LAB1: *BCD to Decimal using FPGA*

OBJECTIVES:

1. Familiarization with Vivado software.
2. Design a BCD to Decimal decoder and implement it in an FPGA using the BASYS3 Artix-7 FPGA Trainer Board.

EQUIPMENTS:

Basys3 Artix-7 FPGA Trainer Board
Vivado Software – Current Version 2019.2

INTRODUCTION:

The Basys3 board is a complete, ready-to-use digital circuit development platform based on the Artix-7™ Field Programmable Gate Array (FPGA) from Xilinx. With its high-capacity FPGA (Xilinx part number XC7A35T-1CPG236C), low overall cost, and collection of USB, VGA, and other ports, the Basys3 can host designs ranging from introductory combinational circuits to complex sequential circuits like embedded processors and controllers. It includes enough switches, LEDs and other I/O devices to allow a large number of designs to be completed without the need for any additional hardware.

In this laboratory session, we will be designing a BCD to decimal decoder by programming the FPGA chip on the Basys3 board as shown in **Figure 1**. The BCD input will be via the four data switches on the bottom right corner and the decimal output will be displayed on the ten LEDs right above the data switches. If the BCD input is "0000" (switch position is down for logic '0'), LED0 will lit. If the BCD input is "1001" (switch position is up for logic '1'), LED9 will lit etc.

The simplified design flow for designing a digital logic circuit is shown in **Figure 2**.

PROCEDURE:

1 Copy relevent files from Blackboard

1.1 Create a new folder **D:/ET0901>Lab>Lab1** in your computer. (If your computer have only one drive, then use that drive instead of D Drive)

1.2 Download the following three files from **Blackboard➔Learning Resources➔Lab** and copy into the new folder you have just created.

- Basys3_Master.xdc
- BCD2dec.v
- BCD2dec_tb.v

1.3 Make a copy of the *Basys3_Master.xdc* file and rename it to BCD2dec.xdc. Make sure you don't make any changes to the master file in future.

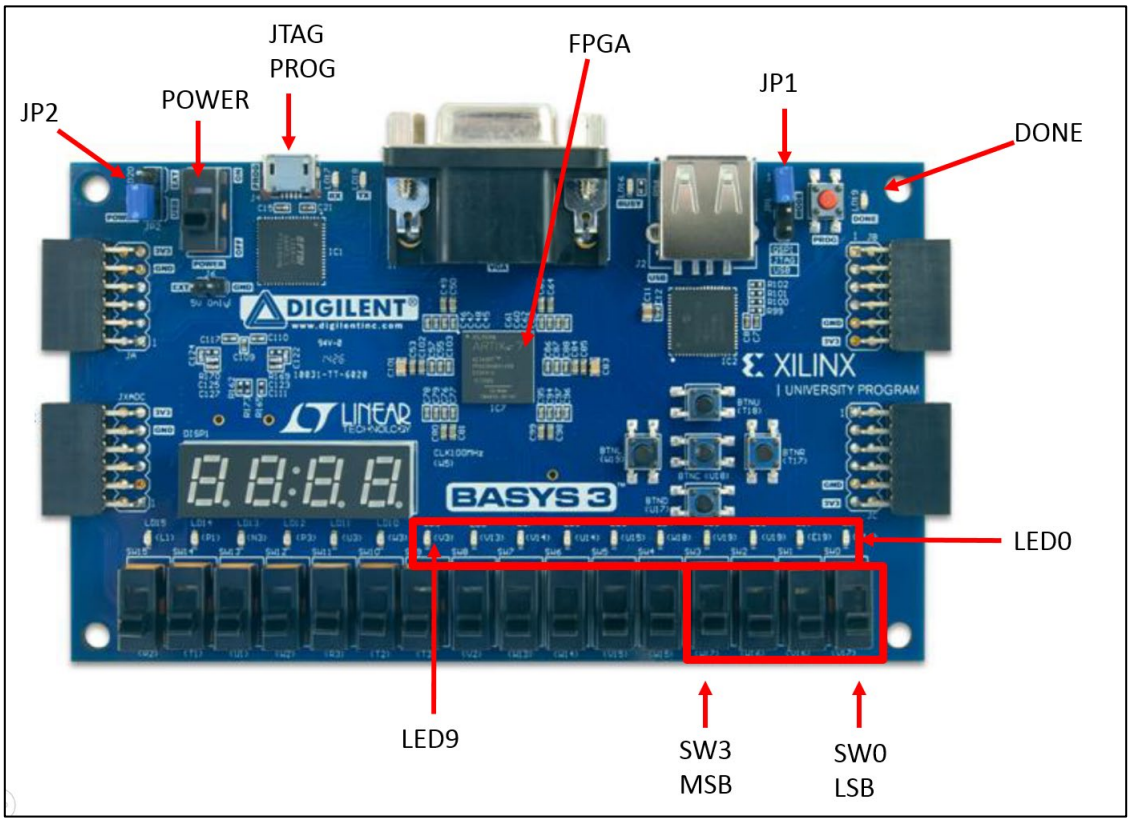


Figure 1: Basys3 Artix-7 FPGA Trainer Board layout

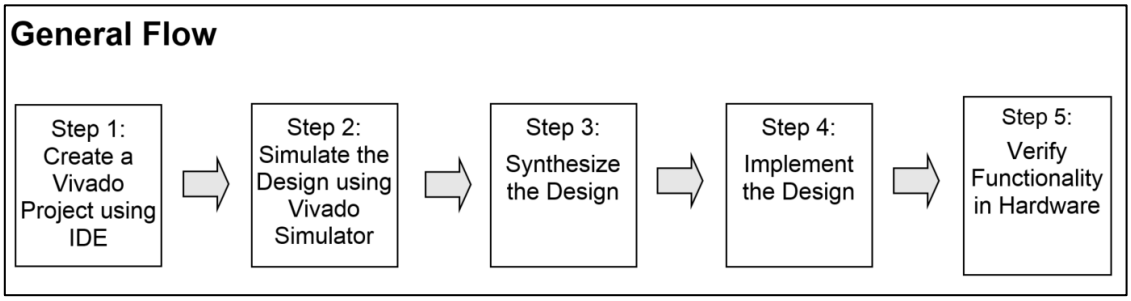


Figure 2: Simplified design flow for designing digital logic circuits on FPGA Board

2 Launch Vivado and create a project targeting XC7A35TCPG236-1 and using the Verilog HDL

2.1 Open Vivado 2019.2 by left click on this icon, 

on your desktop. (If you don't have this icon on your desktop, ask your lecturer to help you create a shortcut)

2.2 Click **Create New Project** to start the wizard. You will see *Create A New Vivado Project* dialog box. Click **Next**.

