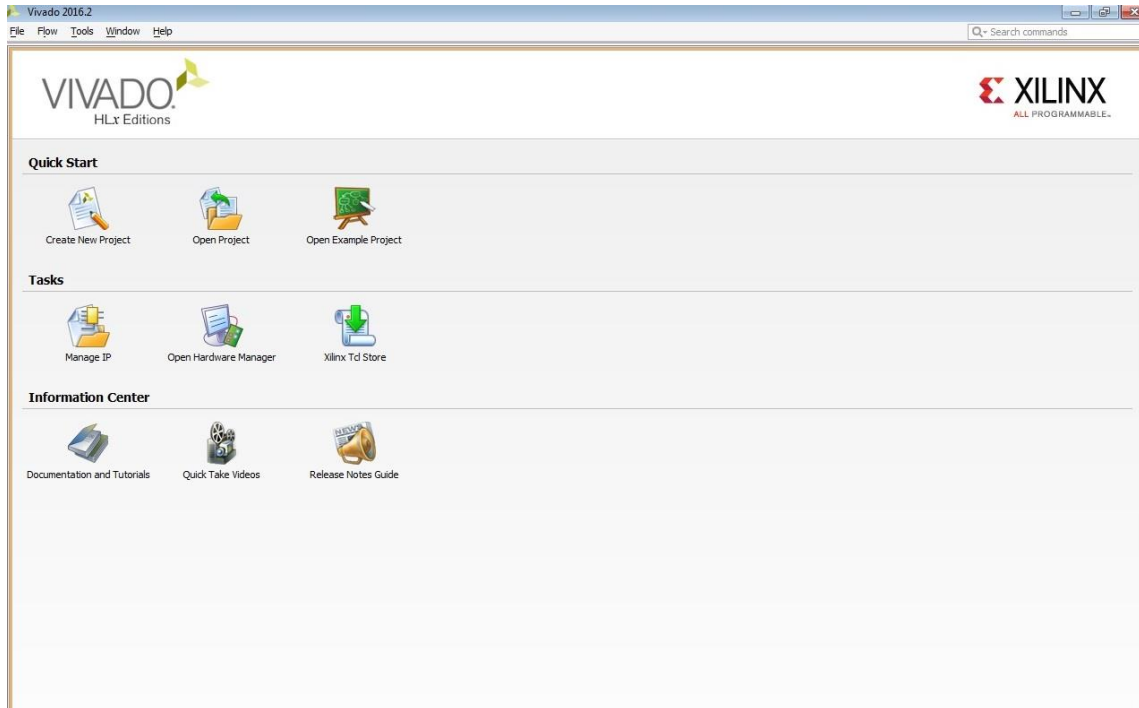


EECS 140/141 Lab 1

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

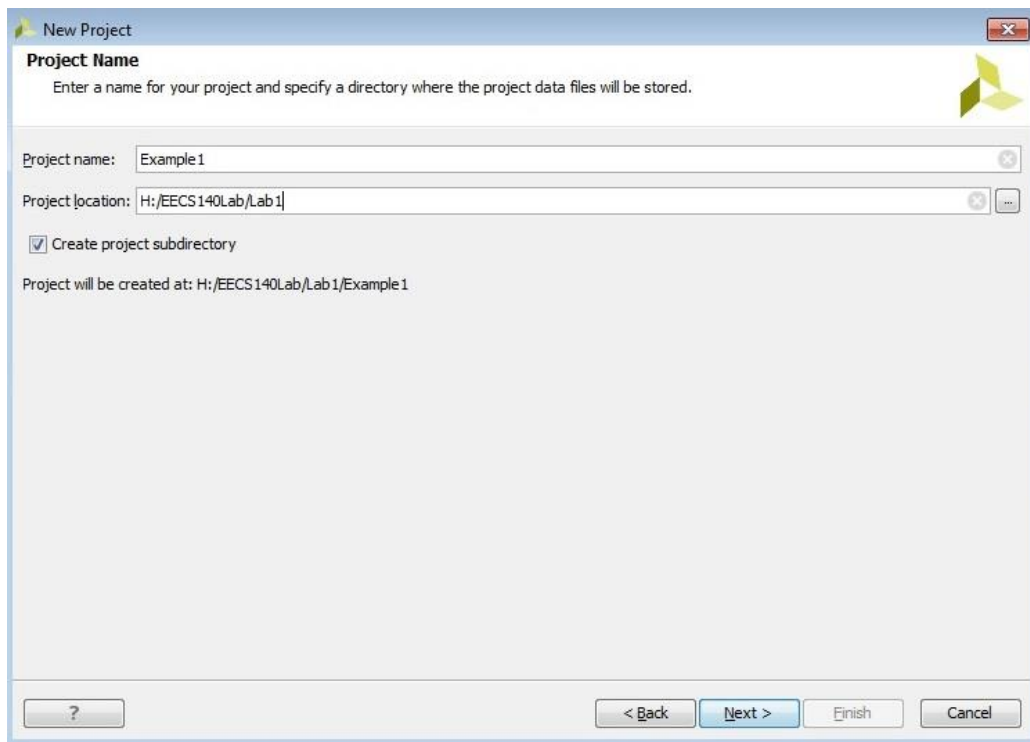
Start Vivado Design Suite:



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.



The 'New Project' dialog box is shown. It has a title bar with a green icon and a close button. The main area is titled 'Project Name' and contains the instruction 'Enter a name for your project and specify a directory where the project data files will be stored.' There are two text input fields: 'Project name:' with 'Example1' entered, and 'Project location:' with 'H:/EECS140Lab/Lab1' entered. A checkbox labeled 'Create project subdirectory' is checked. Below these fields, it says 'Project will be created at: H:/EECS140Lab/Lab1/Example1'. At the bottom, there are four buttons: a help button with a question mark, '< Back', 'Next >', 'Finish', and 'Cancel'.

