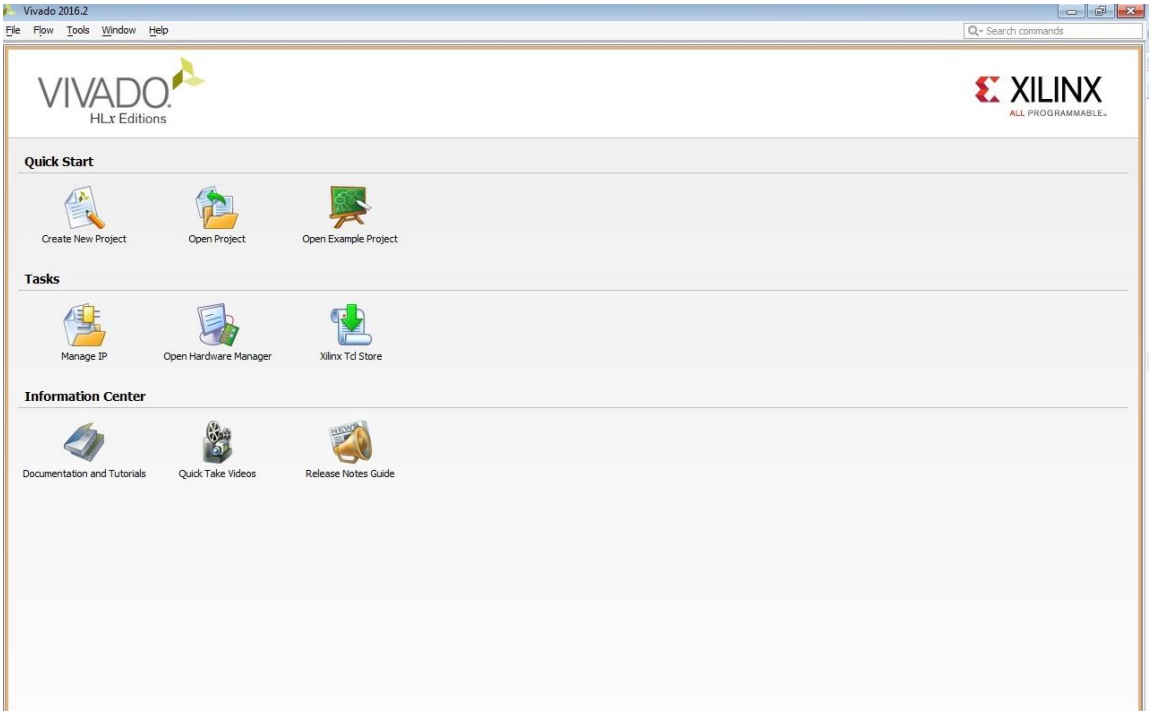
EECS 140/141 Lab 1

This tutorial shows how to create a two input AND gate, that can be implemented on the Basys3 board.

Start Vivado Design Suite:

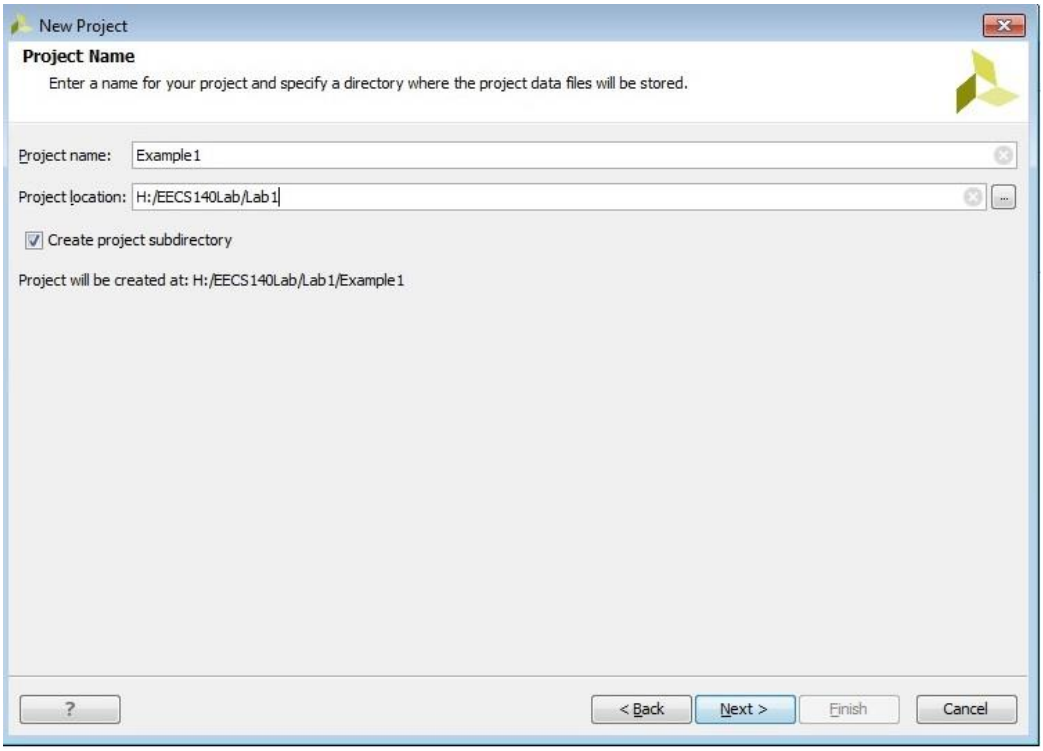


Select Create New Project.

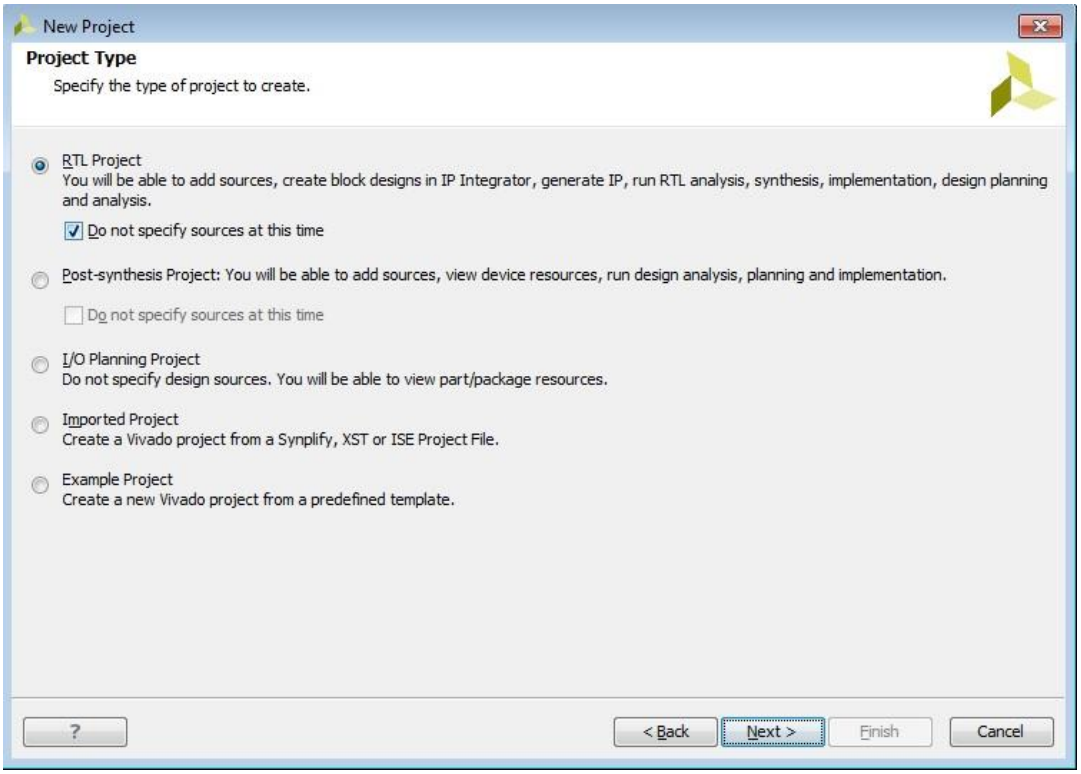
Click Next and then enter a Project name and location for your project:

Note: Make sure you create the project in H:/ Drive. Prior to that, create a folder

“EECS140_Lab” in your H:/ drive for convenience.

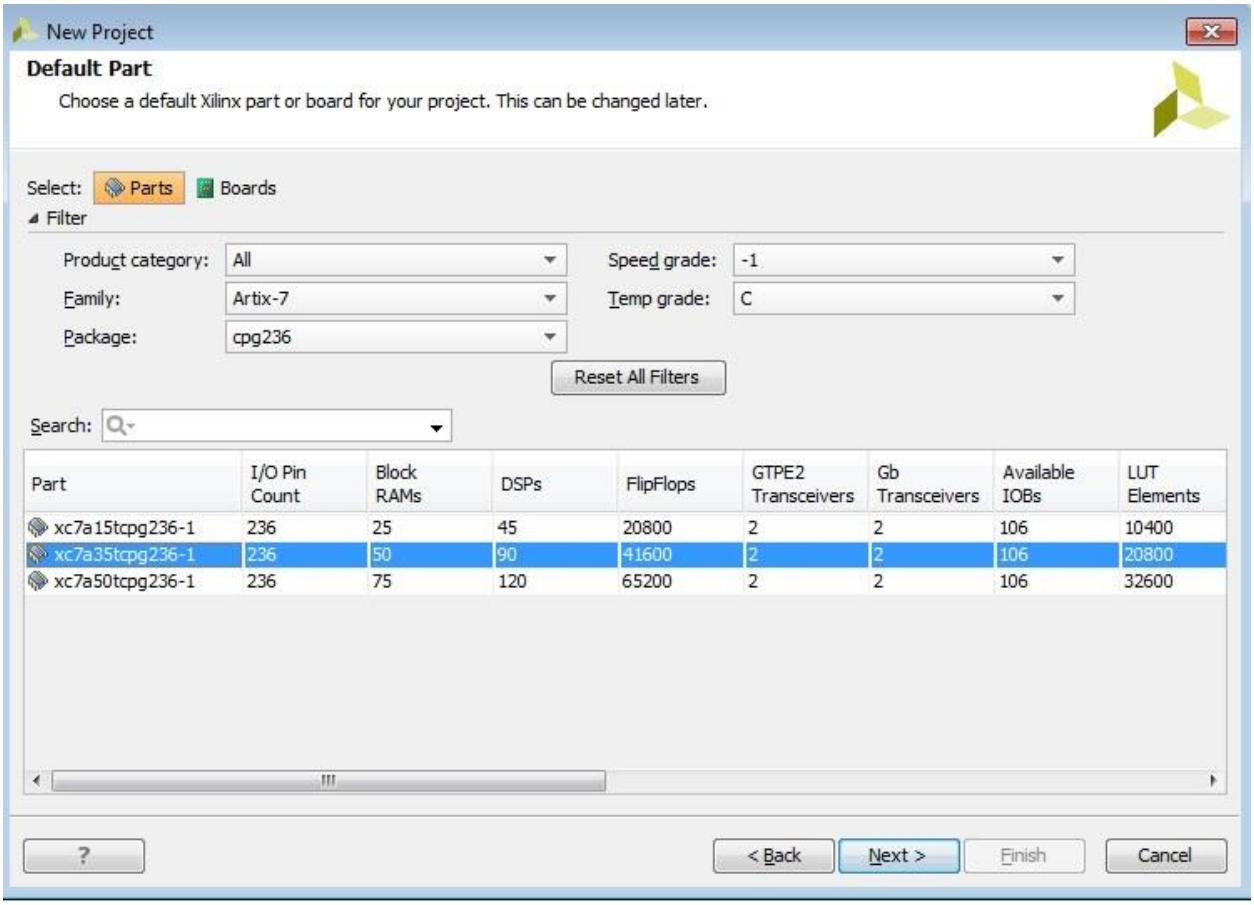


Click Next and select the RTL project type:

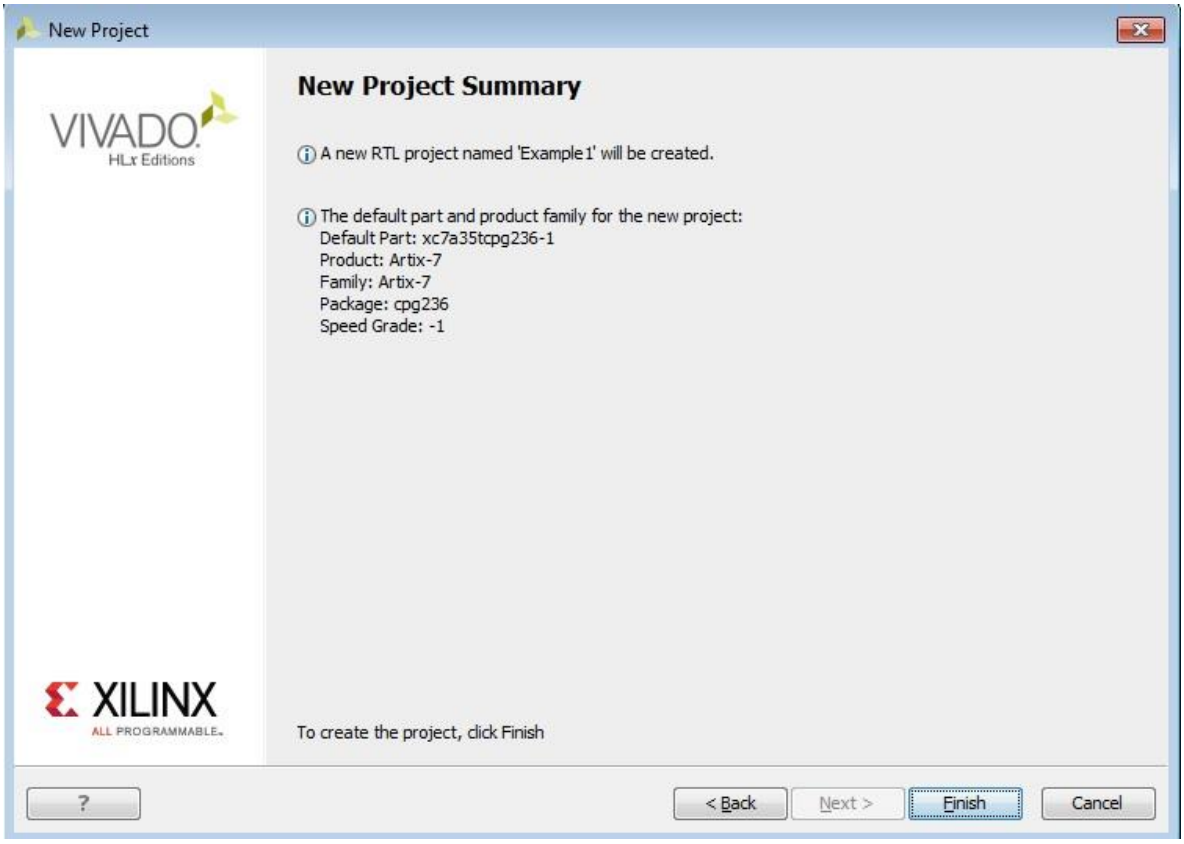


Check the “Do not specify sources at this time” box and click Next:

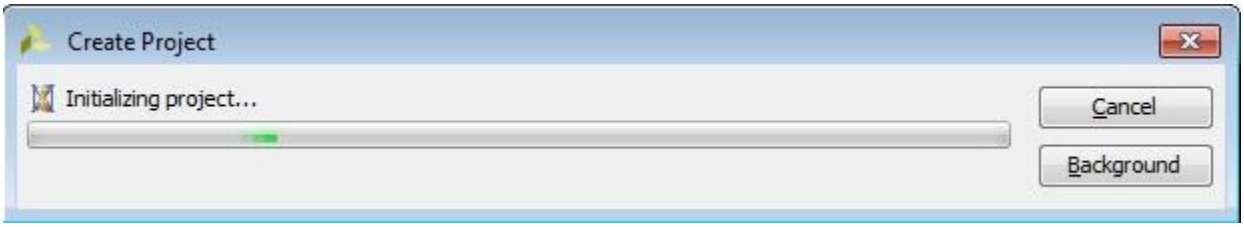
Select the correct Xilinx FPGA that is on the Basys3 board (XC7A35T-1CPG236C) as shown below

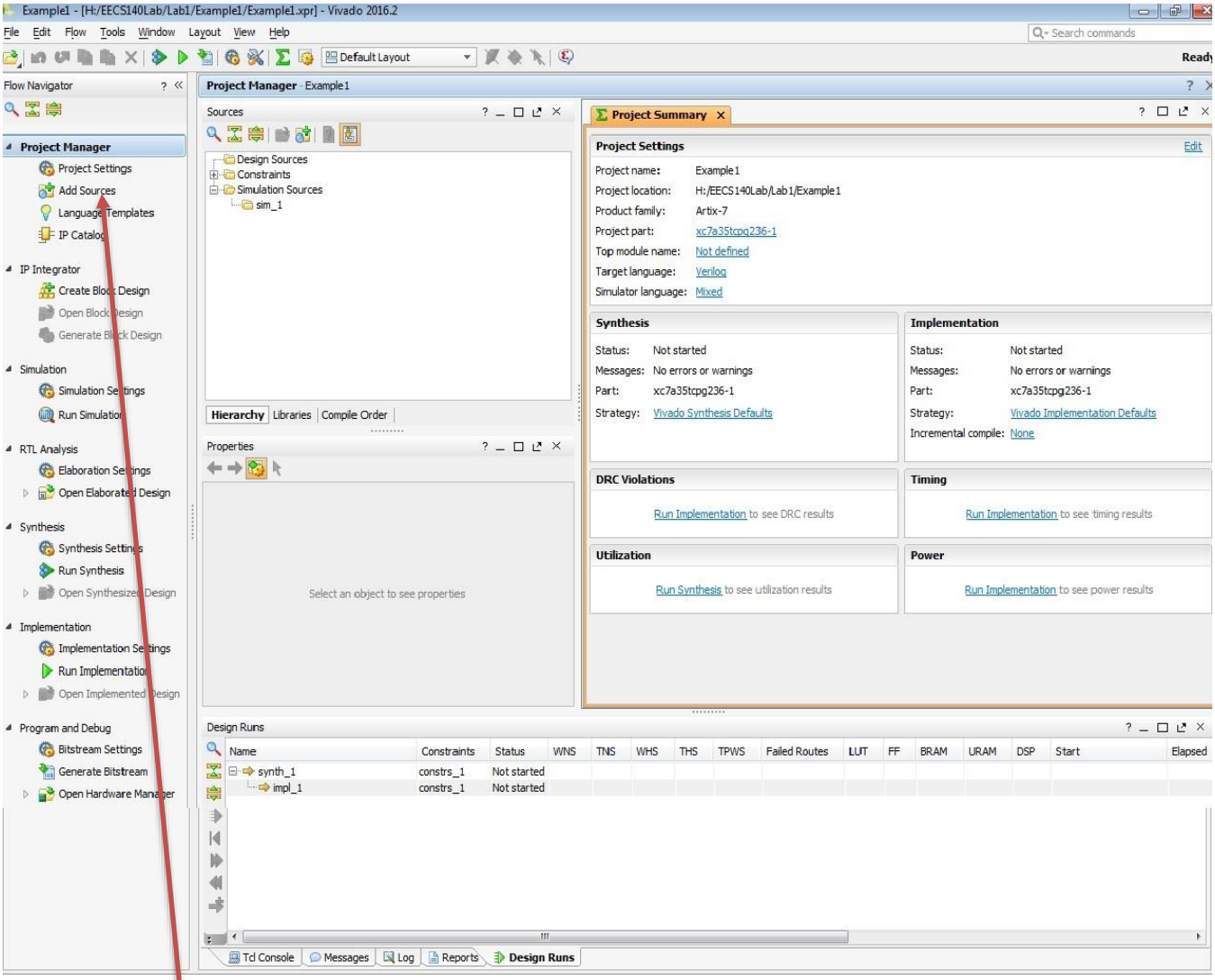


Click Next, and then Finish:



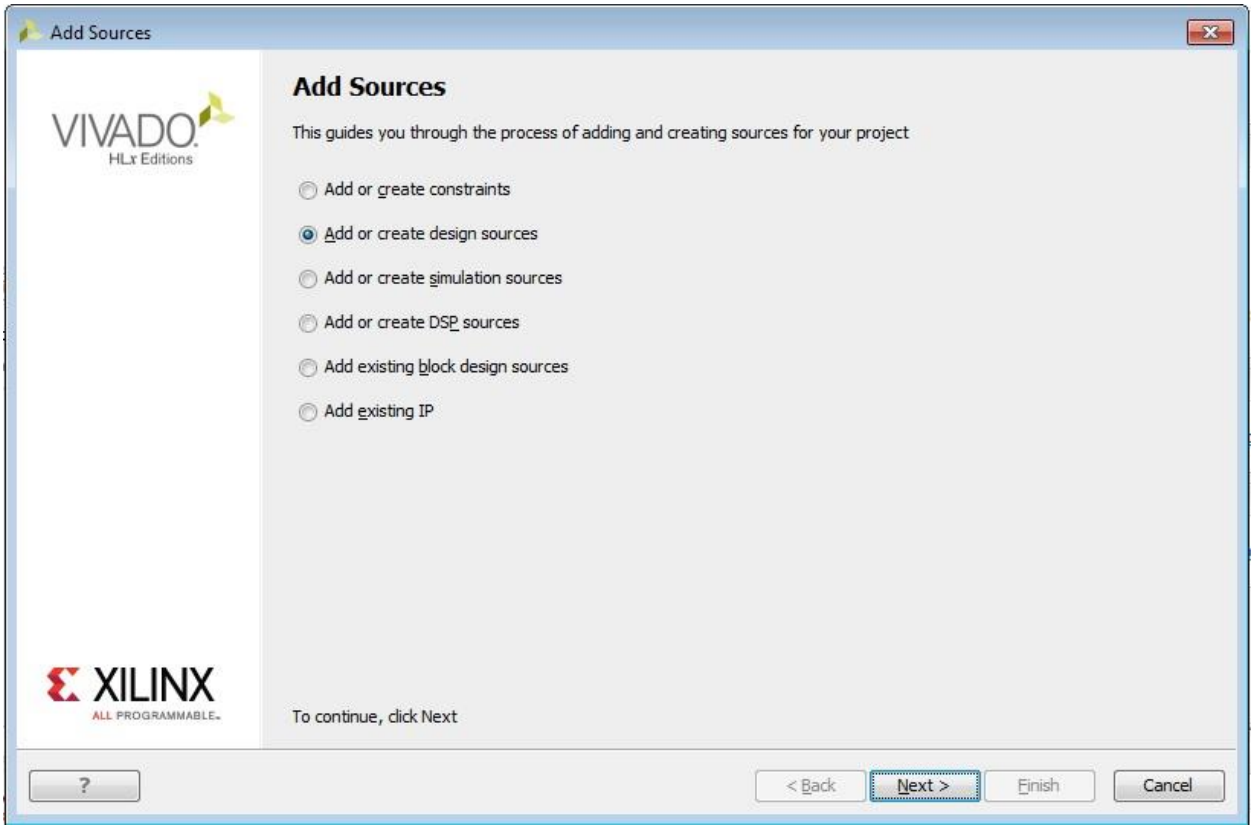
The Project window opens:





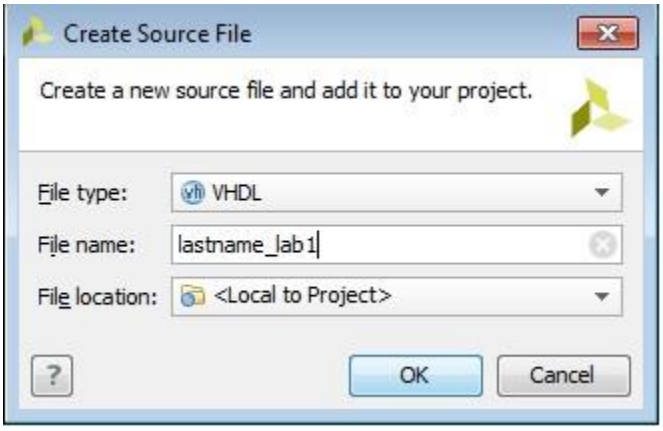
We now need to add a VHDL design source to describe our AND operation.

Click on Add Sources in the left Project Manager window (or select the menu item File => Add Sources)



Select Next,

And then select Create File (click on the + symbol) and enter a suitable file name:

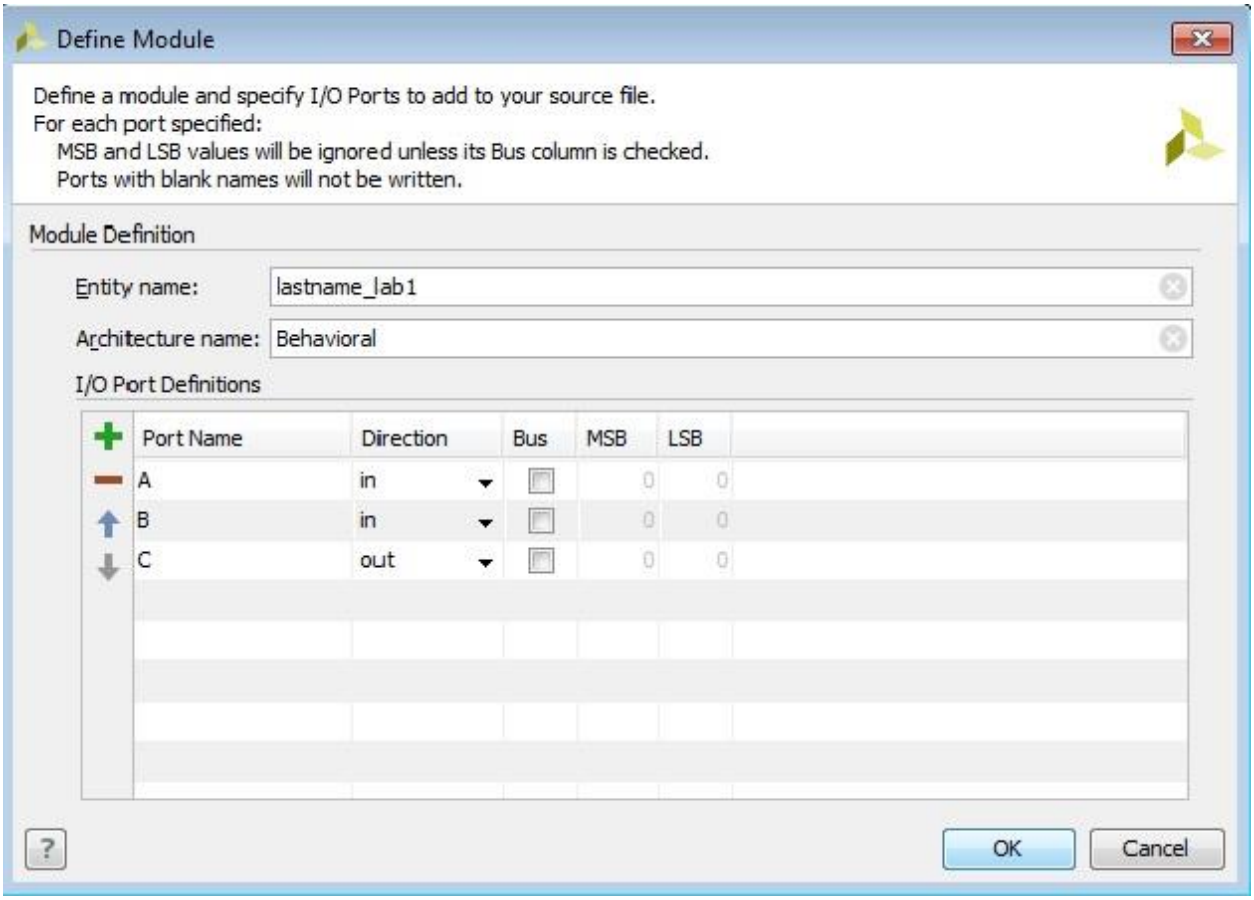


Then click OK and Finish.

Note: Do not start filename with numbers or special characters. Also, ensure that you do

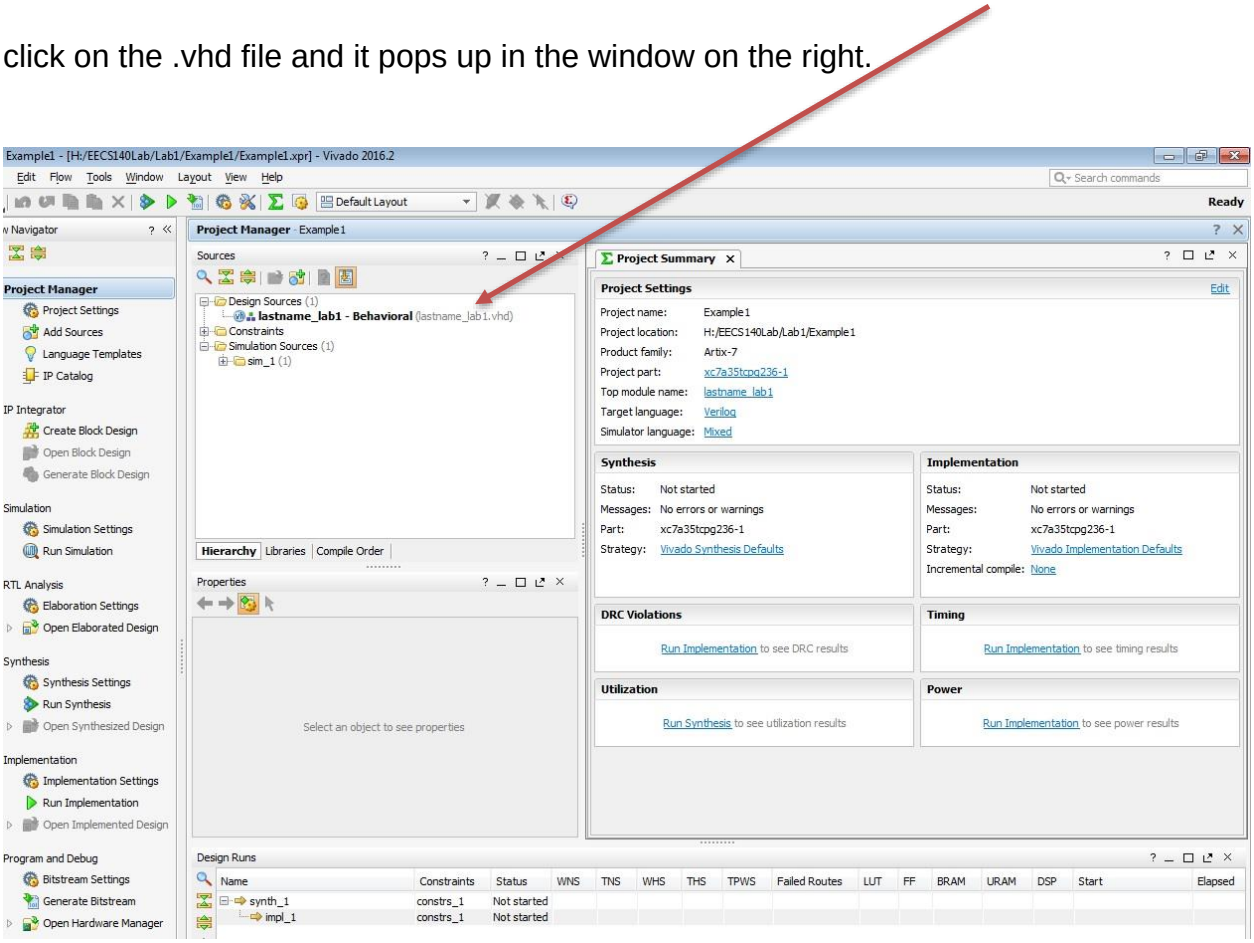
not have spaces in the file name.

We can now specify the inputs and outputs to create our 2 input AND gate. Use A, B as inputs and C as output.



Click OK.

Back in the Project Manager Sources window you can see the new **.vhd** file. Double click on the .vhd file and it pops up in the window on the right.



We can now add the VHDL statements to design our two input AND gate.

