LAB2: *Using Basys3 Seven Segment Display*

OBJECTIVES:

Implement a circuit that drives the time-multiplexed quad seven-segment display on the Basys3 board.

EQUIPMENTS:

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

INTRODUCTION:

The Basys3 board contains one four-digit *common anode* time-multiplexed seven-segment LED display (SSD). Each digit shares eight common control signals to light individual segments. Each individual digit has a separate anode control input as shown in **Figure 1**. More details can be found in Appendix 1. All these control signals are *Active Low*. In order to enable any given digit, drive its anode control signal low. Enabled digits will display the segments selected by the eight *active low* segment controls (CA, CB…CG, DP).

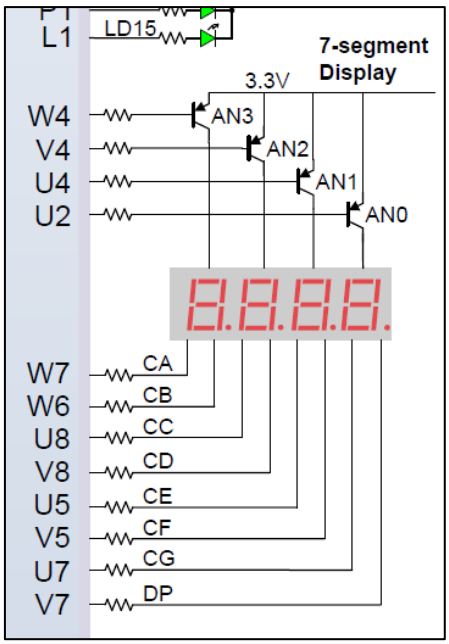


Figure 1: Electronic Connection of the Quad SSD to the Artix-7 FPGA

Since all four digits share the same segment controls, to get a unique display on each of the digits, you must apply a technique called time-multiplexing as shown in **Figure 2**. For the human eyes to see the display correctly, the refresh period should typically fall between 1 ms to 16 ms.

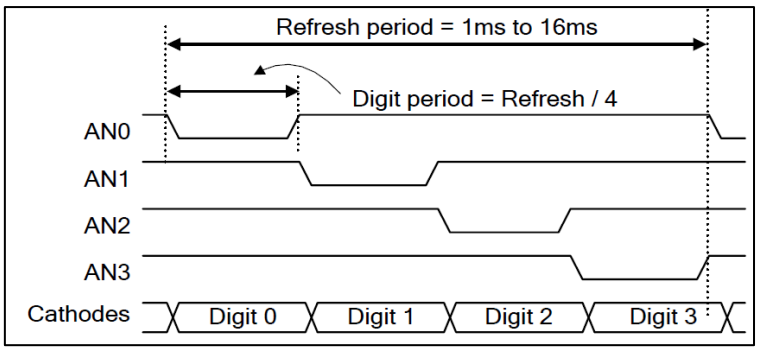


Figure 2: Time-multiplexed Timing Diagram for the four SSD Devices

In this laboratory session, we will be designing a time multiplexed SSD driver by programming the FPGA chip on the Basys3 board as shown in **Figure 3**. The 16 data switches are BCD inputs for the four SSD. The four data switches on right will control the rightmost SSD and display '0' to '9' accordingly. If the BCD input is "0000" (switch position is down for logic '0'), it will display '0'. If the BCD input is "1001" (switch position is up for logic '1'), it will display '9' etc. However, at other combinations the SSD should not be displaying anything. When the reset (centre) button is pressed and held, it will display the rightmost SSD and blank out the rest of the other three SSDs.

Figure 4 shows the symbol for the design and the adjacent table shows the connections of the signals to the Basys3 board. Note that the clk signal is taken from the on-board 100 MHz oscillator, hence we need to divide the clk signal to obtain the correct time-multiplexed anode control signals shown in **Figure 2**.