New Project

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: Example1

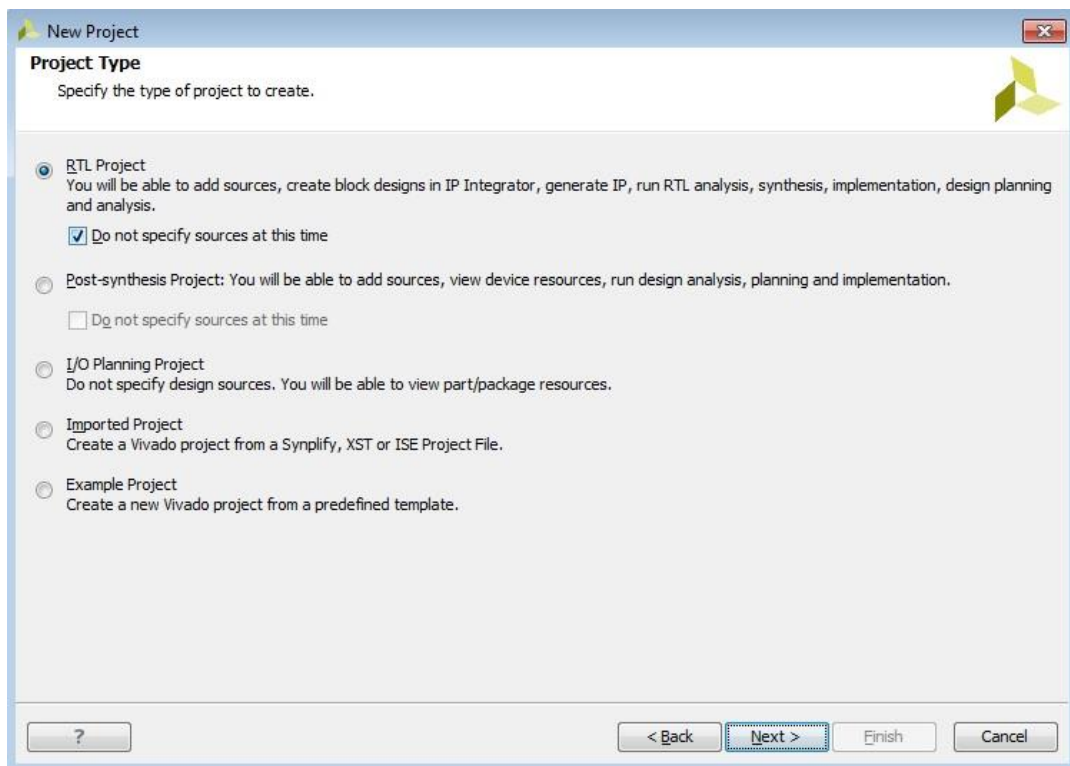
Project location: H:/EECS140Lab/Lab1

☒ Create project subdirectory

Project will be created at: H:/EECS140Lab/Lab1/Example1

? < Back Next > Finish Cancel

Click Next and select the RTL project type:



The 'New Project' dialog box is shown, now on the 'Project Type' tab. The title bar is the same. The main area is titled 'Project Type' and contains the instruction 'Specify the type of project to create.' There are five radio button options: 'RTL Project' (selected), 'Post-synthesis Project', 'I/O Planning Project', 'Imported Project', and 'Example Project'. Each option has a description. Under 'RTL Project', there is a checked checkbox 'Do not specify sources at this time'. At the bottom, there are four buttons: a help button with a question mark, '< Back', 'Next >', 'Finish', and 'Cancel'.

New Project

Project Type
Specify the type of project to create.

☒ **RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
☒ Do not specify sources at this time

☐ **Post-synthesis Project**: You will be able to add sources, view device resources, run design analysis, planning and implementation.
☐ Do not specify sources at this time

☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**
Create a Vivado project from a Synplify, XST or ISE Project File.

☐ **Example Project**
Create a new Vivado project from a predefined template.

? < Back Next > Finish Cancel

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

New Project

Default Part
Choose a default Xilinx part or board for your project. This can be changed later.

Select: ☒ Parts ☐ Boards

Filter

Product category: All Speed grade: -1
Family: Artix-7 Temp grade: C
Package: cpg236

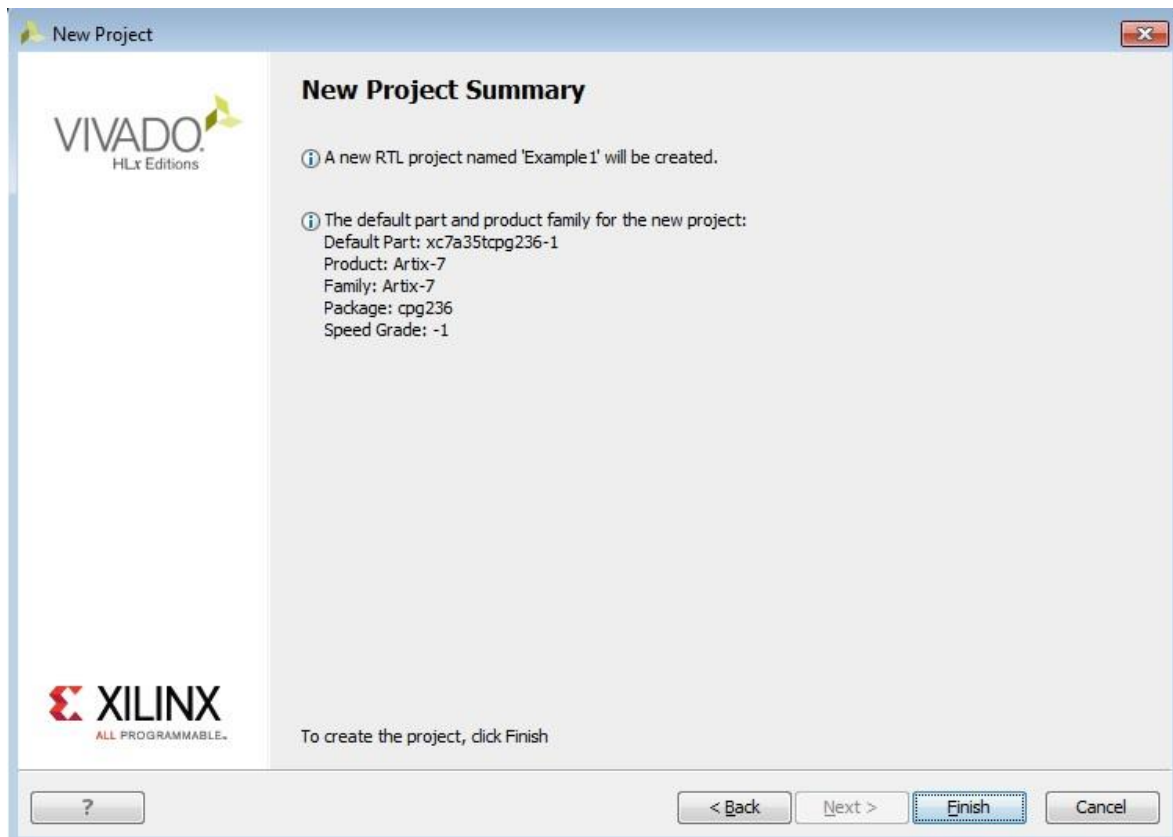
Reset All Filters

Search:

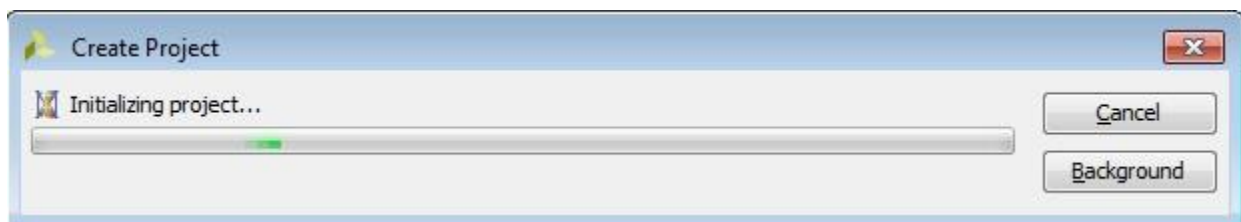
Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	Gb Transceivers	Available IOBs	LUT Elements
xc7a15tcpg236-1	236	25	45	20800	2	2	106	10400
xc7a35tcpg236-1	236	50	90	41600	2	2	106	20800
xc7a50tcpg236-1	236	75	120	65200	2	2	106	32600

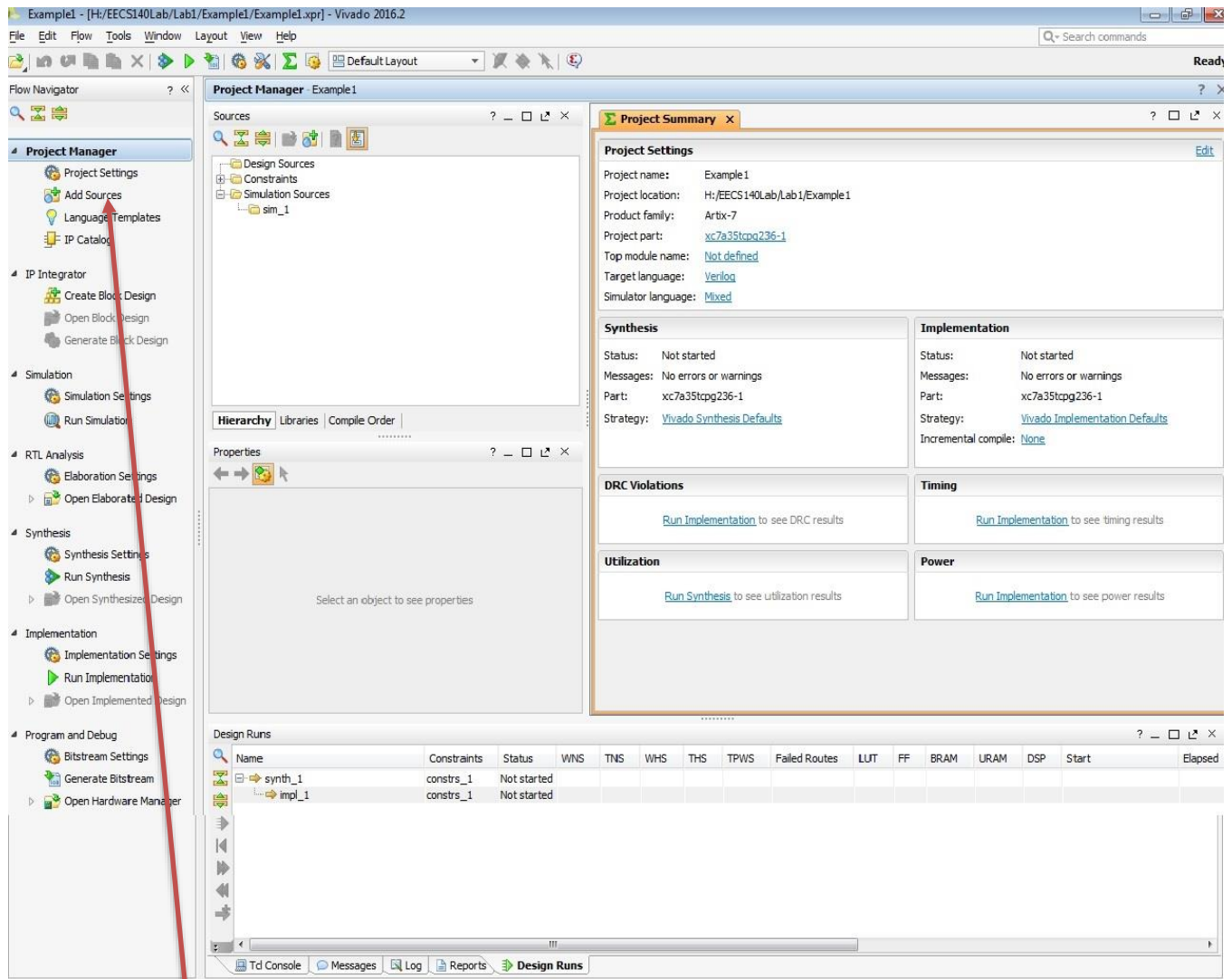
? < Back **Next >** Finish Cancel

Click Next, and then Finish:



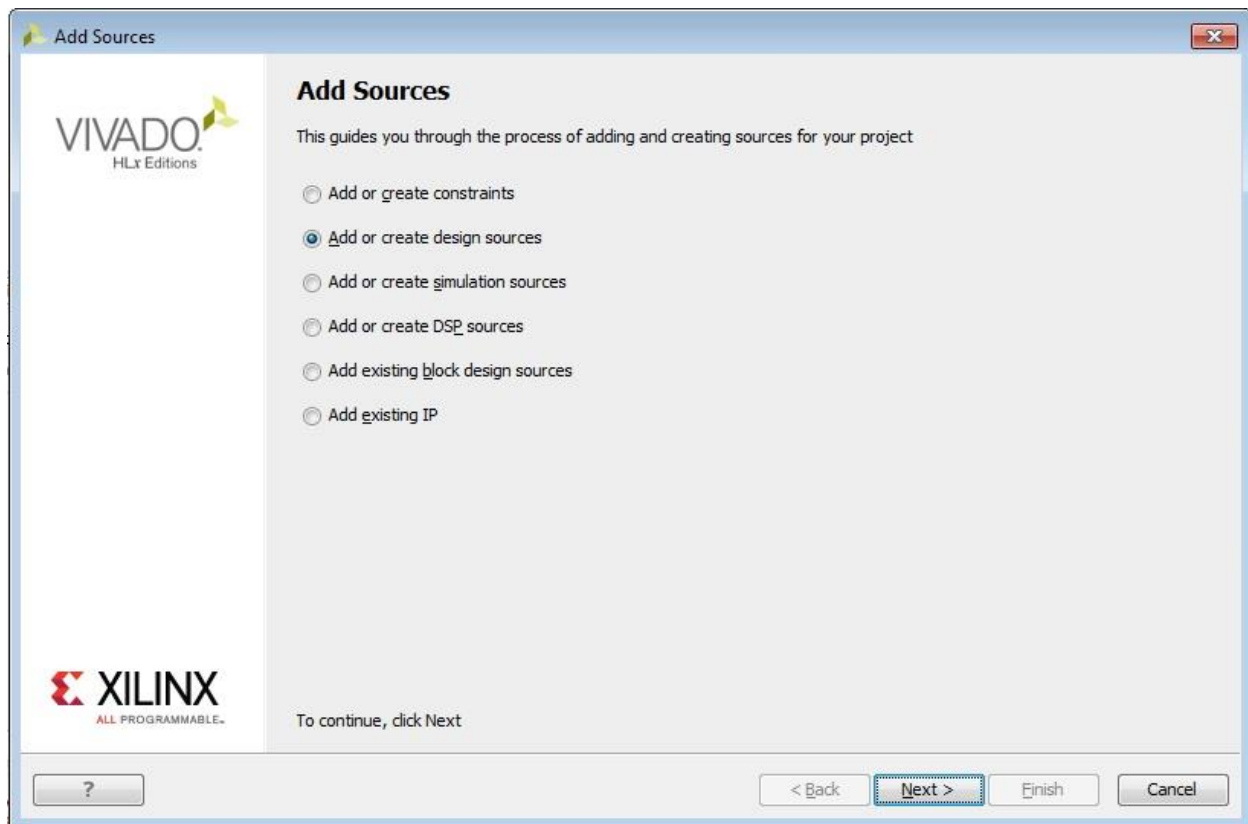
The Project window opens:





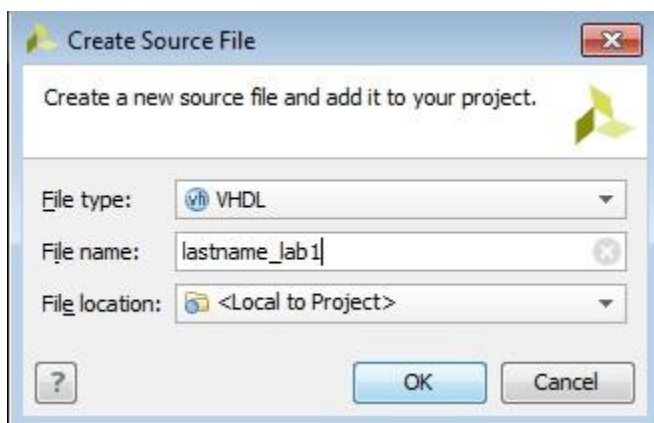
We now need to add a VHDL design source to describe our AND operation.

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:

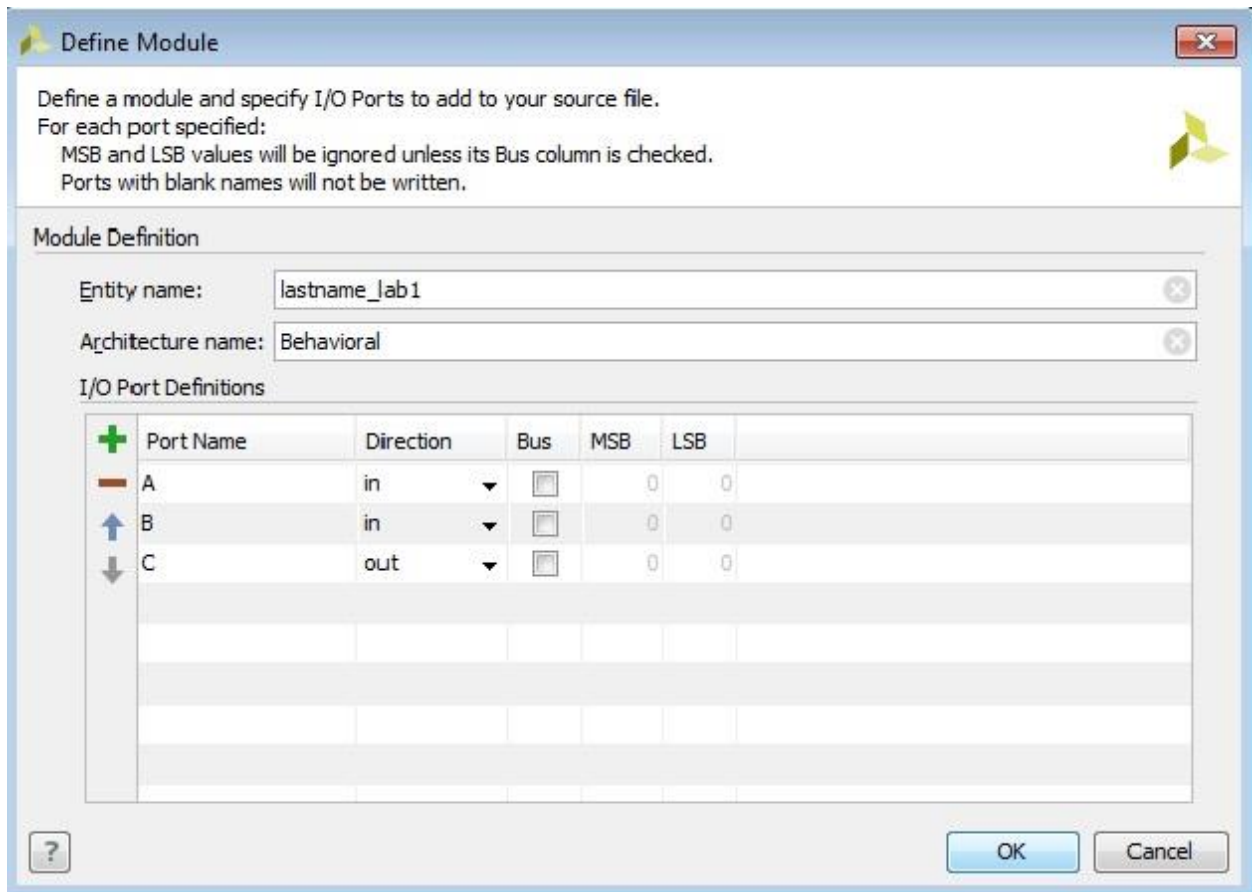


Then click OK and Finish.

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.



