**ECE428/528 Programmable ASIC Design**

# Tutorial of FPGA Design using Xilinx Vivado tool and Basys3 Development Board

## Introduction:

This tutorial demonstrates how to implement an FSM (Finite State Machine) controlled counter circuit on Basys 3 board in Vivado design flow. The FSM detects if a push down button is pushed and released. Once it detects such an event, it will enable the counter for one clock cycle and hence the counter is counting up by 1. The counter output is displayed by a 7-segment LED. The input signal values of the 7-segment LED for each decimal digit are listed below.

| Counter value (binary) | LED inputs (binary) | Counter value (decimal) |
|---|---|---|
| 0000 | 1000000 | 0 |
| 0001 | 1111001 | 1 |
| 0010 | 0100100 | 2 |
| 0011 | 0110000 | 3 |
| 0100 | 0011001 | 4 |
| 0101 | 0010010 | 5 |
| 0110 | 0000010 | 6 |
| 0111 | 1111000 | 7 |
| 1000 | 0000000 | 8 |
| 1001 | 0010000 | 9 |
| none of above | 1111111 | Turn off the LED display |

## Create a new project:

First, find Vivado on your desktop and start it. Click on “Create New Project” and then click “Next” on the pop-up window as shown in Fig. 1.

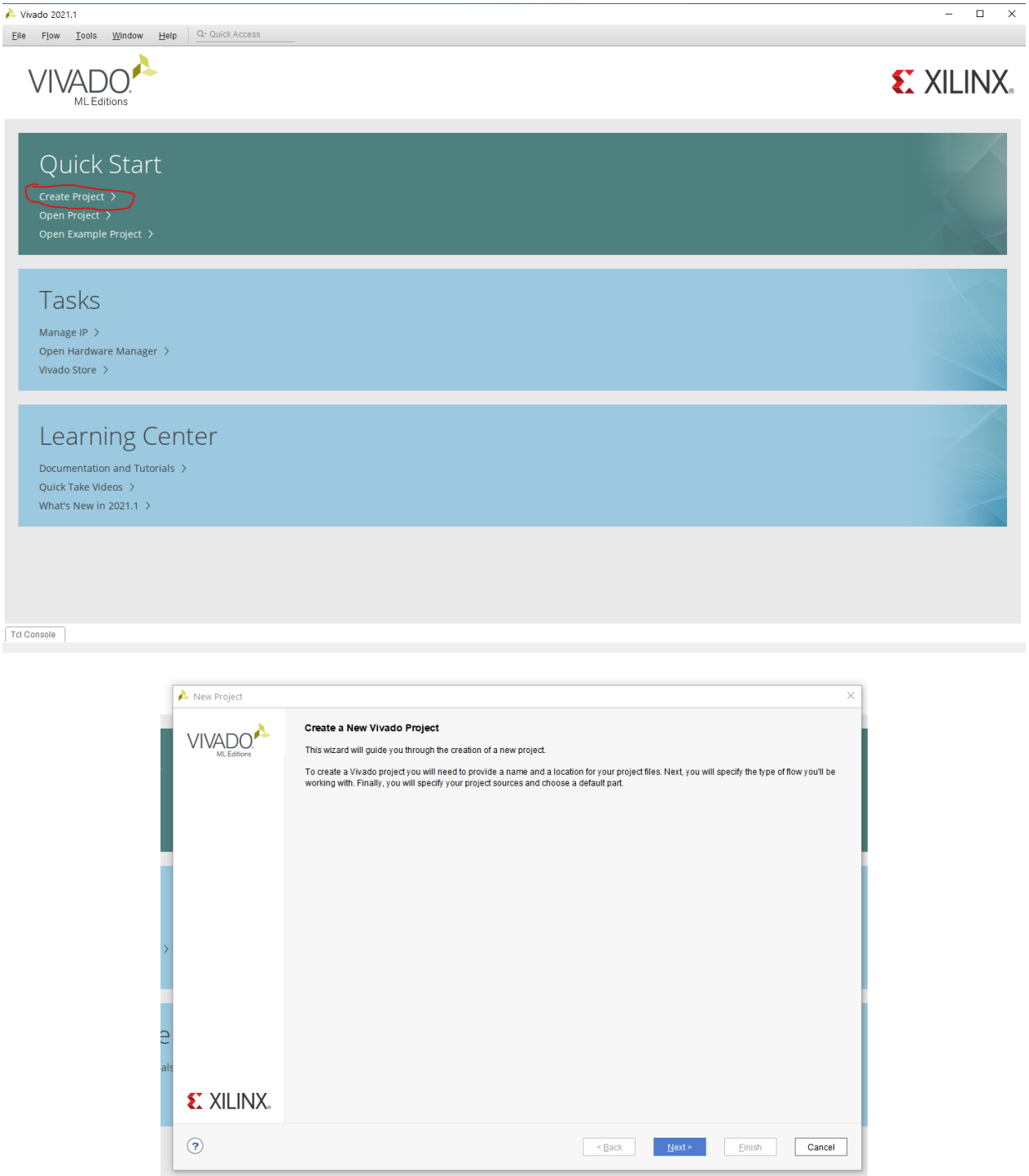



Fig. 1 Create a new project.

Enter your project name under your personal folder on the PC. All your data might be erased once you log off the computer. It is recommended to back up your files to a USB drive before you log off.

Then click “Next”.

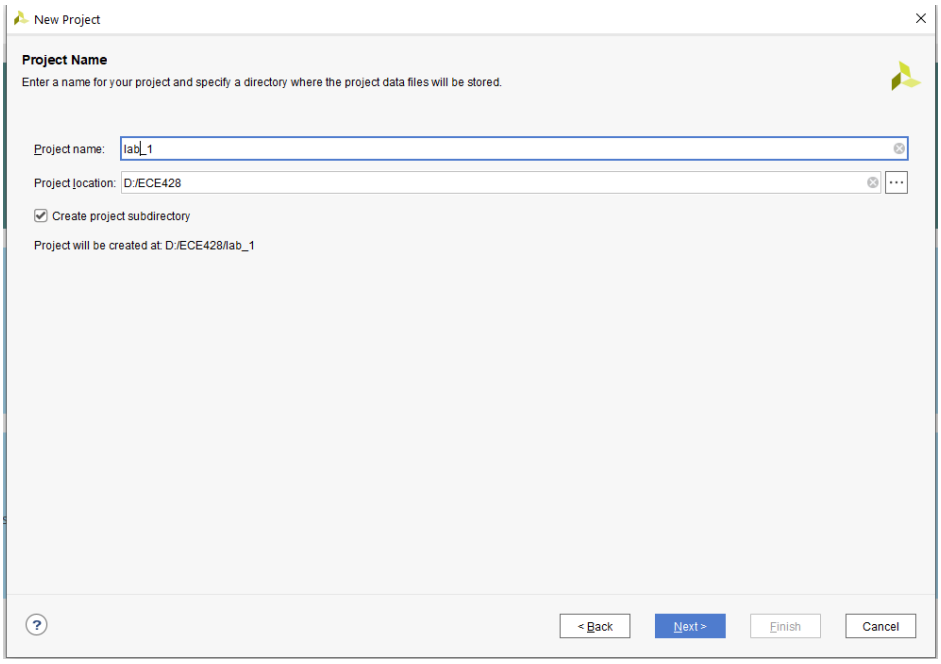



Fig. 2 Enter project name.

Select “RTL Project” as shown in Fig. 3, then click “Next”.