PROCEDURE:

1 Copy relevent files from Blackboard

1.1 Create a new folder **D:/ET0901>Lab>Lab2** 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.

- ssd.v

- ssd_tb.v

- ssd.xdc

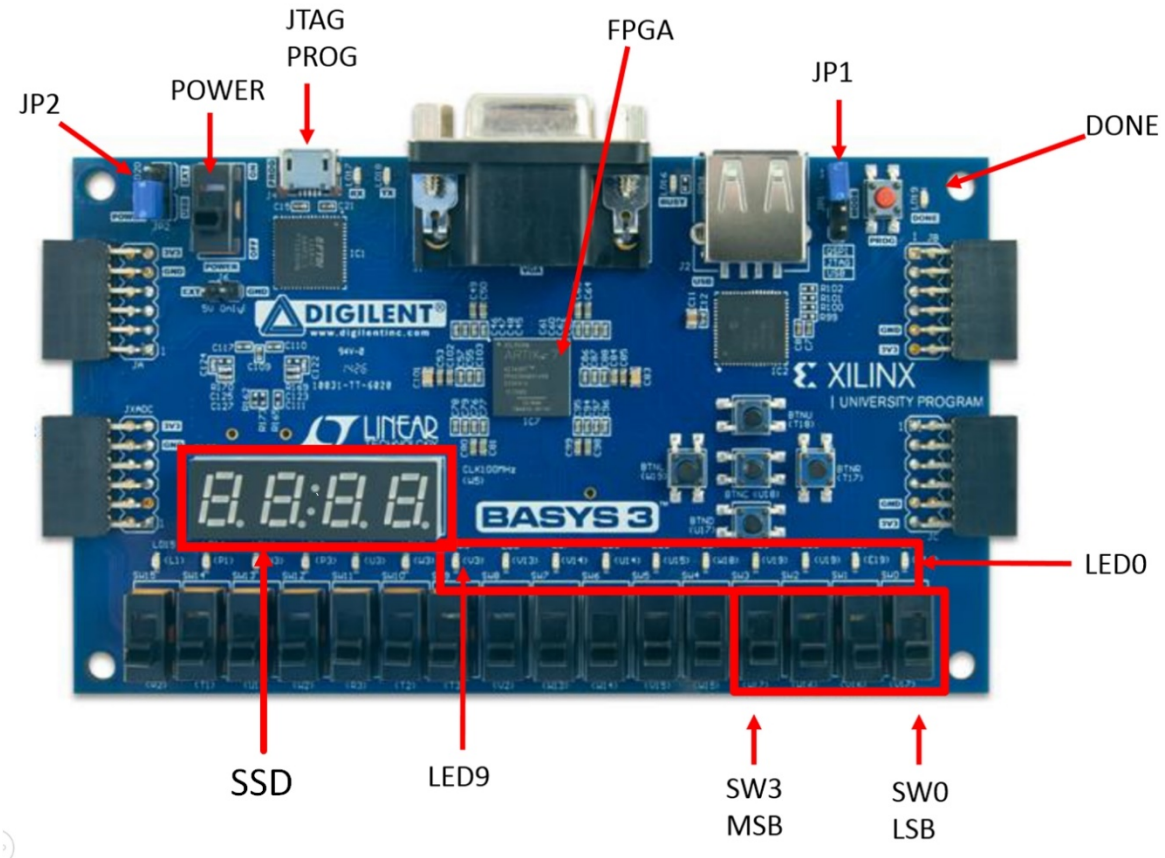
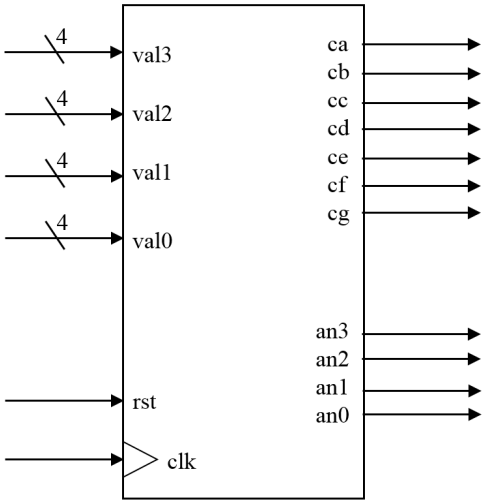