2.3 Click the Browse button of the *Project location* field of the New Project form, browse to **D:/ET0901>Lab>Lab1** and click Select.

2.4 Enter **lab1** in the *Project name* field. Make sure that the *Create Project Subdirectory* box is checked. Click **Next**.

2.5 Select **RTL Project** option in the *Project Type* form, then uncheck the box of "Do not specify sources at this time", and then click **Next**.

2.6 Using the drop-down buttons, select **Verilog** as the *Target Language* and *Simulator Language* in the *Add Sources* form as shown in **Figure 3**. (Verilog is a hardware description language which is covered in Chapter 3)

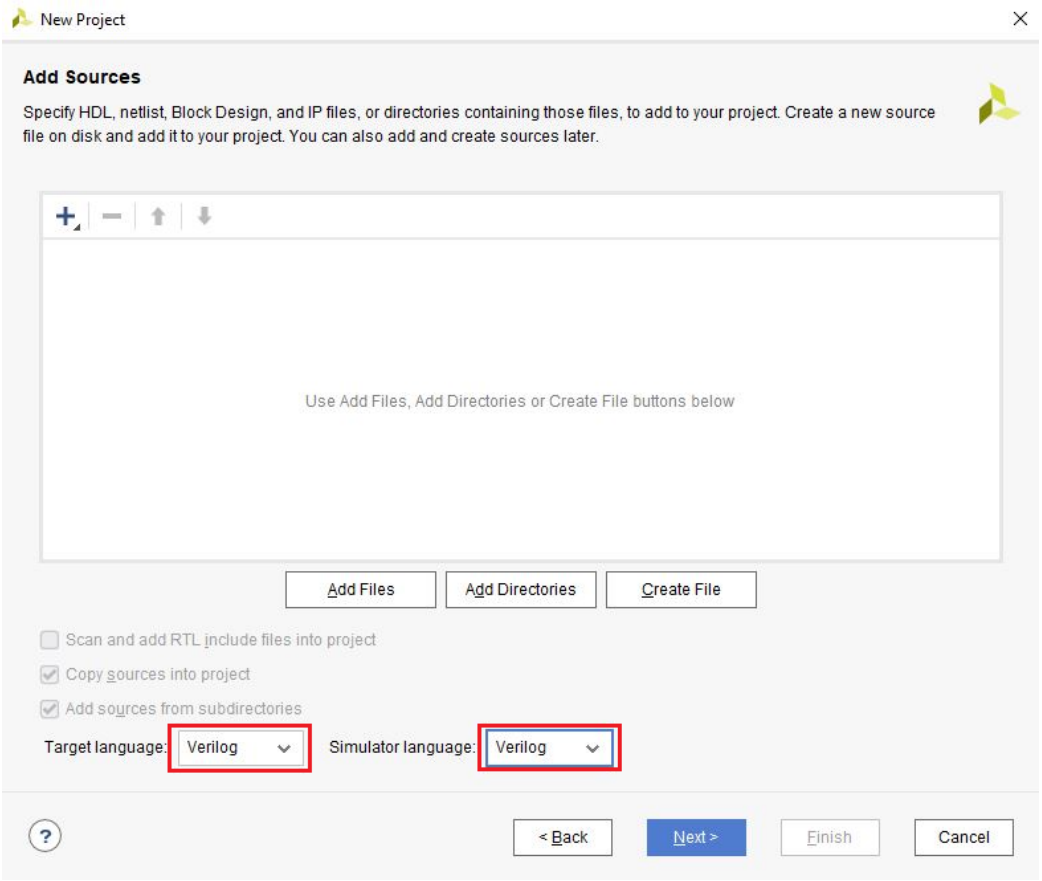


Figure 3: Selecting Language in Add Source form

2.7 Click the **Add Files** box and browse to the **D:/ET0901>Lab>Lab1** directory, select *BCD2dec.v,* click **OK**. If it isn't already checked, check **Copy sources into project** and then click **Next** to get to the *Add Constraints* form.

2.8 Click on the **Add Files** box and browse to the **D:/ET0901>Lab>Lab1** directory, select *BCD2dec.xdc* and click **OK** (if necessary), and then click **Next.** This Xilinx Design Constraints file assigns the physical input-output locations on FPGA to the switches and LEDs located on the board. This information can be obtained in Appendix 1.

2.9 In the *Default Part* form, use the **Parts** option and the **Search** function as shown in **Figure 4**, select the **xc7a35tcpg236-1** part**.** This is the FPGA chip on the Basys3 board that we will be programming in Step-6.