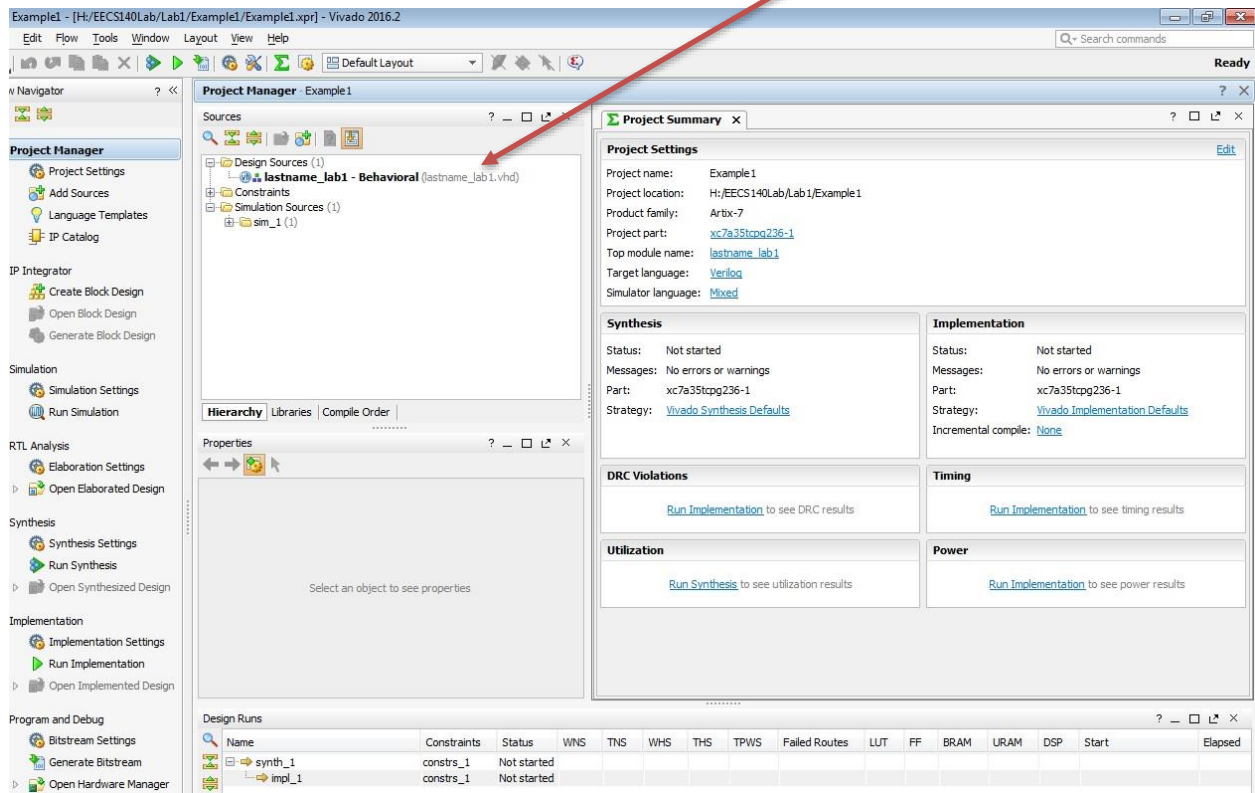
The 'Define Module' dialog box is shown. It contains instructions at the top: 'Define a module and specify I/O Ports to add to your source file. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written.' Below this is the 'Module Definition' section with 'Entity name' set to 'lastname_lab1' and 'Architecture name' set to 'Behavioral'. The 'I/O Port Definitions' section contains a table with three rows: A (input), B (input), and C (output). Each row has a 'Bus' checkbox, 'MSB' and 'LSB' value fields, and a 'Port Name' field.

+	Port Name	Direction	Bus	MSB	LSB
—	A	in	<input type="checkbox"/>	0	0
↑	B	in	<input type="checkbox"/>	0	0
↓	C	out	<input type="checkbox"/>	0	0

Buttons: ? OK Cancel

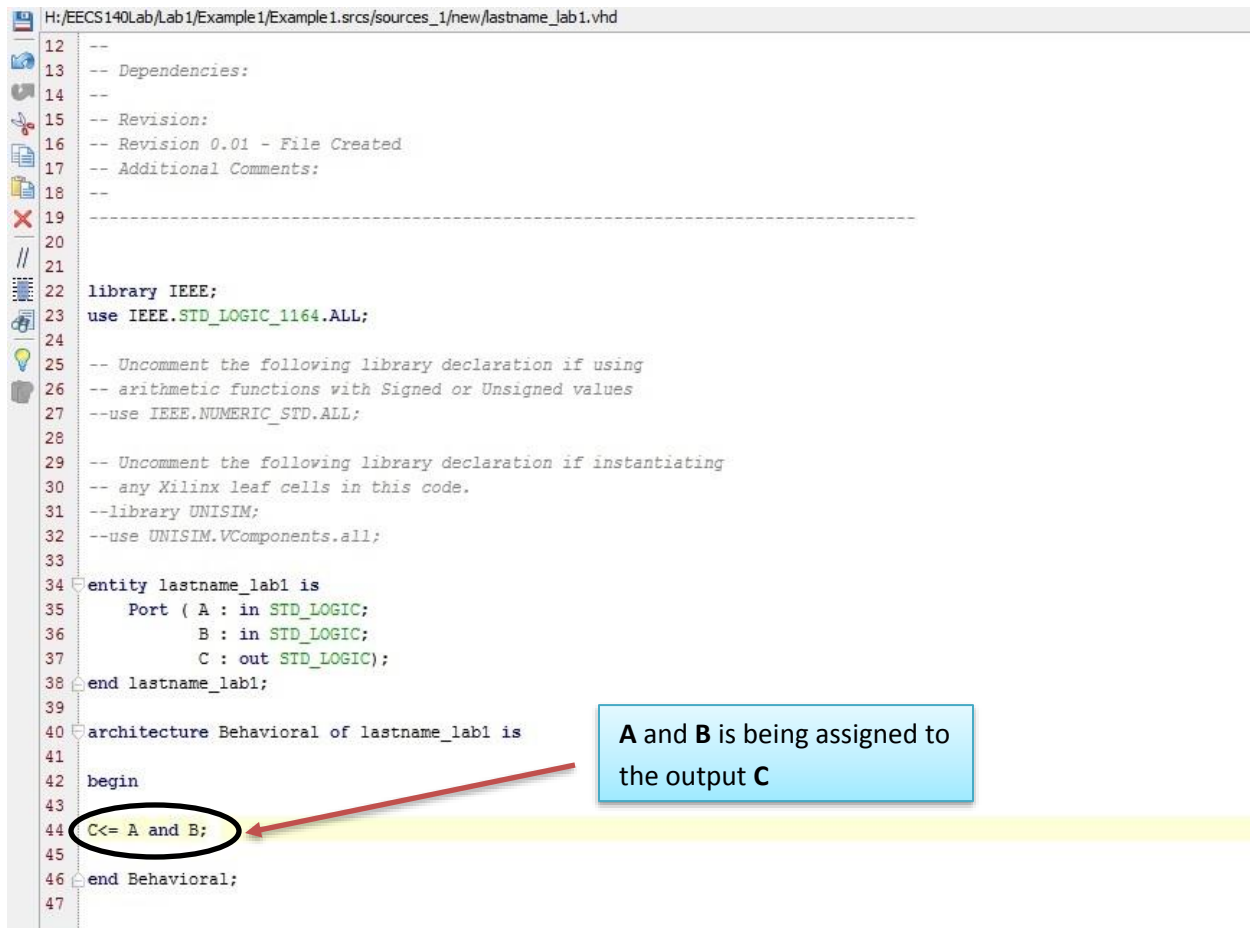
Click OK.