Figure 3: Basys3 Artix-7 FPGA Trainer Board Layout



Design Signal	Basys 3 connection	Remarks
val3[3:0]	sw15 to sw12	BCD value of left most SSD
val2[3:0]	sw11 to s 8	BCD value of left center SSD
val1[3:0]	sw7 to sw4	BCD value of right center SSD
val0[3:0]	sw3 to sw0	BCD value of right most SSD
rst	btn c	reset to display right most SSD only
clk	100 MHz oscillator	
ca to cg	refer to figure 1	control for the cathodes of SSD
an3	refer to figure 1	anode control for left most SSD
an2	refer to figure 1	anode control for left center SSD
an1	refer to figure 1	anode control for right center SSD
an0	refer to figure 1	anode control for right most SSD

Figure 4: Symbol of the time multiplexed quad SSD driver design

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.

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>Lab2** and click Select.

2.4 Enter **lab2** 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 5**.

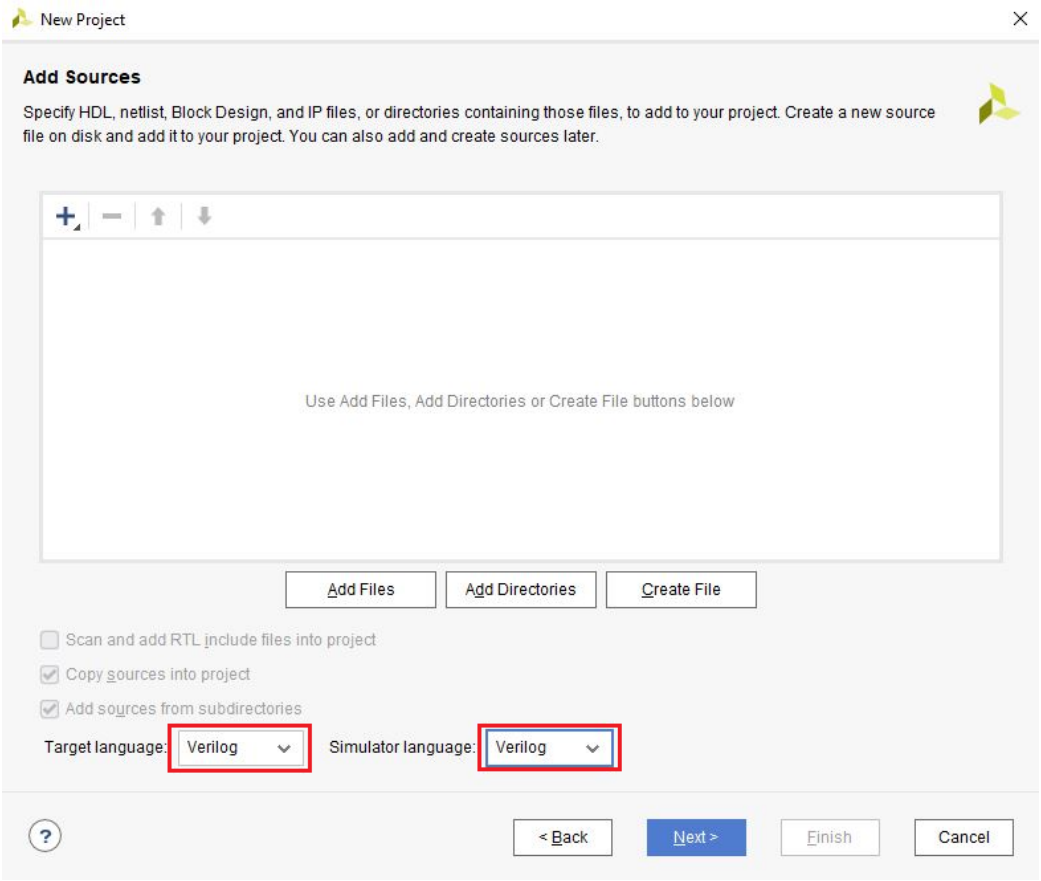


Figure 5: Selecting Language in Add Source form

2.7 Click the **Add Files** box and browse to the **D:/ET0901>Lab>Lab2** directory, select *ssd.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>Lab2** directory, select *ssd.xdc* and click **OK** (if necessary), and then click **Next.**