H:/EECS140Lab/Lab1/Example1/Example1.srcs/sources_1/new/lastname_lab1.vhd

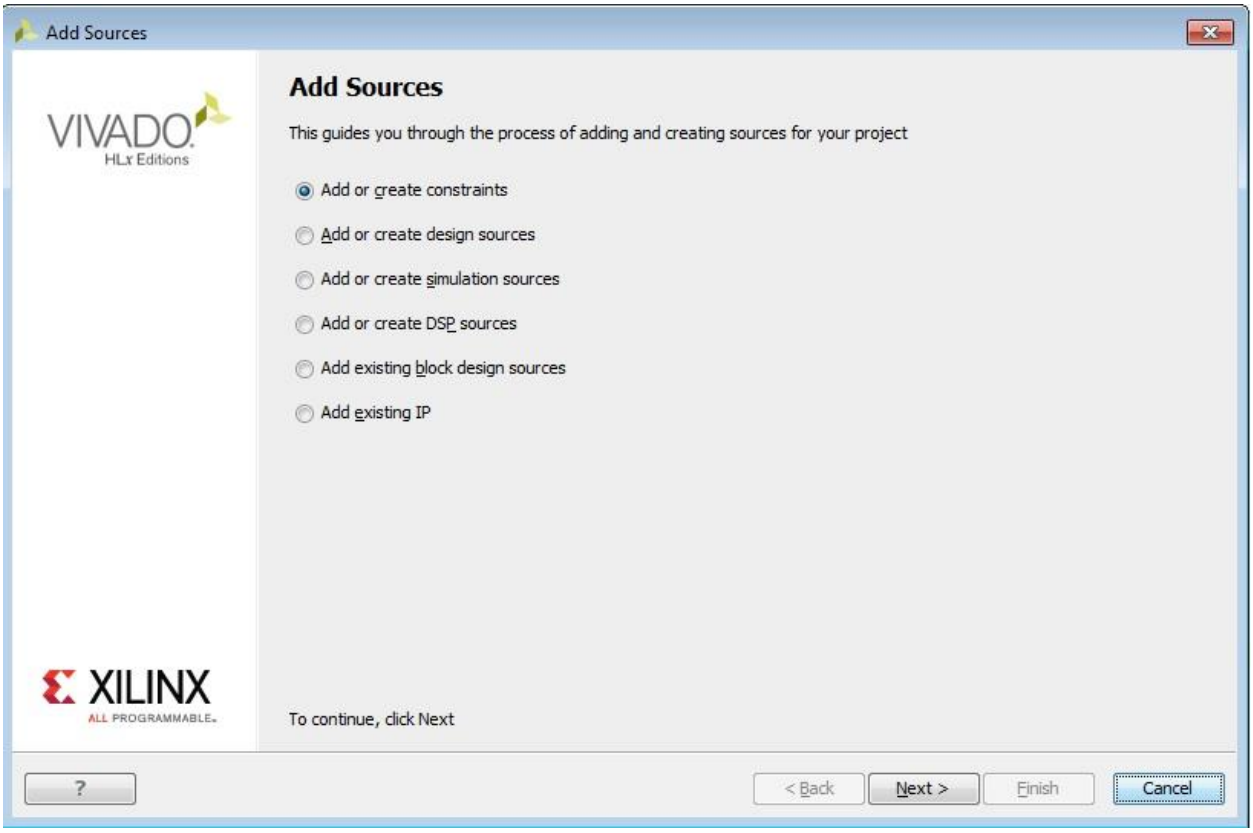
```
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity lastname_lab1 is
35      Port ( A : in STD_LOGIC;
36             B : in STD_LOGIC;
37             C : out STD_LOGIC);
38  end lastname_lab1;
39
40  architecture Behavioral of lastname_lab1 is
41
42  begin
43
44  C<= A and B;
45
46  end Behavioral;
47
```

A and **B** is being assigned to the output **C**

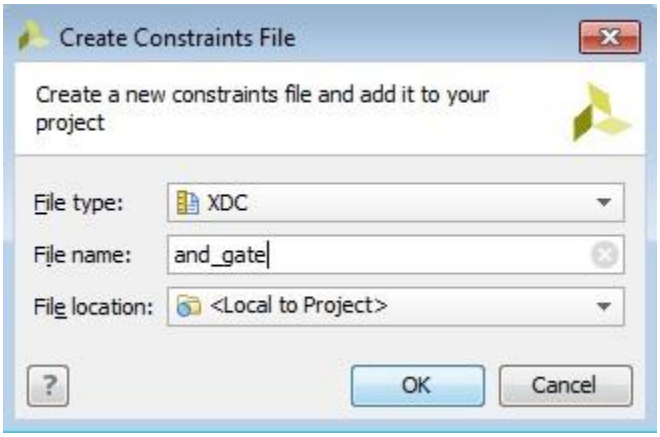
Let us now specify the FPGA pins that will be used for the Switch (SW) inputs and LED outputs.

We use the Basys3 manual to determine the FPGA pins.

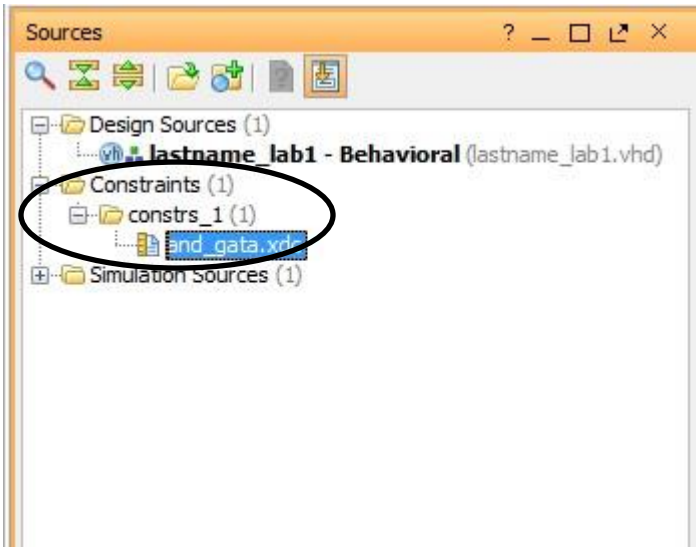
Next click Add Sources and select ‘Add or create constraints’:



We can name this and_gate



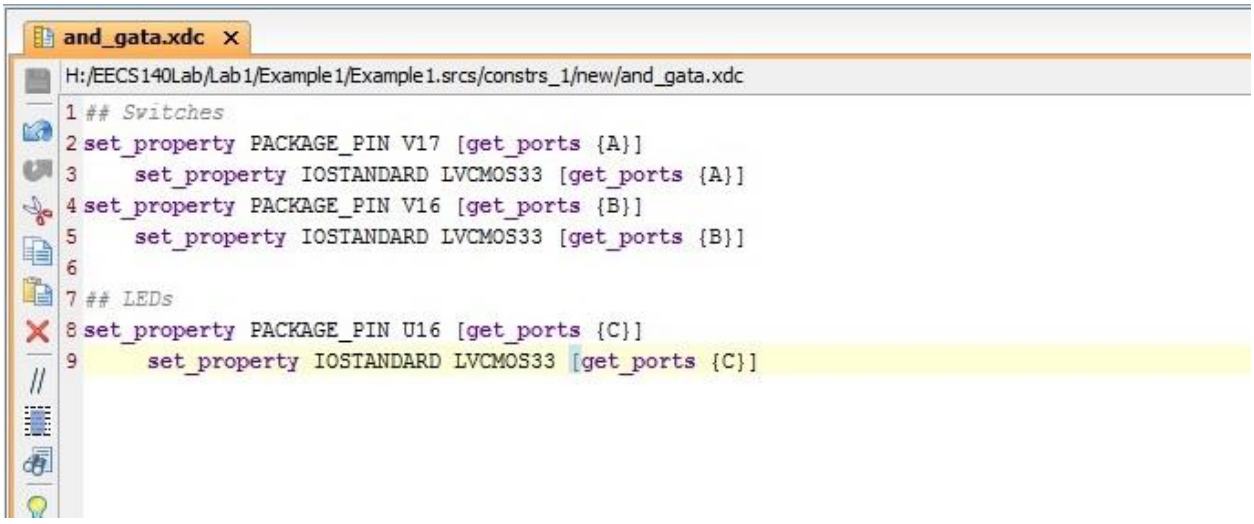
In the sources window, select the constraints file by the hierarchy Constraints => constrs_1 => and_gate.xdc



We can then add location constraints for all the inputs and outputs.