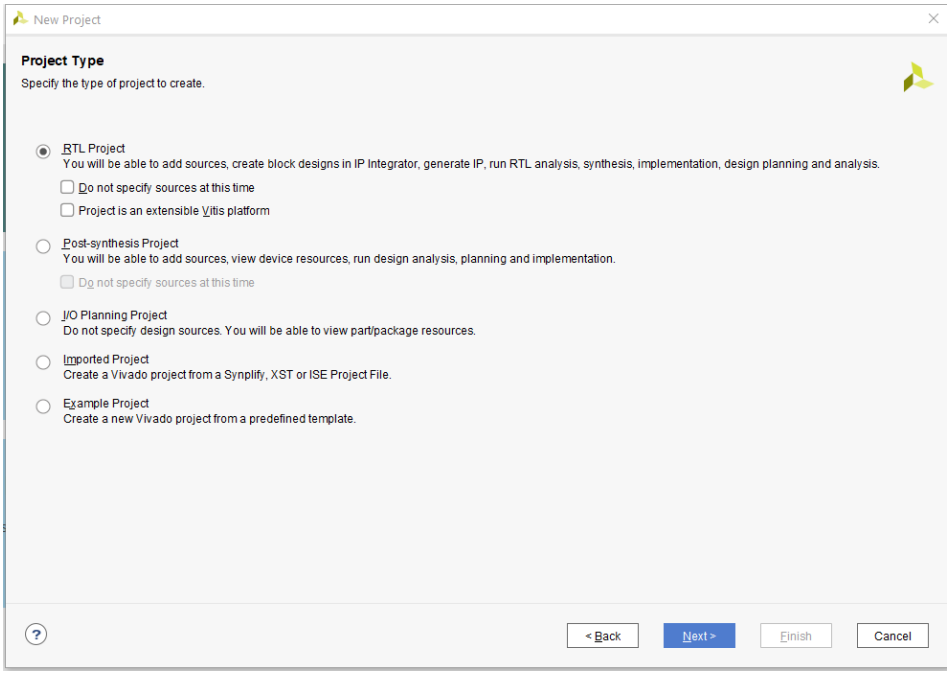



Fig. 3 Select project property.

You can add or create source files by clicking proper buttons. At this time, you will add source files later. So, you may click “Next” button.

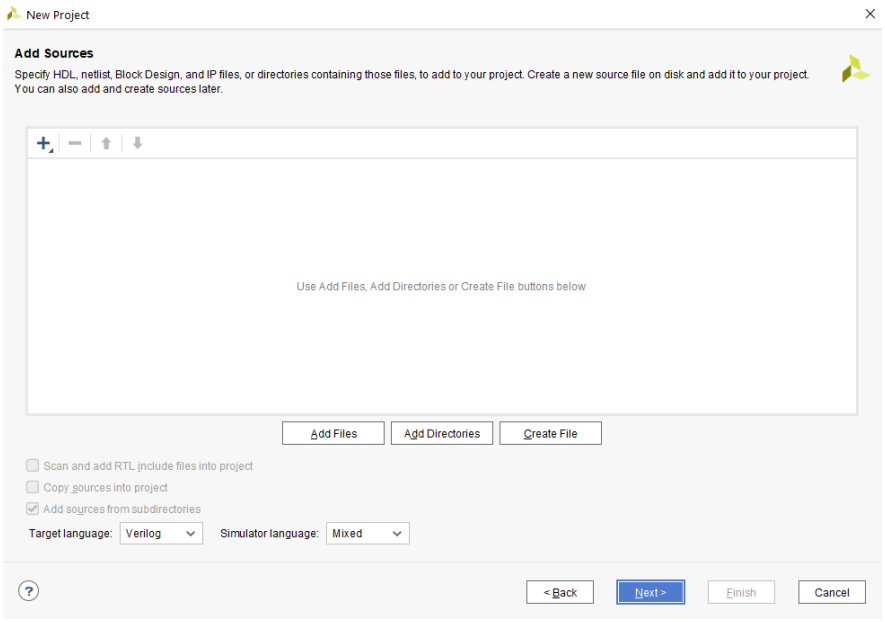



Fig. 4 Add and create source files.

After clicking Next, Add Constraints window appears. You can download the constraint file user.xdc from D2L and then add it to your project.