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

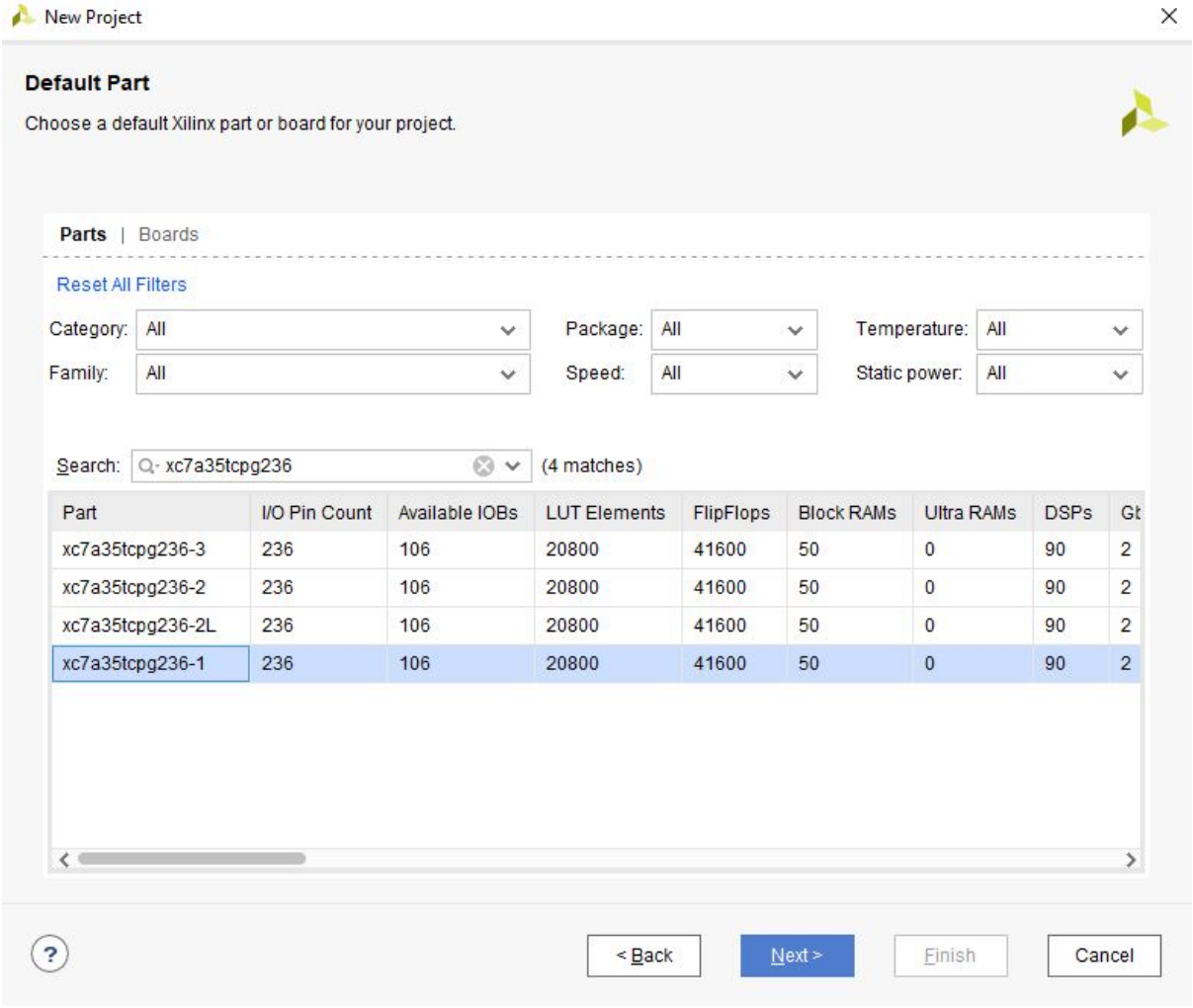


Figure 6: Selecting the target FPGA part

2.10 Click **Next** and then click **Finish** to create the Vivado project.

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

2.12 In the *Sources* pane, expand the *Constraints* folder and double-click the **ssd.xdc** entry to open the file in text mode. Analyze the constraint file and understand it with reference to **Figure 4**.