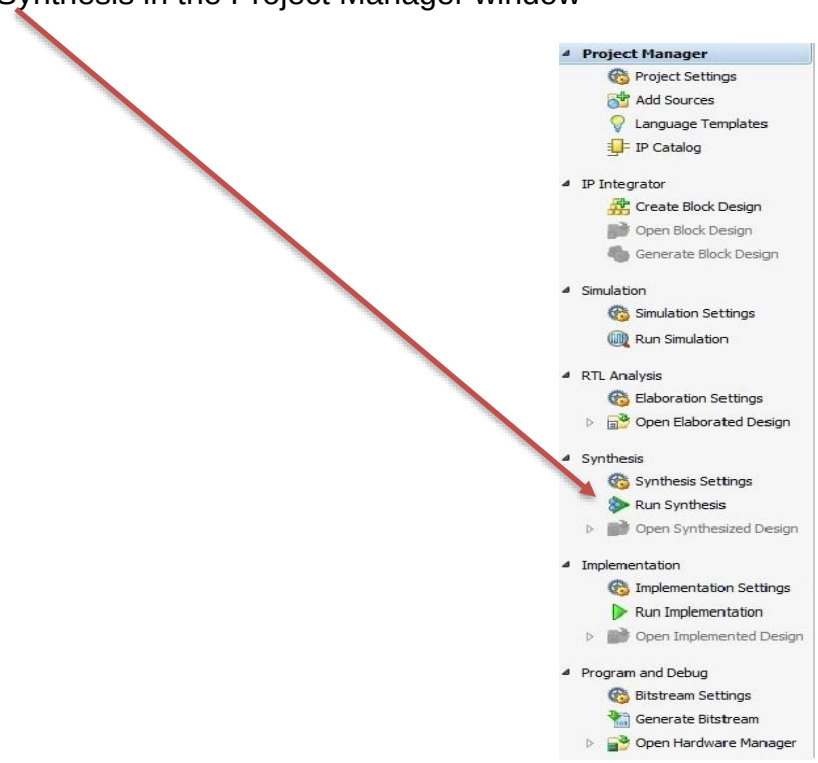
(you can download a copy of the Basys3 XDC constraints from the Digilent website (this is provided to you!!) – just copy the pins you are using for the design as shown below):

These constraints specify the pins to use for each signal and what type of interface.



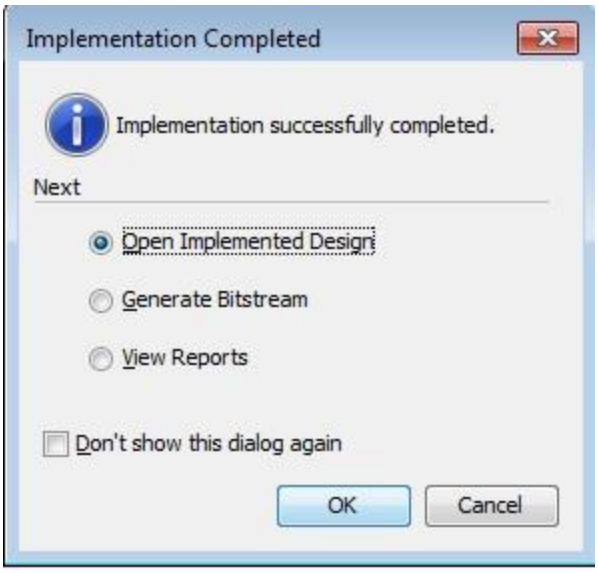
Let us now synthesize the design.

Click Run Synthesis in the Project Manager window



After synthesis is complete with no errors or warnings, you can see the following window.

Click on Run Implementation



On Successful completion of implementation, the above window pops out. Now, click on Generate Bitstream (as shown in the above window).

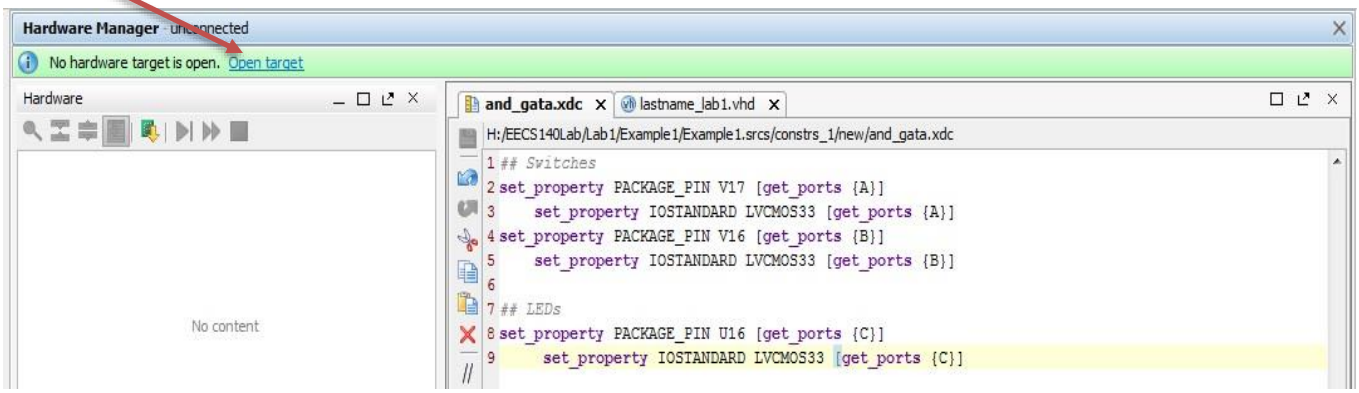
Once the Bitstream Generation is complete, you get to see the window shown below.

Plug in your device to the computer using the USB cable, and power on the board. An

orange LED blinks continuously which indicates that the device is busy.

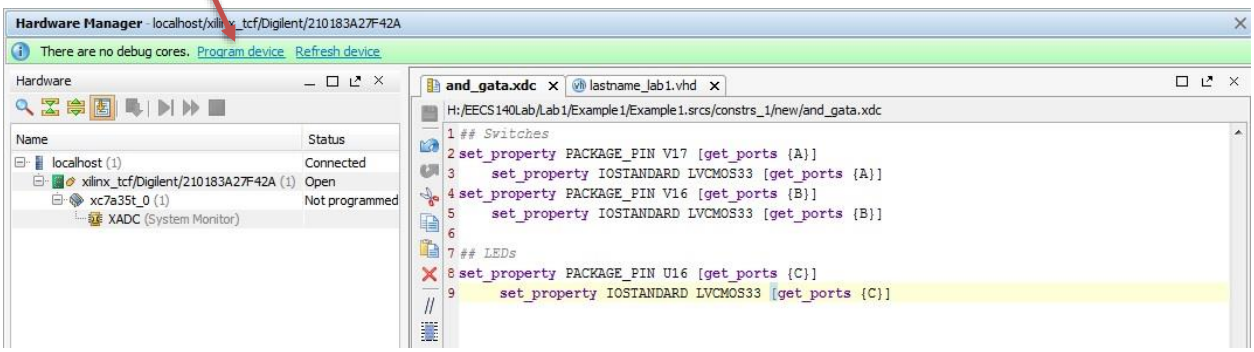
Now, Click on “Open Hardware Manager” from the above window.