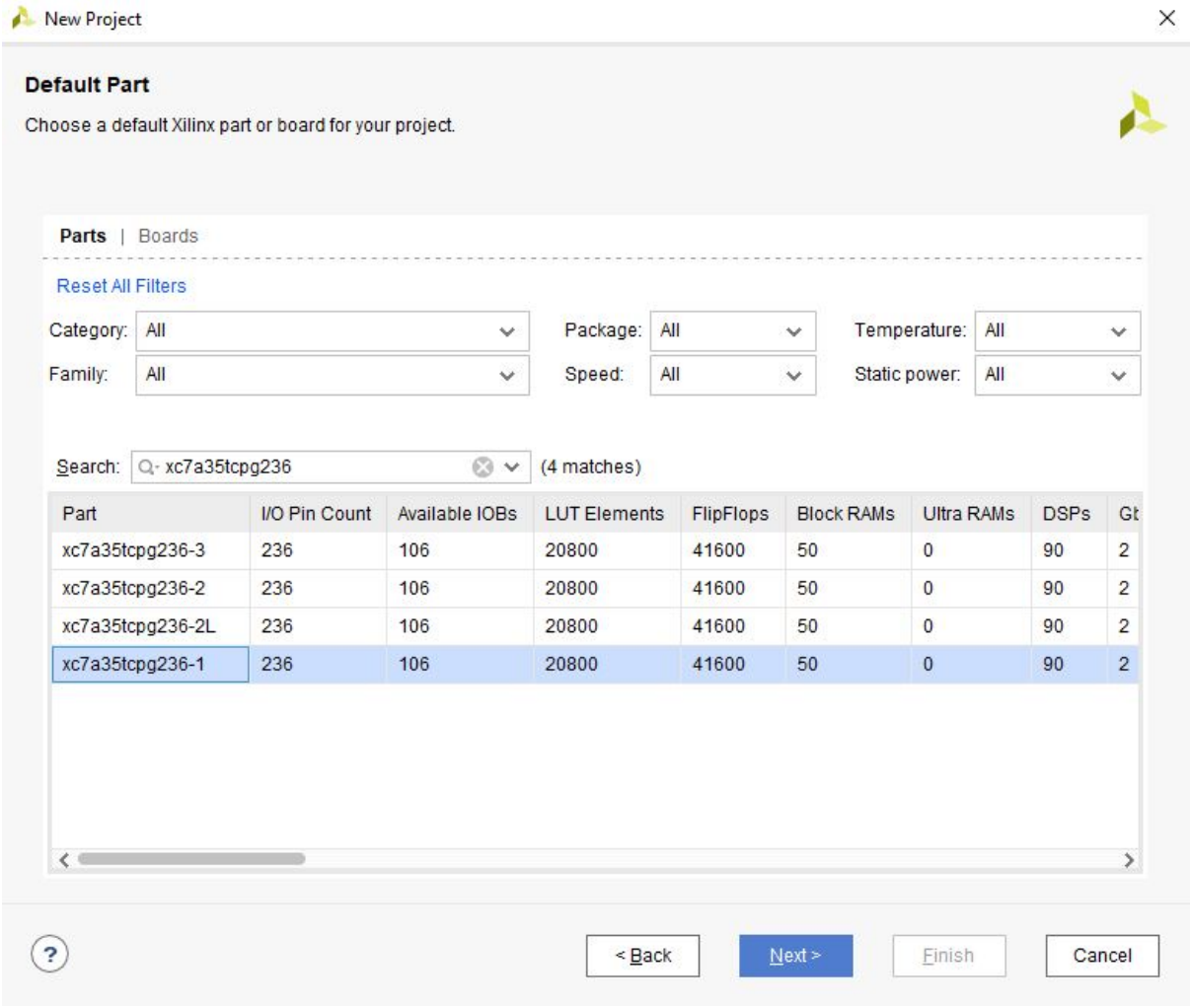


Figure 4: Selecting the target FPGA part

2.10 Click **Next** and then click **Finish** to create the Vivado project. Up to this stage, you should be seeing the overview of your project window, as shown in **Figure 5**.

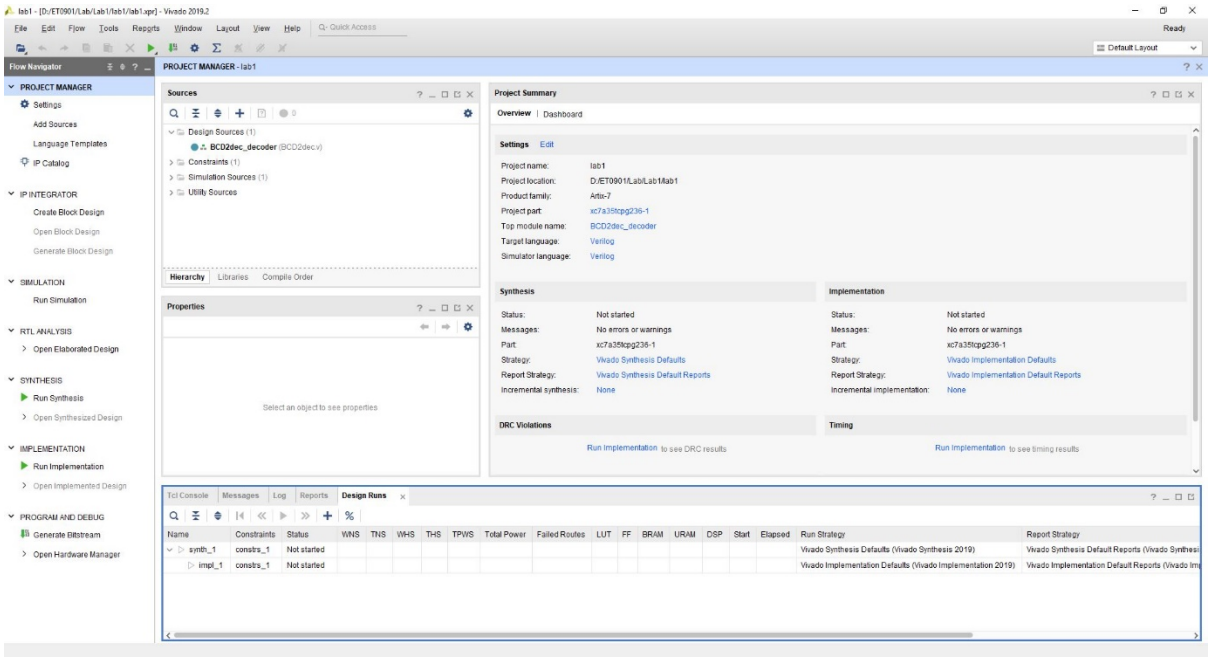


Figure 5: Overview of Project Window

2.11 In the *Sources* pane, double-click the **BCD2dec.v** entry to open the file in text mode, as shown in **Figure 6**. Analyze the Verilog source code and understand it. You may refer to relevant sections of Chapter 3 lecture notes if in doubt.