2.13 Expand the *Open Elaborated Design* entry under the *RTL Analysis* tasks of the *Flow Navigator* pane and click on **Schematic** as shown in **Figure 7**. Click OK in the *Elaborated Design* dialog box. The design will be elaborated and a block diagram of the design is displayed.

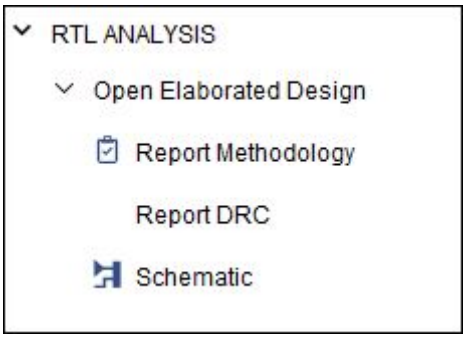


Figure 7: RTL Analysis

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 8**.

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

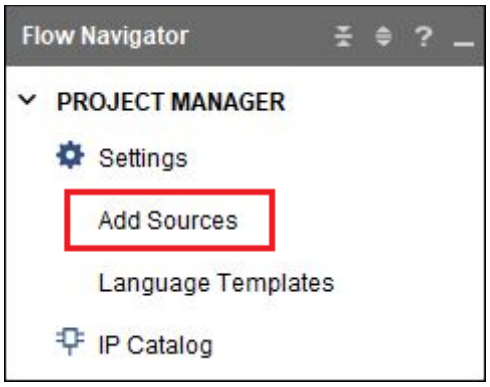


Figure 8: Add Sources option in Project Manager

3.3 Click the **Add Files** box and browse to the **D:/ET0901>Lab>Lab2** directory, select *ssd_tb.v,* click **OK** and then click **Finish**.

3.4 Select the *Sources* tab and expand the *Simulation Sources* group as shown in **Figure 9**. The *ssd_tb.v* file is added under the *Simulation Sources* group.

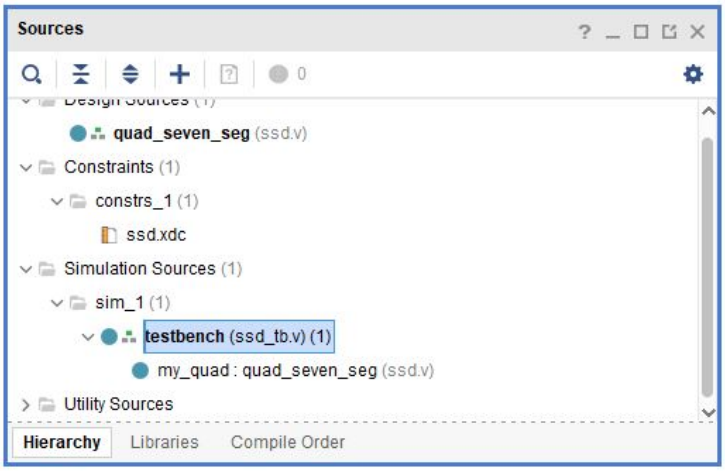


Figure 9: Simulation Sources

3.5 Double-click on the *ssd_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 *ssd.v* design (DUT).

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 50,000,000 ns (or 50 ms) and click **OK** as shown in **Figure 10**.