The following window comes up after you click on Open Hardware Manager”. Click on Open target

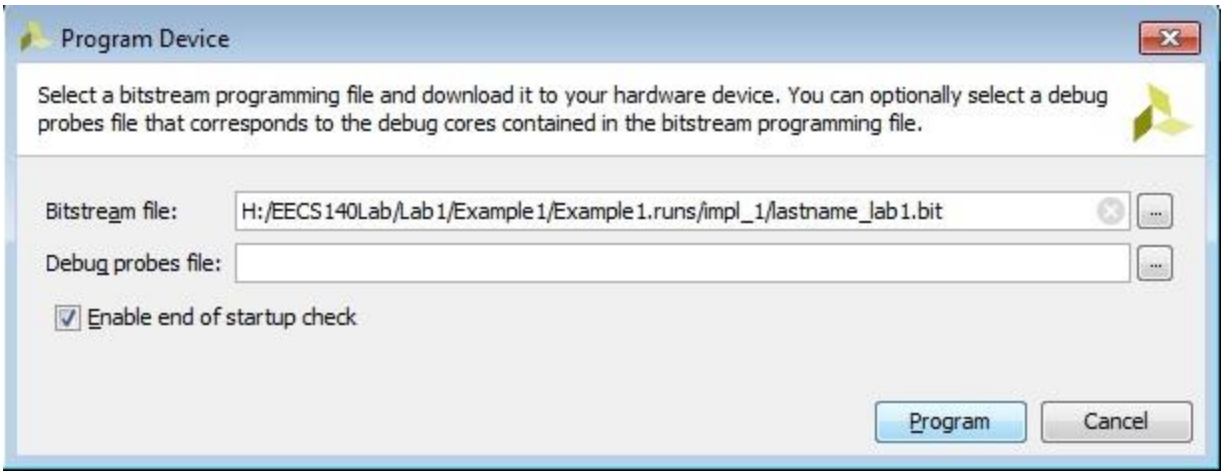


And then click Auto Connect. You see the window shown below.

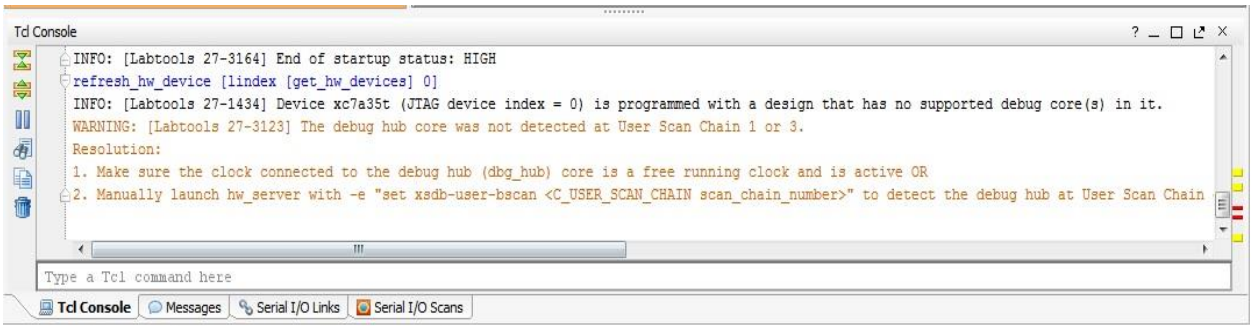
Click on Program Device and select the device (XC7A35T-1CPG236C).



The following window is displayed on the screen that shows the path of the Bitstream file. Click Program.



If you do not get any error messages on the Tcl console, shown below, then this is the right time for you to play with the device. This time, a green LED glows on the board.



Congratulations!!! You have successfully programmed your FPGA.

When it is time for you to leave the lab, consider the following guidelines:

Turn off your FPGA board, gently unplug the USB cable from the board and coil it.

Place the board in the box.

The DIGILENT BASYS 3 Board

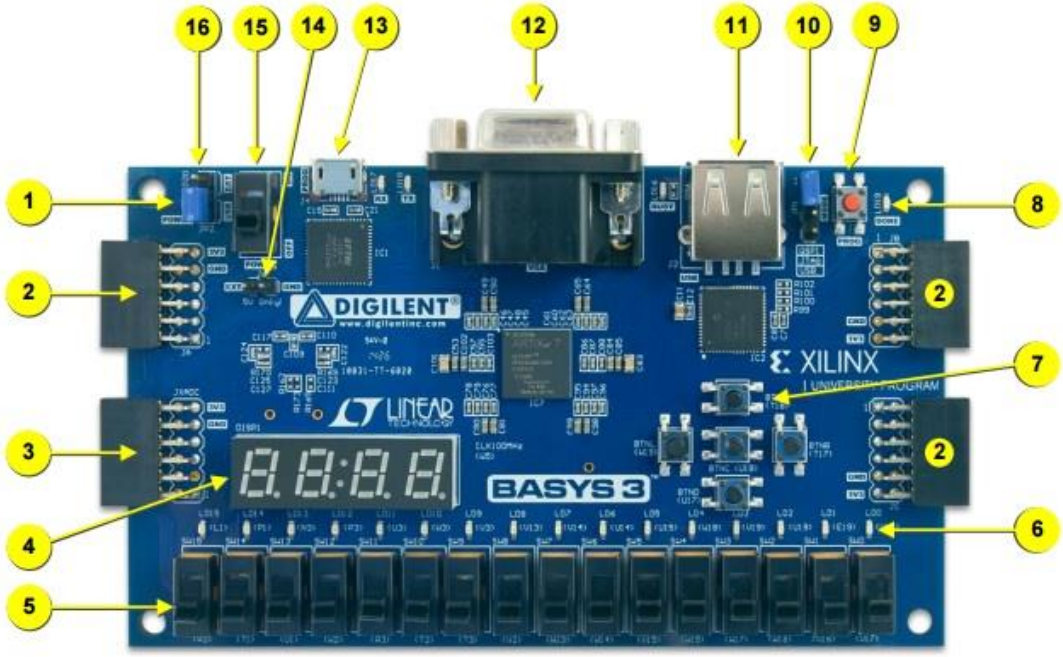


Figure 1. Basys 3 FPGA board with callouts.

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod port(s)	10	Programming mode jumper
3	Analog signal Pmod port (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Table 1. Basys 3 Callouts and component descriptions.