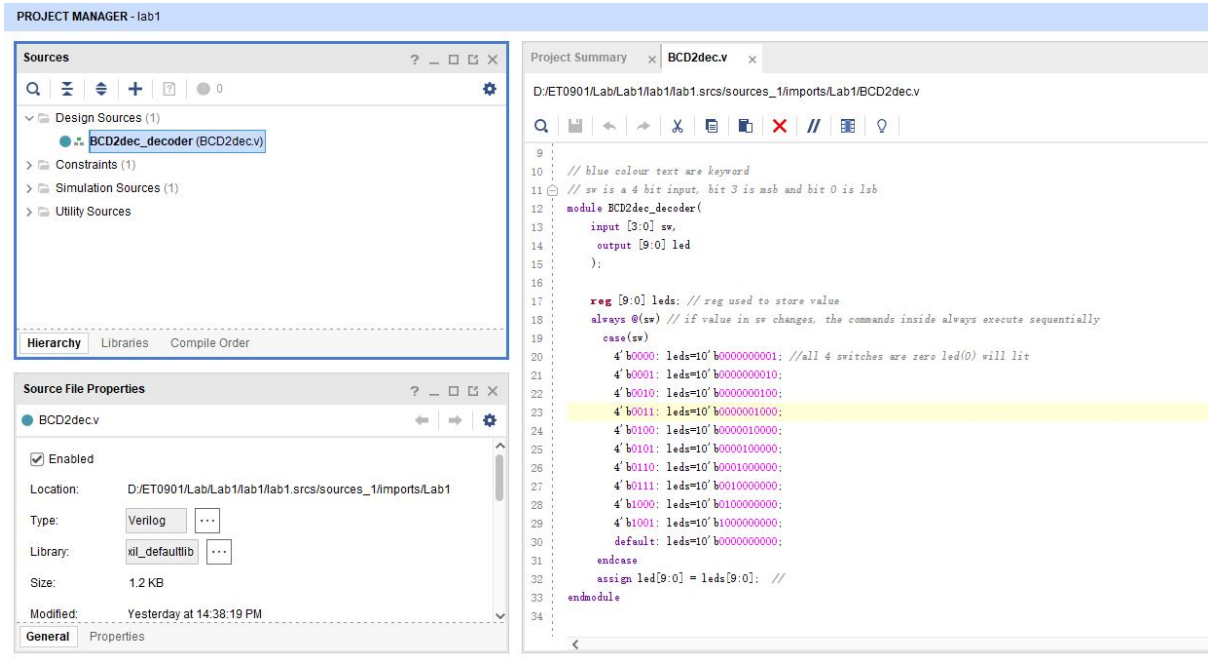


Figure 6: The Sources Pane

2.12 In the *Sources* pane, expand the *Constraints* folder and double-click the **BCD2dec.xdc** entry to open the file in text mode. As this file was originally copied from the master

constraint file, all the commands are commented out by the #. In this project, we need to assign the correct FPGA pins to Basys3 board's ten LEDs and four data switches.

Un-comment lines 12-19 and lines 47-66 using the // icon. The desired outcome is shown in Figure 7 and 8, respectively.

```
11 ## Switches
12 set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
13     set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
14 set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
15     set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
16 set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
17     set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
18 set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
19     set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
20 #set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
21     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
```

Figure 7: Un-comment sw[3] to sw[0] in BCD2dec.xdc

```
46 ## LEDs
47 set_property PACKAGE_PIN U16 [get_ports {led[0]}]
48     set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
49 set_property PACKAGE_PIN E19 [get_ports {led[1]}]
50     set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
51 set_property PACKAGE_PIN U19 [get_ports {led[2]}]
52     set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
53 set_property PACKAGE_PIN V19 [get_ports {led[3]}]
54     set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
55 set_property PACKAGE_PIN W18 [get_ports {led[4]}]
56     set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
57 set_property PACKAGE_PIN U15 [get_ports {led[5]}]
58     set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
59 set_property PACKAGE_PIN U14 [get_ports {led[6]}]
60     set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
61 set_property PACKAGE_PIN V14 [get_ports {led[7]}]
62     set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
63 set_property PACKAGE_PIN V13 [get_ports {led[8]}]
64     set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
65 set_property PACKAGE_PIN V3 [get_ports {led[9]}]
66     set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
67 #set_property PACKAGE_PIN W3 [get_ports {led[10]}]
68     #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
```

Figure 8: Uncomment led[9] to led[0] in BCD2dec.xdc

2.13 Click **Ctrl+S** to save the changes. You may also use the Save File Button .

2.14 Expand the *Open Elaborated Design* entry under the *RTL Analysis* tasks of the *Flow Navigator* pane and click on **Schematic** as shown in **Figure 9**. Click OK in the

Elaborated Design dialog box. The design will be elaborated, and a block diagram of the design is displayed to indicate the inputs and outputs as shown in **Figure 10**.