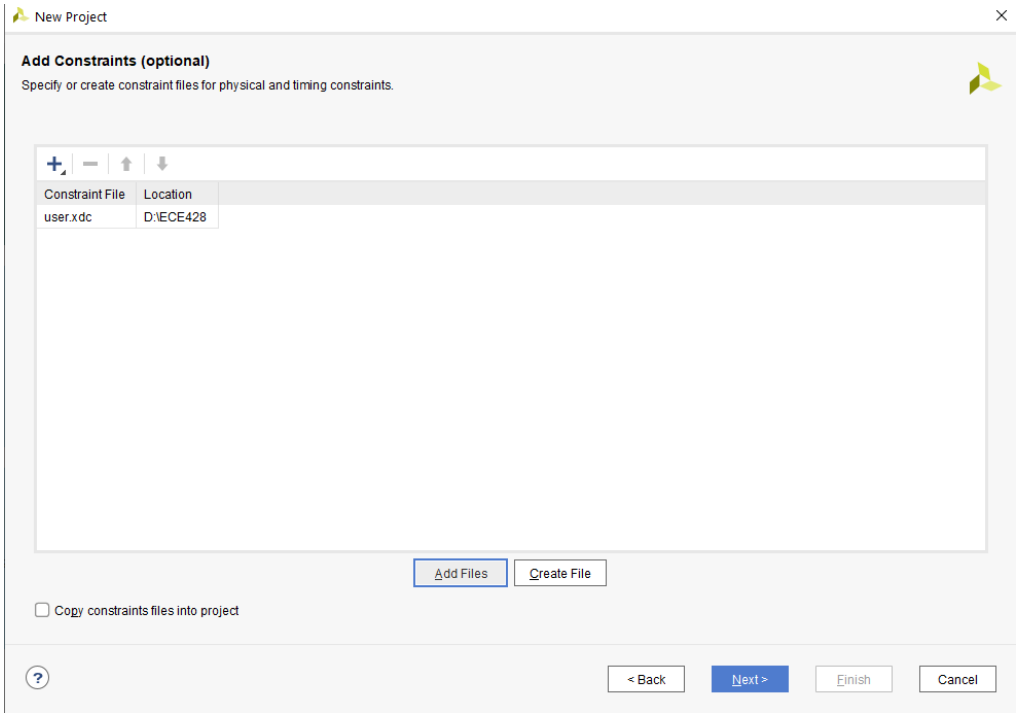



Fig. 5 Add user constraint file.

Select the correct Xilinx FPGA (xc7a35tcpg236-1). The information about Basys 3 can be found online. Or you can simply configure according to Fig. 6.