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
```

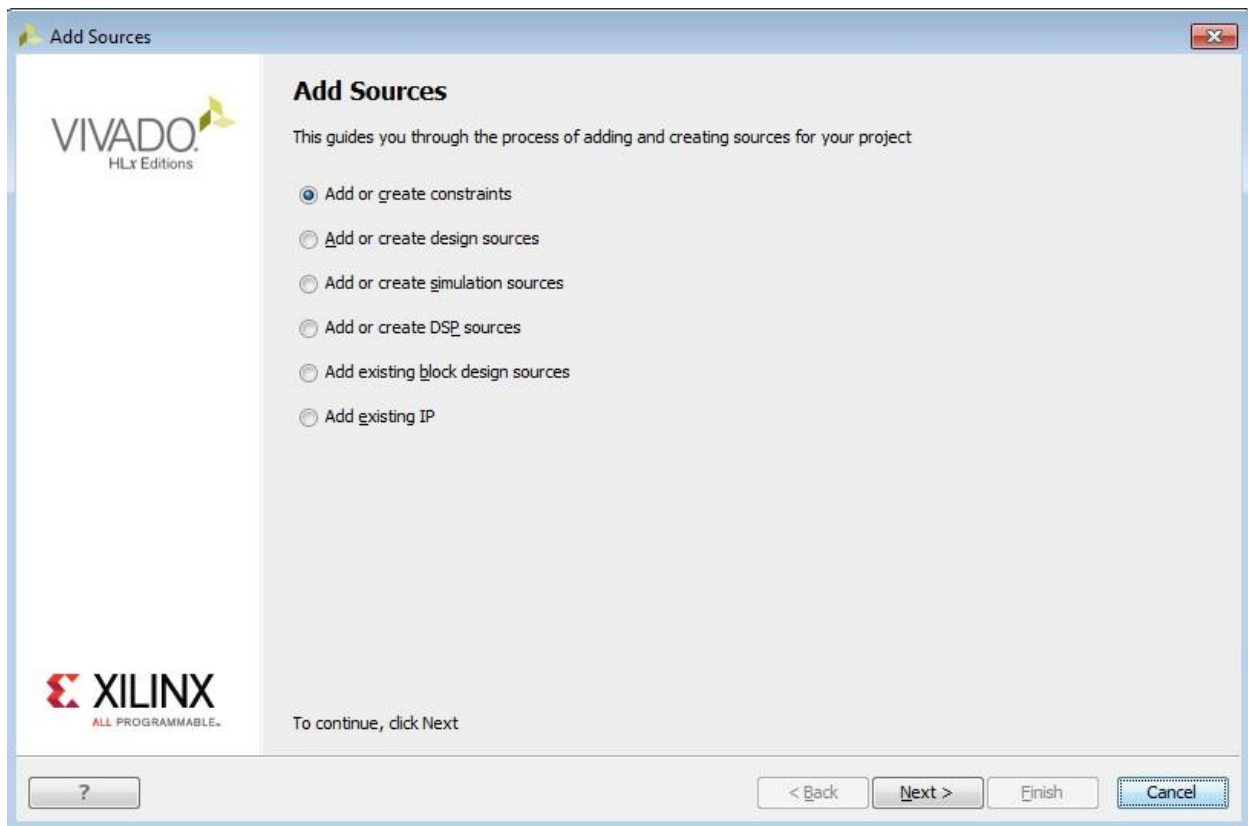


A and B is being assigned to the output C

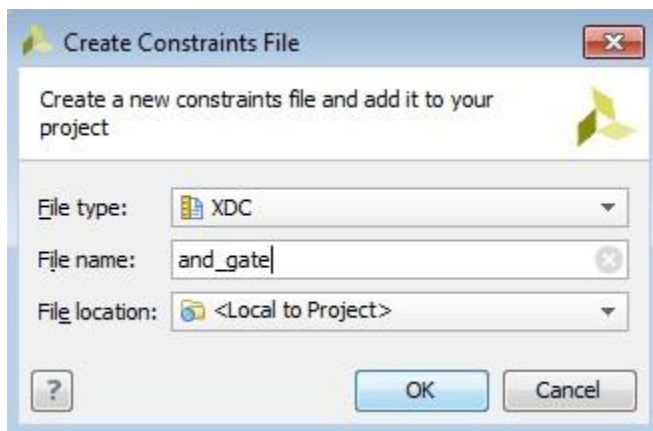
Let us now specify the FPGA pins that will be used for the Switch (SW) inputs and LED outputs.

We use the Basys3 manual to determine the FPGA pins.

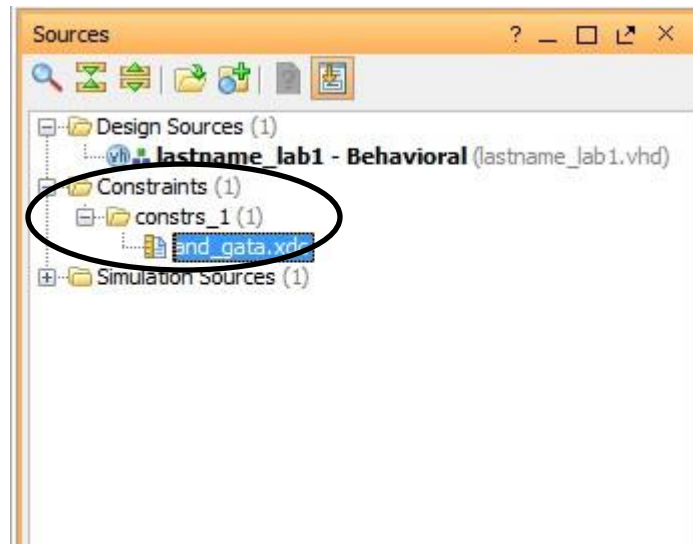
Next click Add Sources and select 'Add or create constraints':



We can name this and_gate



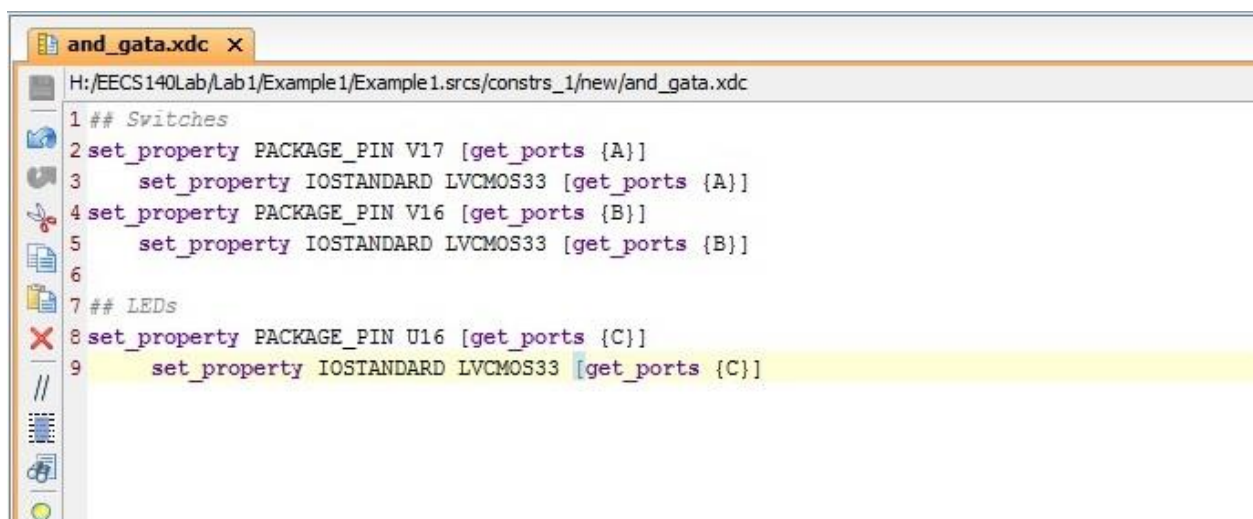
In the sources window, select the constraints file by the hierarchy Constraints => constrs_1 => and_gate.xdc