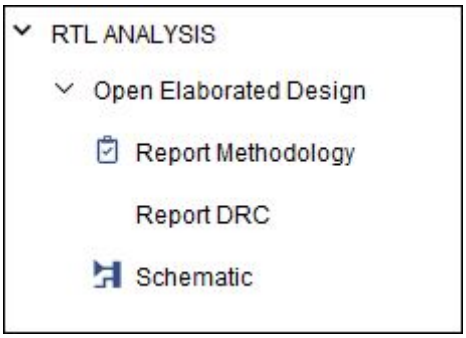


Figure 9: RTL Analysis

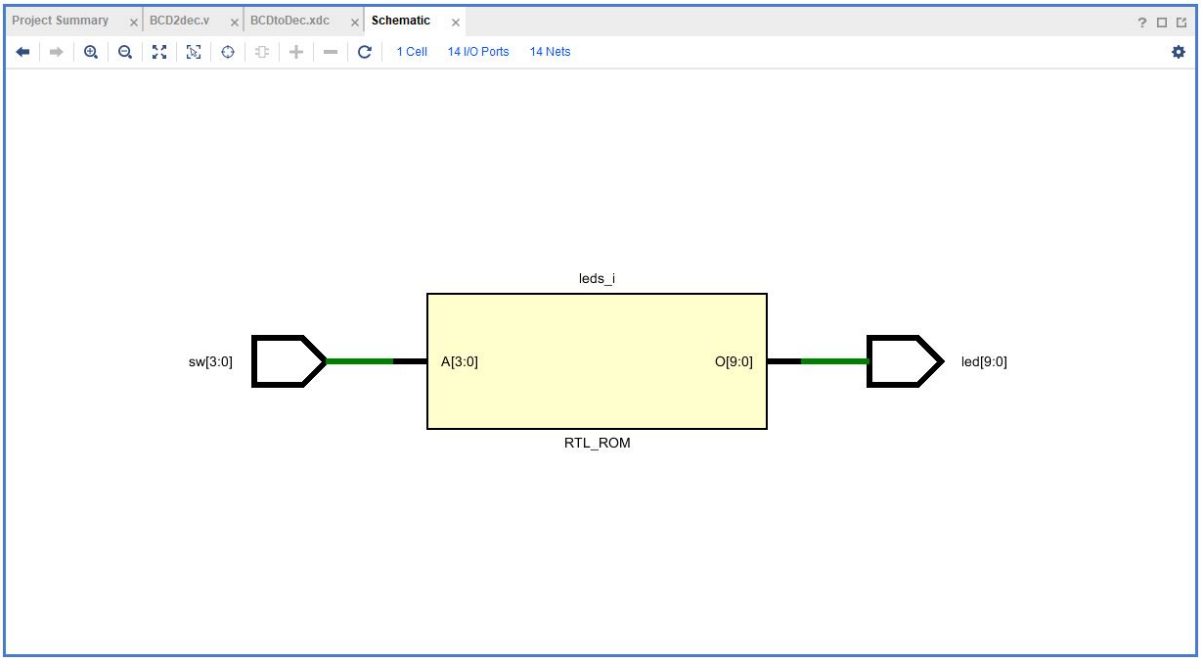


Figure 10: Schematic View of the Logic Design

3 Simulate the Design using the Vivado Simulator

3.1 Click **Add Sources** under the *Project Manager* tasks of the *Flow Navigator* pane as shown in **Figure 11**.

3.2 Select the *Add or Create Simulation Sources* option in the *Add Sources* pane and click **Next**.

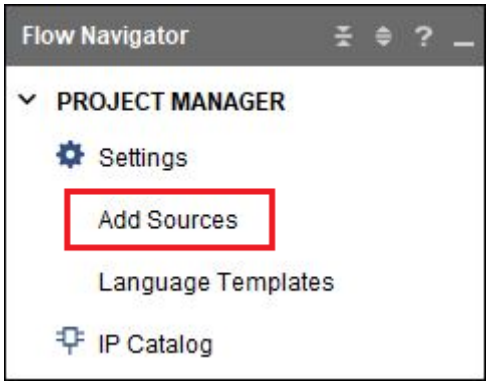


Figure 11: Add Sources option in Project Manager

3.3 Click the **Add Files** box and browse to the **D:/ET0901>Lab>Lab1** directory, select *BCD2dec_tb.v,* click **OK** and then click **Finish**.

3.4 Select the *Sources* tab and expand the *Simulation Sources* group as shown in **Figure 12**. The *BCD2dec_tb.v* file is added under the *Simulation Sources* group, and the source file *BCD2dec.v* is automatically placed in its hierarchy as a DUT (Device Under Test) instance.

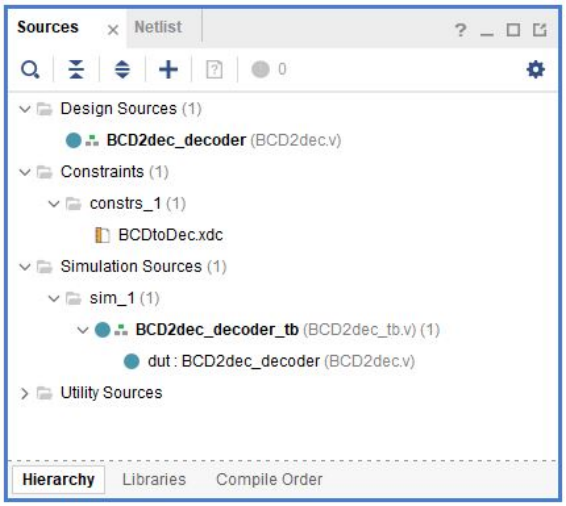


Figure 12: Simulation Sources

3.5 Double-click on the *BCD2dec_tb.v* in the *Sources* pane to view and analyze its contents. This Verilog code describes the waveforms of the input switches to test for the *BCD2dec.v* design.

3.6 Click on **Settings** under the *Project Manager* tasks of the *Flow Navigator* pane. A **Project Settings** form will appear. Select the **Simulation** properties form.

3.7 Select the **Simulation** tab, and set the **Simulation Run Time** value to 160 ns and click **OK** as shown in **Figure 13**.

