Wait 30 seconds*** and apply power by carefully plugging the power supply into the board.
  - Program the device again.

18. Congratulations. You have successfully synthesized and implemented your 2-input AND gate onto the Spartan3E Starter Board. You can test your AND gate by changing the input switches. When both switches are enabled (in the up position) the LED should illuminate. For all other switch configurations, the LED will be off.