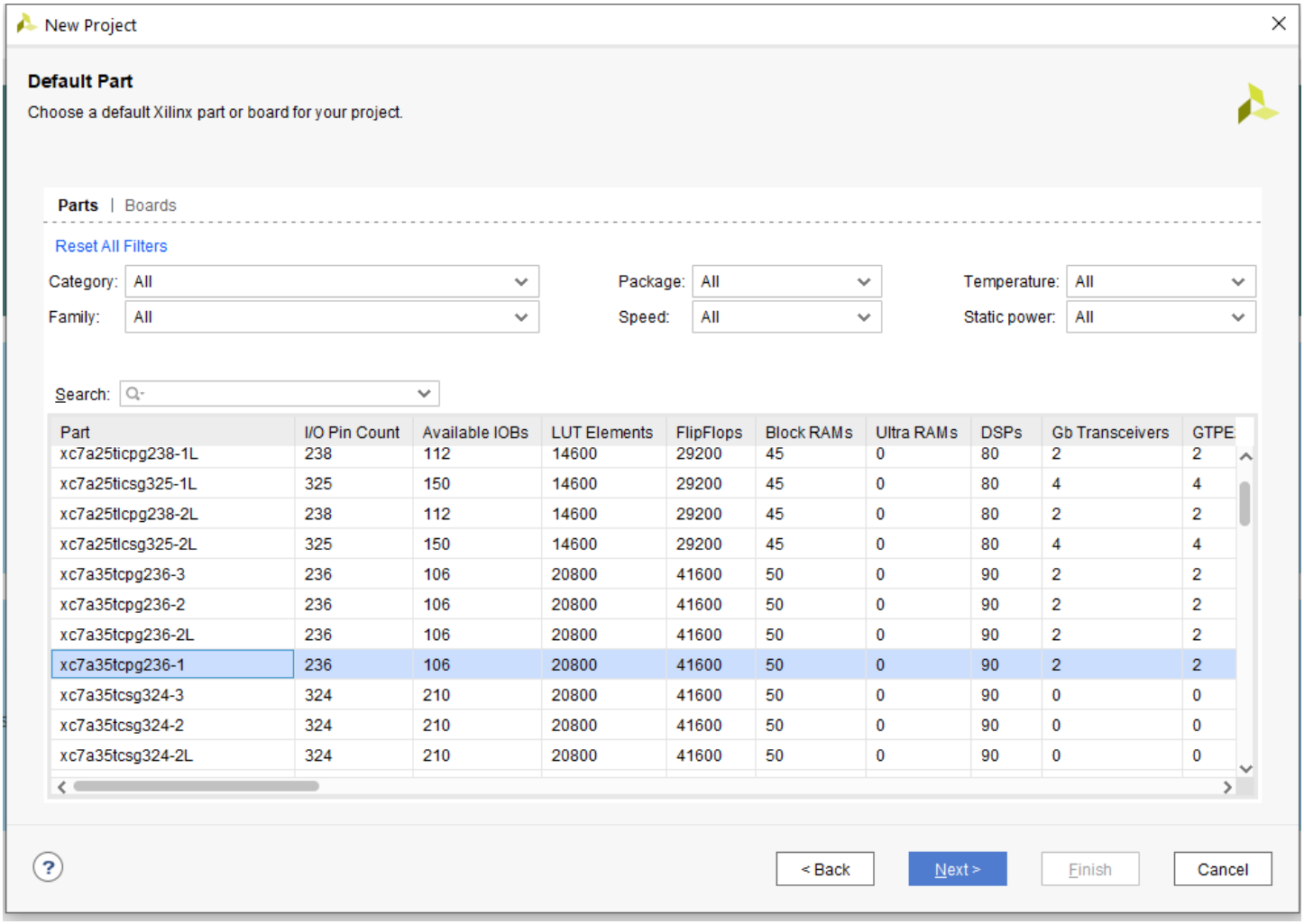



Fig. 6 FPGA part selection.

Then click “Next” until “Finish”. After this step, you successfully created a new project. A project window will appear as shown in the next page.

## Add and create source files:

In the Flow Navigator panel, go to “Project Manager” and select “Add Sources”. A pop-up window is shown as Fig. 7. Select “Add or create design sources” (which can be HDL codes, schematic, etc.).

(We already added constraints when creating the project and will add simulation sources later)

Select the correct item and go “Next”.