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

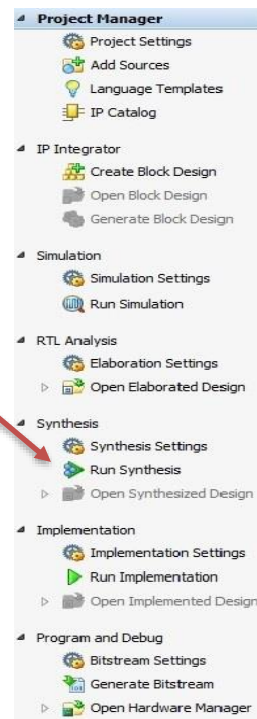
(you can download a copy of the Basys3 XDC constraints from the Digilent website (this is provided to you!!) – just copy the pins you are using for the design as shown below):

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



Let us now synthesize the design.

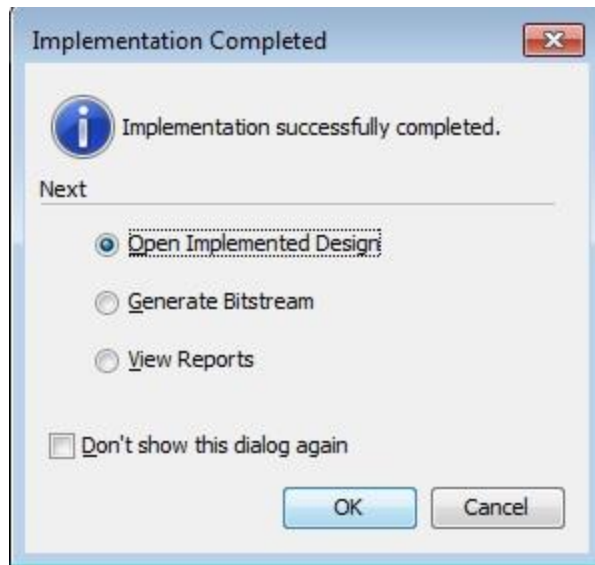
Click Run Synthesis in the Project Manager window



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.

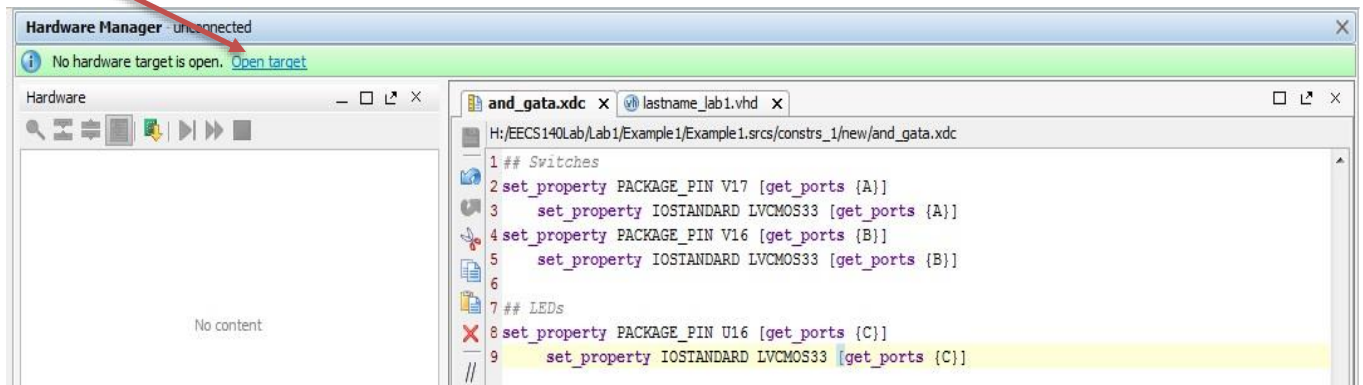


Plug in your device to the computer using the USB cable, and power on the board. An