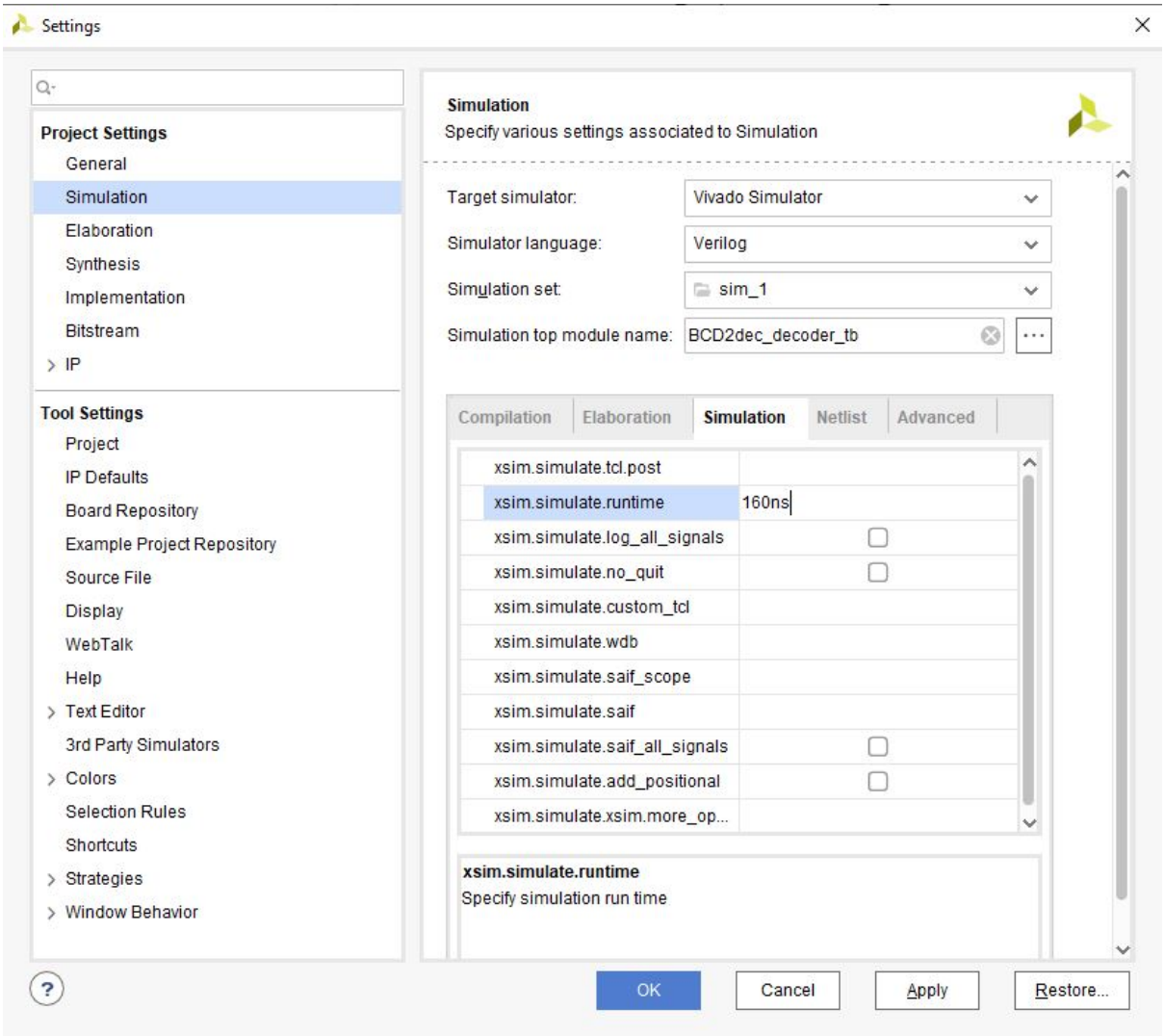


Figure 13: Setting simulation run time in simulation settings

3.8 Click on **Run Simulation > Run Behavioral Simulation** under the *Project Manager* tasks of the *Flow Navigator* pane. The test-bench and source files will be compiled and the Vivado simulator will run if no error occurs. You will see a simulator output. Click on the Zoom Fit ( ) button and you will see the output similar to **Figure 14**. You need to expand switches and leds to see the individual signals, as shown in **Figure 15**. Study the waveforms and ensure that the outputs correspond to those of the BCD to decimal decoder. You may right click on switches, ***select Radix>Binary*** to change the signal value displayed to binary. Click on a signal to view its logic value at a certain time point using the cursor. In short, familiarize yourself with the options in the simulator pane.

3.9 Close the simulator by selecting **File > Close Simulation**. Click **OK** and then click **Discard** to close it without saving the waveform.

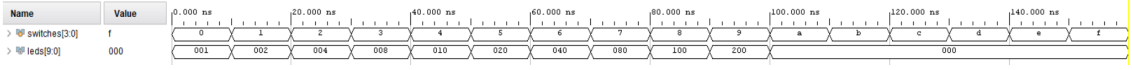


Figure 14: Simulator Output

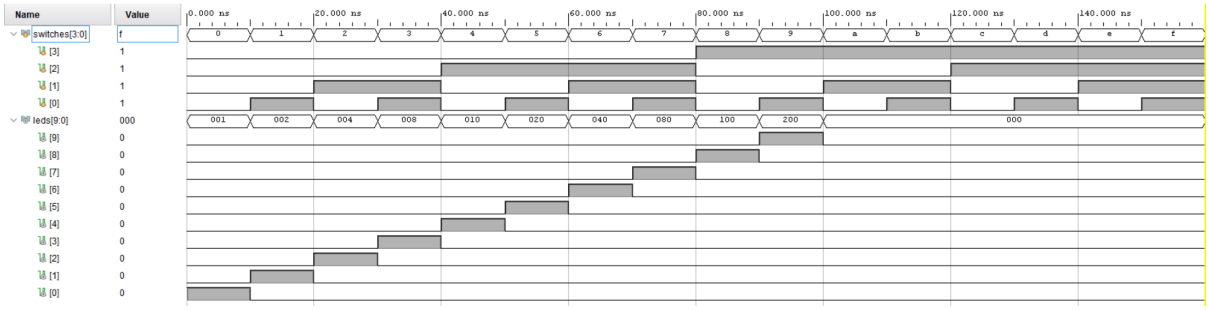


Figure 15: Expanded Signal View in the Simulation Pane

4 SYNTHESIZE design with Vivado synthesis tool and Analyze the Project Summary Output

4.1 Click on **Run Synthesis** under the *Synthesis* tasks of the *Flow Navigator* pane and then click OK in the *Launch Runs* dialog box. The synthesis process will be running on the *BCD2dec.v* file (and all its hierarchical files if they exist). It will take a few minutes. When the process is completed a *Synthesis Completed* dialog box with three options will be displayed.

4.2 Select the ***Open Synthesized Design*** option and click **OK** as we want to look at the synthesis result before progressing to the implementation stage. Click **Yes** to close the elaborated design if the dialog box pops up.

4.3 Select the **Project Summary** tab and understand the various windows. If you don't see the Project Summary tab, then select **Layout > Default Layout** or click the **Project Summary** icon Σ. Click on the various links to see what information they provide and which allows you to change the synthesis settings.

4.4 Click on the **Table** tab in the **Project Summary** tab as shown in **Figure 16**. Notice that there are an estimated 5 LUTs and 14 IOs (4 inputs and 10 outputs) that are used. LUTs are look-up-table which is covered in Chapter 2 of the lecture notes.