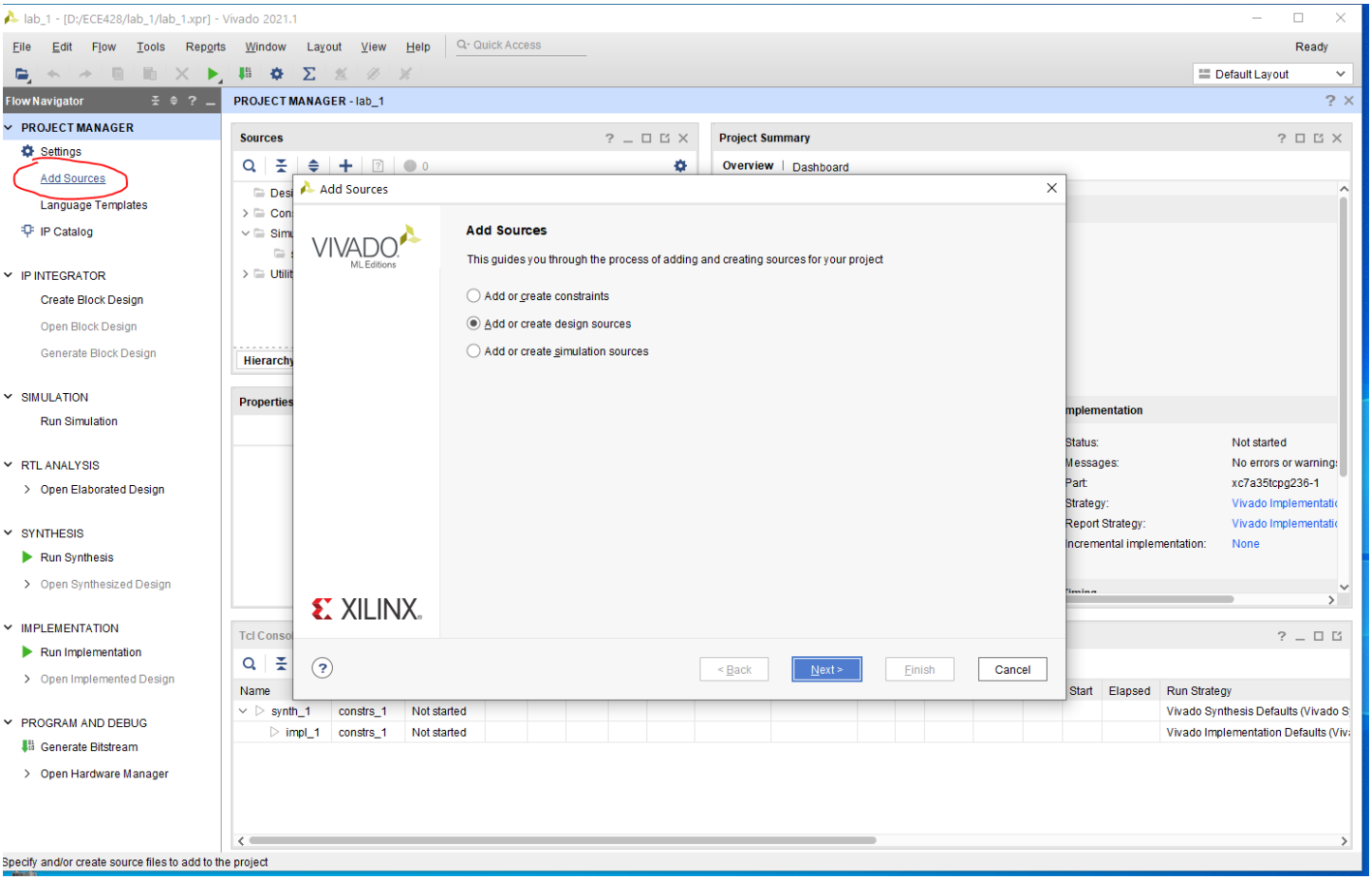



Fig. 7 Create a source file.

In the popped up “Add or Create Design Sources” window, click “Add