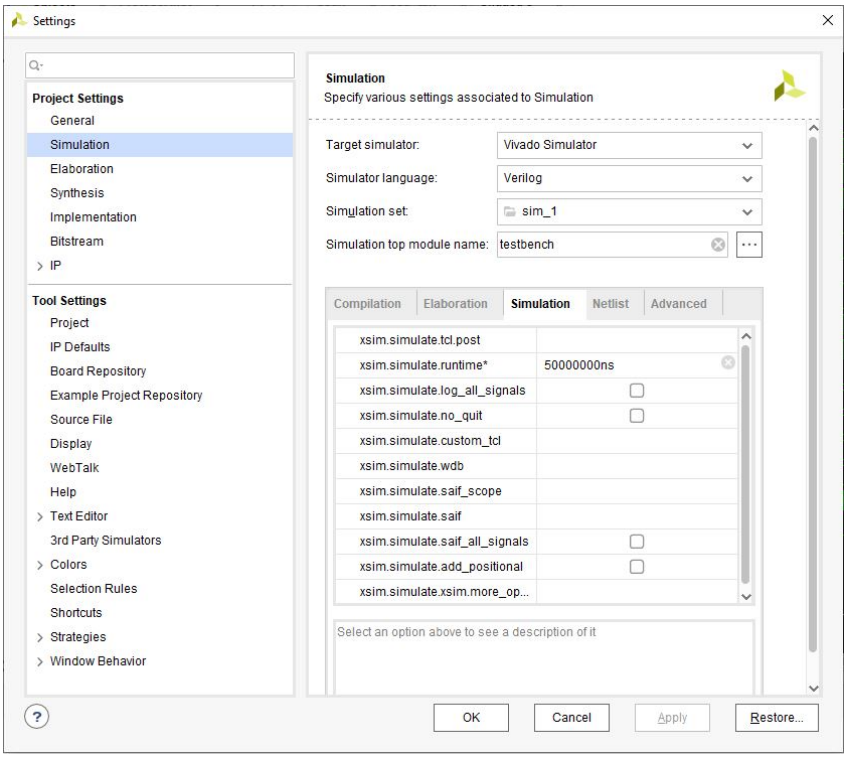


Figure 10: Setting simulation run time in simulation settings

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 11**. The signals can be repositioned by clicking and dragging. Study the waveforms and ensure that the outputs are correct for the four SSDs at the correct time interval.

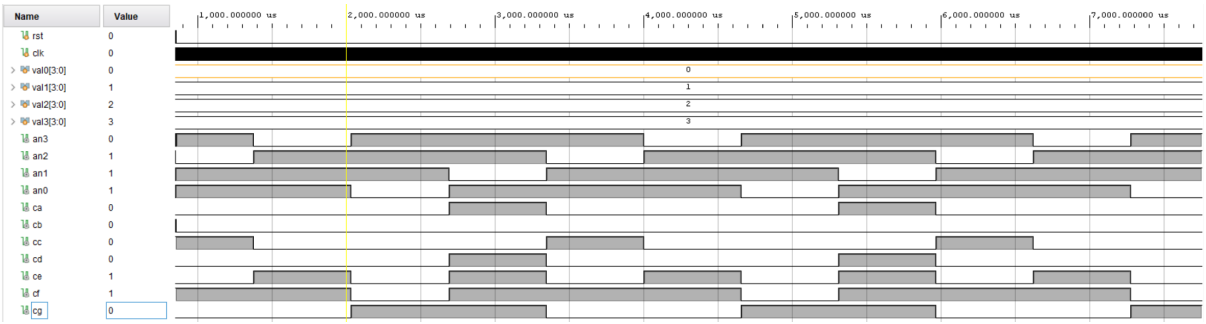


Figure 11: Simulator Output

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

4 Synthesize the design with the 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 *ssd.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 output 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. Answer the following questions.

# of Look-Up Table (LUT) = _____; Utilization % of LUT = _____

# of Flip-Flops (FF) = _____ ; Utilization % of FF = _____

# of Input/Output (I/O) = _____; Utilization % of IO = _____

# of Global Buffers (BUFG) = _____; Utilization % of BUFG = _____

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 in detail.

5.4 Close the implemented design view by selecting **File > Close 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 3**) of the Basys3 board and the USB socket to your computer. Make sure that the JP1 jumper is connected as shown in **Figure 3**.

6.2 Make sure that the JP2 jumper is connected to use USB power as shown in **Figure 3**. 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 12**.