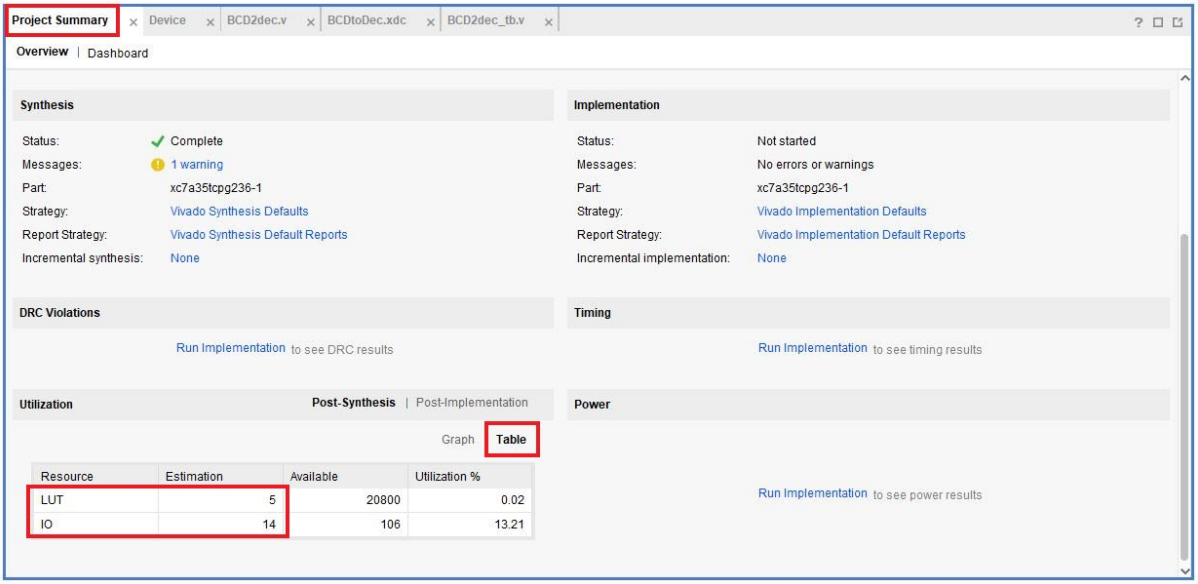


Figure 16: Synthesized Result – Project Summary on LUT and IO

4.5 In the *Flow Navigator*, under *Synthesis* (expand *Synthesized Design* if necessary), click on **Schematic** to view the synthesized design in a schematic view.

5 Implement the design with the Vivado Implementation Defaults settings.

5.1 Click on **Run Implementation** under the *Implementation* tasks of the *Flow Navigator* pane and then click OK in the *Launch Runs* dialog box. The implementation process will start based on the synthesized design. Once again, this process will take a few minutes. When the process is completed an *Implementation Completed* dialog box with three options will be displayed.

5.2 Select **Open implemented design** and click **OK** as we want to look at the implemented design in a Device view tab.

5.3 Click **Yes,** if prompted, to close the synthesized design. The implemented design will be opened. This is the design in the FPGA internal structure. You may zoom in to view the details, like shown in **Figure 17**.

5.4 Close the implemented design view by selecting **File > Close Implemented Design.**

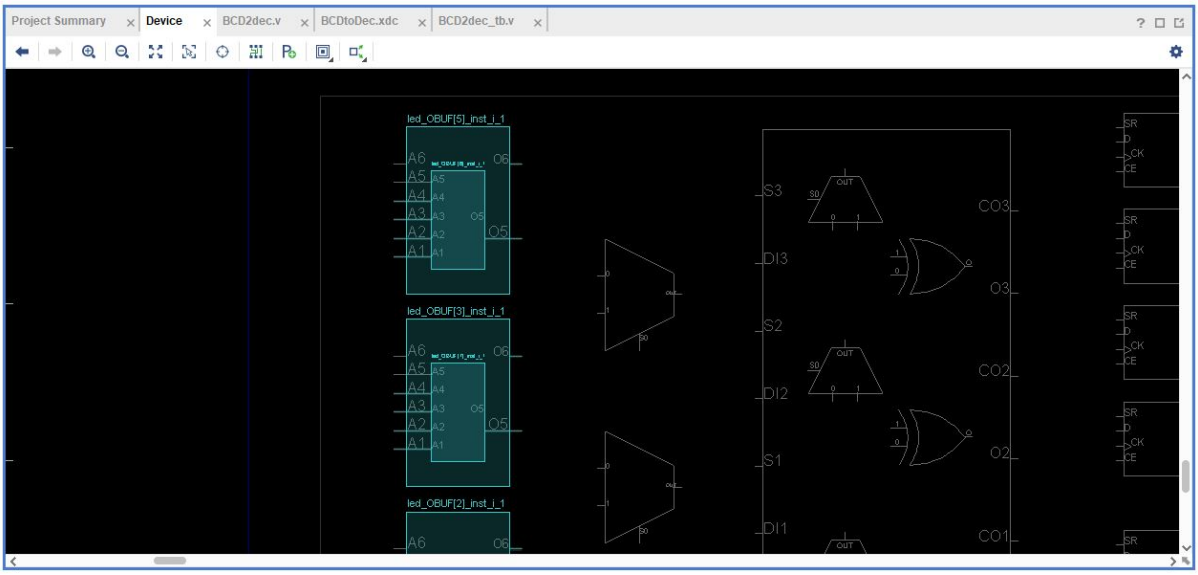


Figure 17: Zoomed Device View in Implemented Design

6 Generate the Bitstream and programming the FPGA to verify functionality of the design on the Basys3 board

6.1 Connect the Micro-USB socket of the cable to the JTAG PROG connector (refer to **Figure 1**) of the Basys3 board and the USB socket to your computer. Make sure that the JP1 jumper is connected as shown in **Figure 1**.

6.2 Make sure that the JP2 jumper is connected to use USB power as shown in Figure 1. Power **ON** the board.

6.3 Click on the **Generate Bitstream** entry under the *PROGRAM AND DEBUG* tasks at the bottom of the *Flow Navigator* pane and then click OK in the *Launch Runs* dialog box. The bitstream generation process will begin on the implemented design. When the process is completed a *Bitstream Generation Completed* dialog box with three options will be displayed.

6.4 Select the *Open Hardware Manager* option and click OK.

6.5 Expand the *Open Hardware Manager* option and click *Open Target* option. Click on the Select *Auto Connect* option as shown in **Figure 18**.

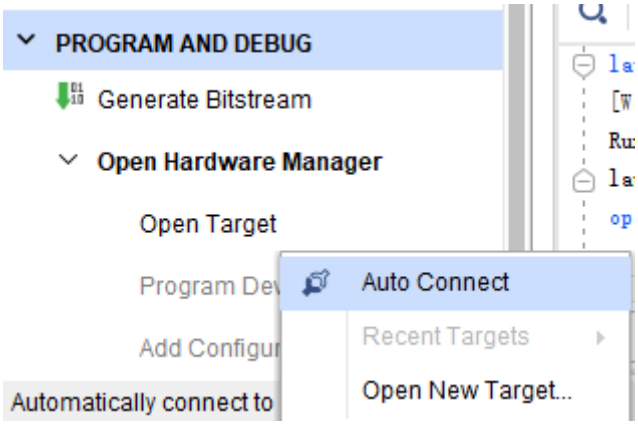


Figure 18: Hardware Manager, Auto Connect to Target Device

6.6 The Hardware Manager Window will open automatically. You can find your device listed in the *Hardware Window* as shown in **Figure 19**.