Files” to add lab.v file (you may download it from D2L) to your project.

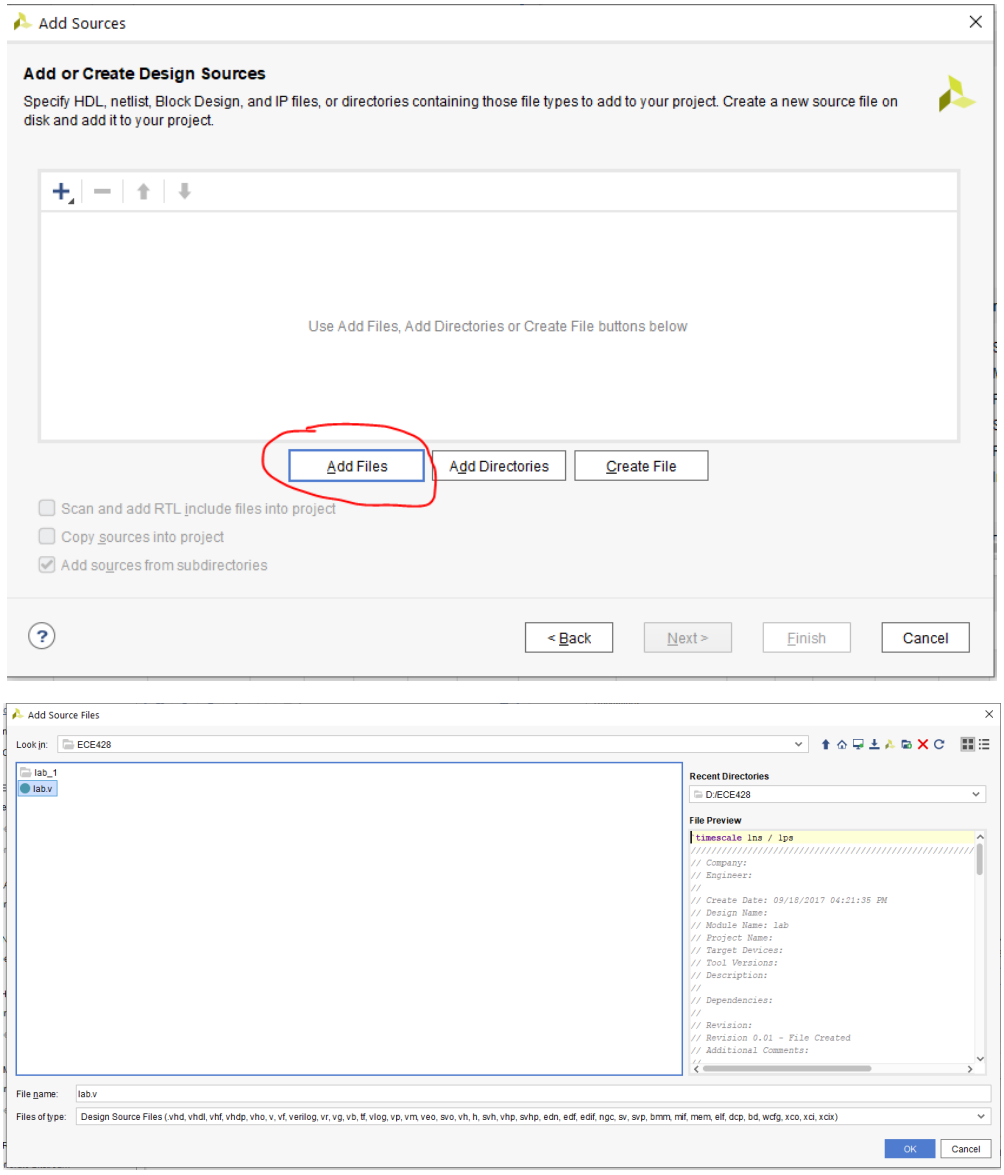



Fig. 8 Add a source file.