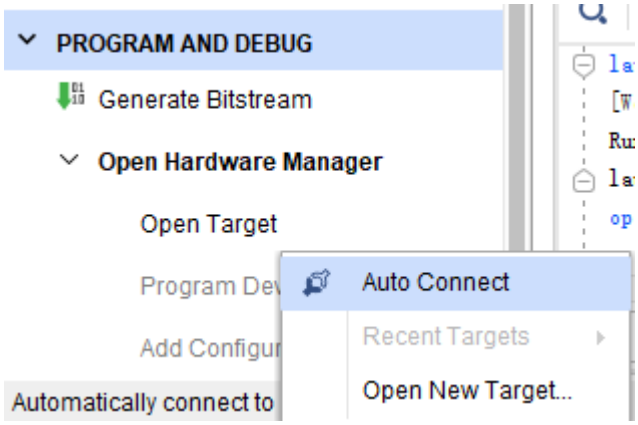


Figure 12: Hardware Manager, Auto Connect to Target Device

6.6 The Hardware Manager Window will open automatically. You can find your device listed in the *Hardware Window* as shown in **Figure 13**.

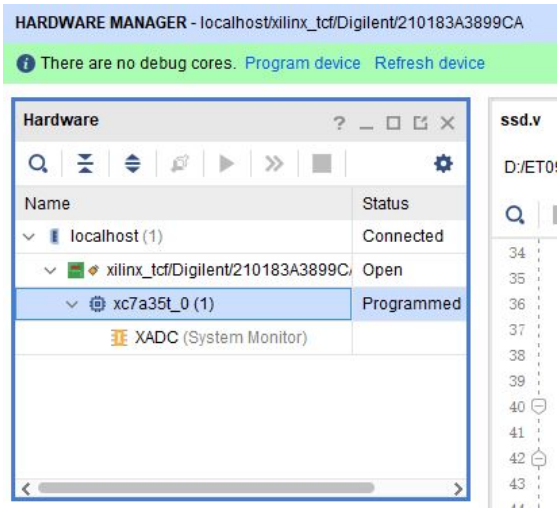


Figure 13: Hardware Manager Pane with Connected Target Device

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 14** will be displayed.

6.8 Click **Program** to program the FPGA. The DONE light (refer to **Figure 3**) will light when the device is programmed. You may see some other LEDs lit depending on switch positions.

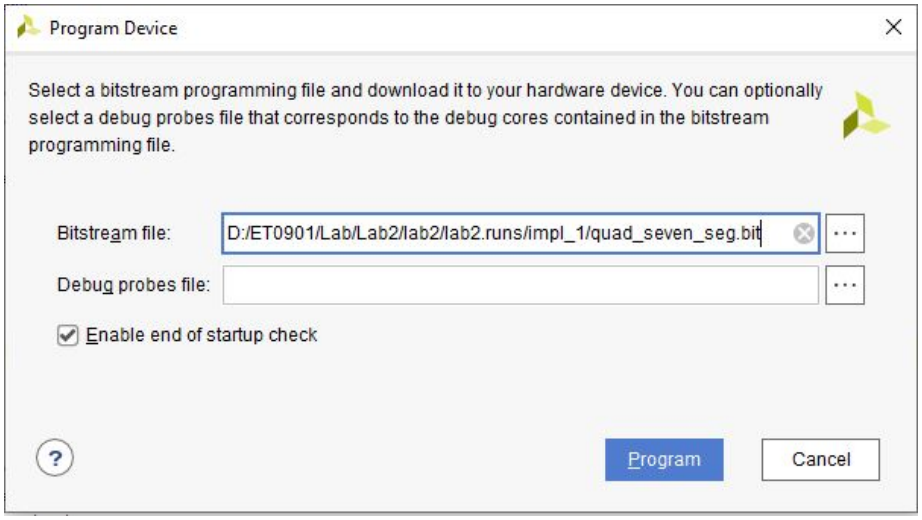


Figure 14: Program Device dialog box

6.9 Verify the functionality by flipping switches 3 to 0 accordingly and observing the output on the right most SSD then follow by the other switches and SSDs. Press the Centre Push Button, BTNC, to verify that only the rightmost SSD shows the corresponding BCD number. (If you are not sure you may refer to the video in **Blackboard>Learning Resources>Lab**)

7 Extra Practice: Reverse the Display

7.1 Modify the source code such that the SSD display is reversed. That is, the left-most four switches will control the right-most SSD, the 2nd left-most four switches will control the 2nd right-most SSD, etc.

7.2 Synthesize, implement, and generate bitstream for your design. Program the FPGA and verify that the function is correct. If not successful, further modify the source code.

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

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

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