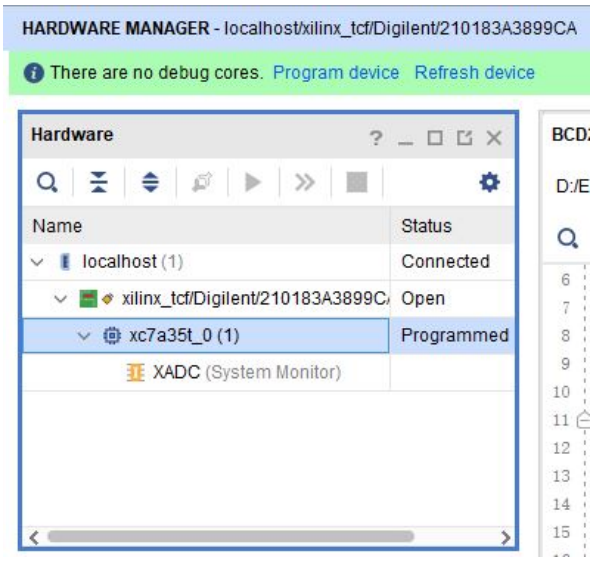


Figure 19: Hardware Manager Pane with Connected Target Device

6.7 Click on the *Program device* link in the green information bar to program the target FPGA device. The *Program Device* dialog box as shown in **Figure 20** will be displayed.

6.8 Click **Program** to program the FPGA. The DONE light (refer to Figure 1) will light when the device is programmed. You may see some other LEDs lit depending on switch positions.

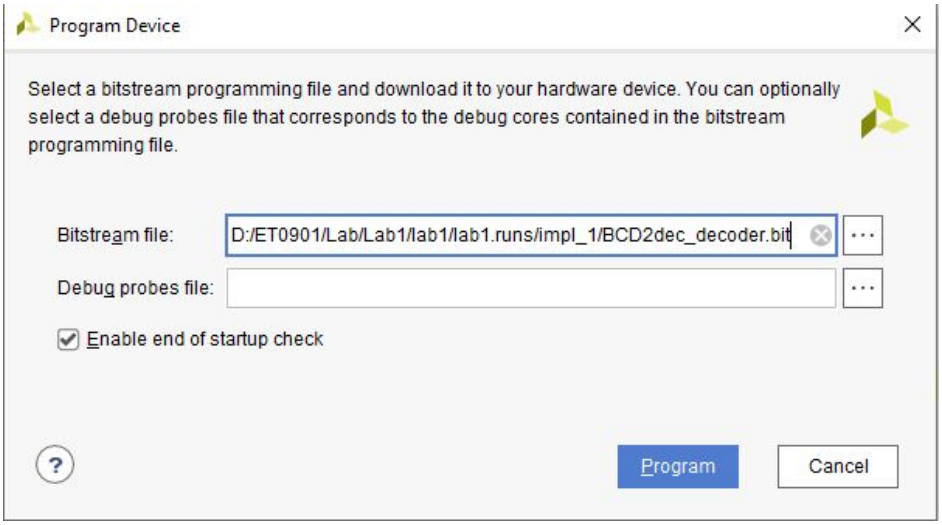
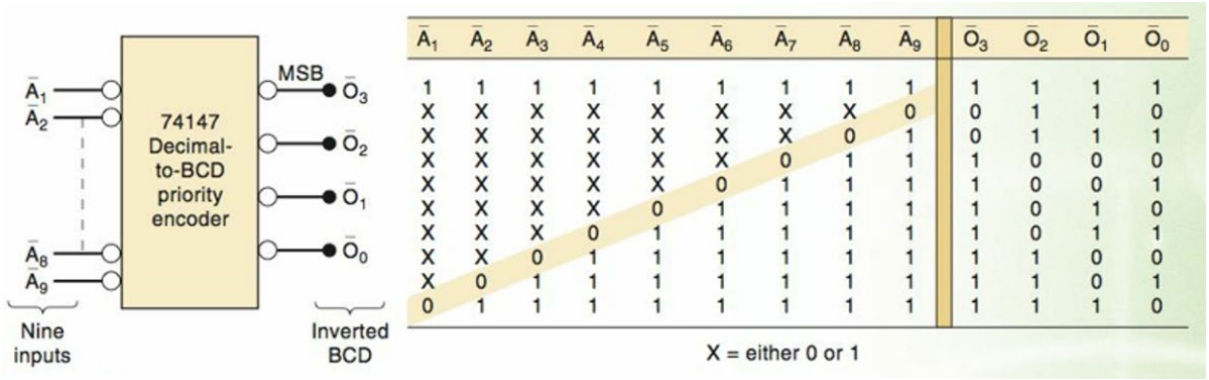


Figure 20: Program Device dialog box

6.9 Verify the functionality by flipping switches 3 to 0 accordingly and observing the output on the LEDs. (If you are not sure you may refer to the video in **Blackboard>Learning Resources>Lab**)

7 Extra Practice: Design a 74LS147 Decimal-to-BCD Priority Encoder

7.1 Recall your knowledge in Digital Electronics 2, a 74LS147 IC has 9 active-LOW inputs representing the decimal digits 1 through 9, and it produces the *inverted* BCD code corresponding to the highest numbered activated input. **Figure 21** shows the logic symbol and the truth table for the 74LS147 IC.



$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	$\bar{A}_8$	$\bar{A}_9$	$\bar{O}_3$	$\bar{O}_2$	$\bar{O}_1$	$\bar{O}_0$
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

X = either 0 or 1

Figure 21: 74LS147 Decimal-to-BCD Priority Encoder

7.2 Use 9 switch inputs and 4 LED outputs, design and implement a 74LS147 decimal-to-BCD priority encoder using Verilog hardware description language. You may choose to modify *BCD2dec.v* file directly, or create another source file and add into the project (in this case you will need to remove the *BCD2dec.v* file from the project). Take note that all inputs and outputs are active-LOW.

7.3 Modify the constraint file *BCD2dec.xdc*, such that the input and output ports are properly defined according to your source code.

7.4 Synthesize, implement, and generate bitstream for your design. Program the FPGA and verify that the function is compatible to the truth table. If not successful, further modify the source code and/or the constraint file.

7.5 When satisfied, power **OFF** the board. Close the hardware session by selecting **File > Close Hardware Manager.** Click **OK** to close the session.

7.6 Close the **Vivado** program by selecting **File > Exit** and click **OK**.

---------- ***END OF THIS LAB*** ----------