Alternatively, you may create new design files by clicking “Create Files”, In this case, a small popup window appears to allow you select file type and specify file name.

After adding or creating files, click “Finish” button to close “Add or Create Design Sources” window.

In the Project manager window, you may expand the design sources menu and double click lab.v to show the Verilog code

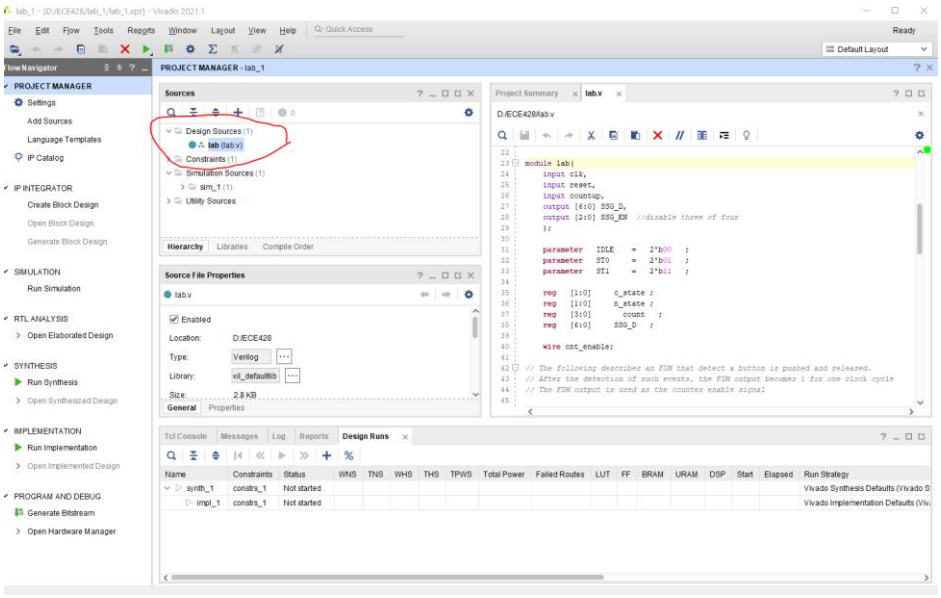


Fig. 9 Viewing lab.v code.

## Behavioral Simulation:

In the Flow Navigator panel, under “SIMULATION” menu, click “Run Simulation” and select “Run Behavioral Simulation” as shown in Fig. 10.