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

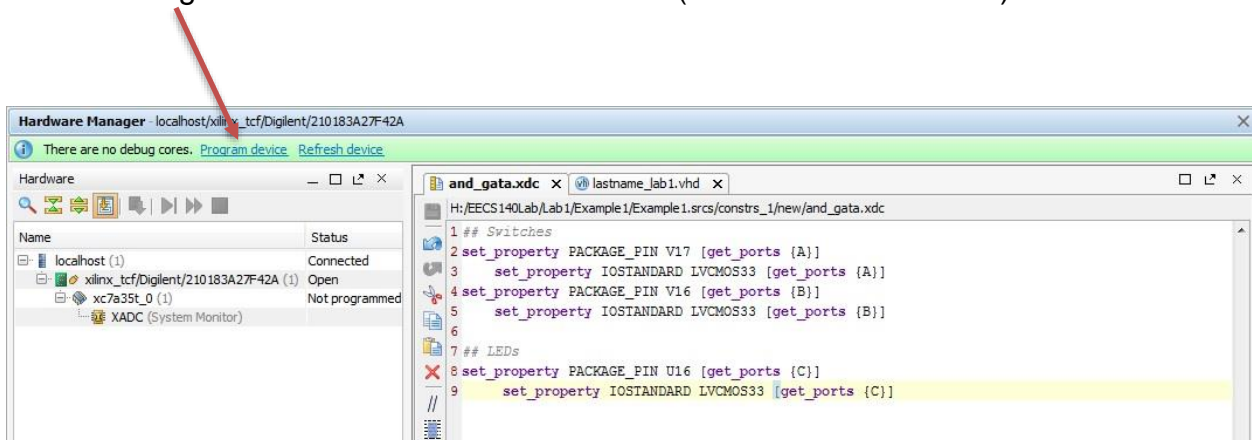
Now, Click on “Open Hardware Manager” from the above window.

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

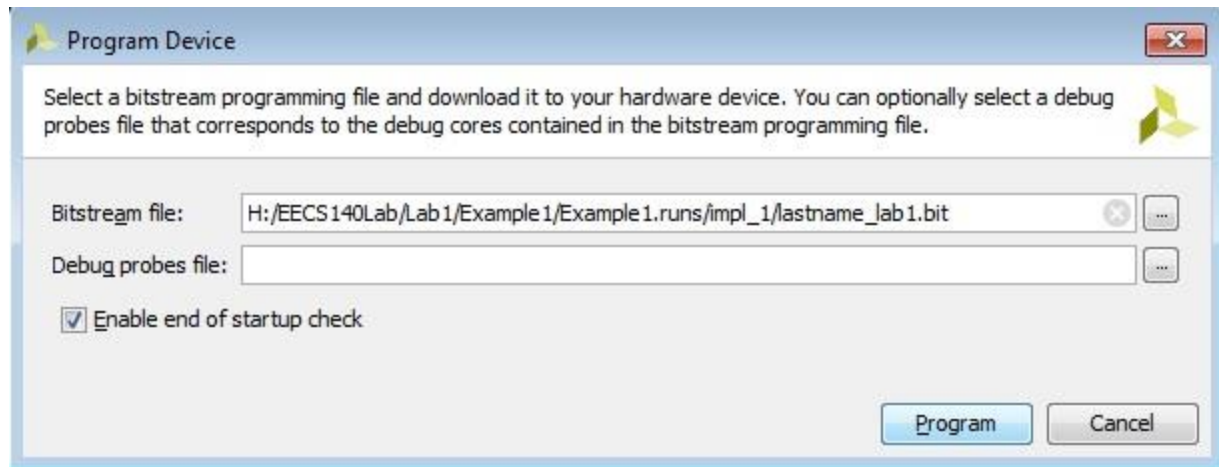


And then click Auto Connect. You see the window shown below.

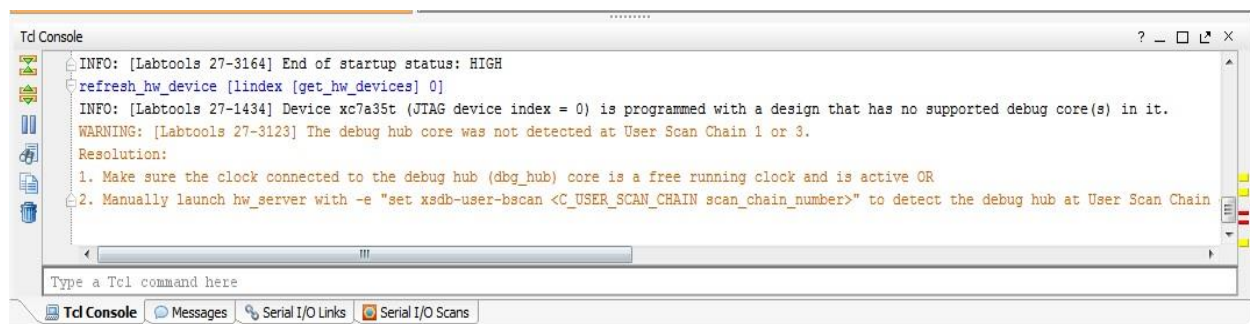
Click on Program Device and select the device (XC7A35T-1CPG236C).



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

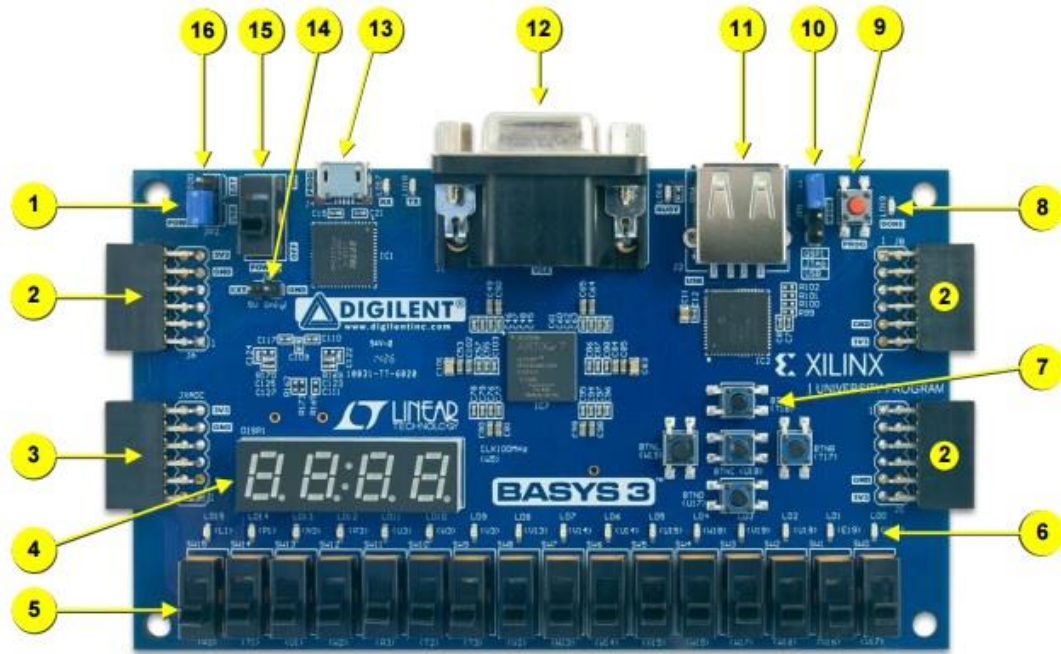


Figure 1. Basys 3 FPGA board with callouts.

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.