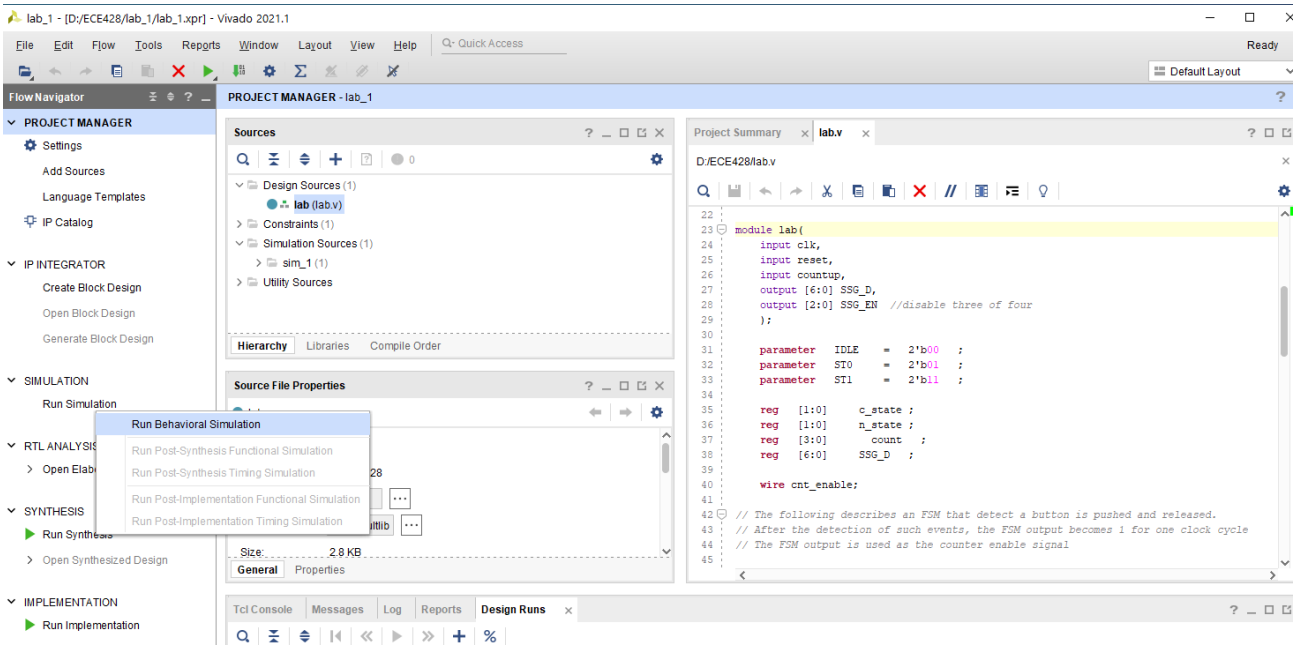



Fig. 10 Behavioral Simulation

A new simulation window will be opened to show the signal names and values. First, we change the simulation time after each click to 100 ns as shown in the figure below.

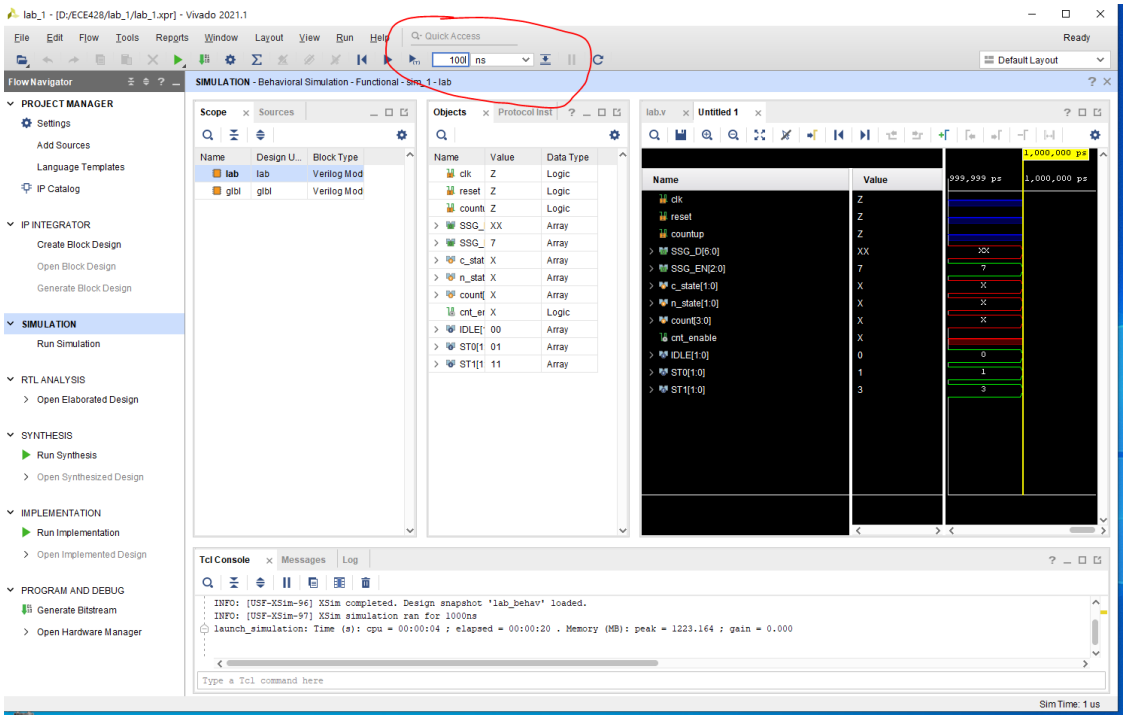
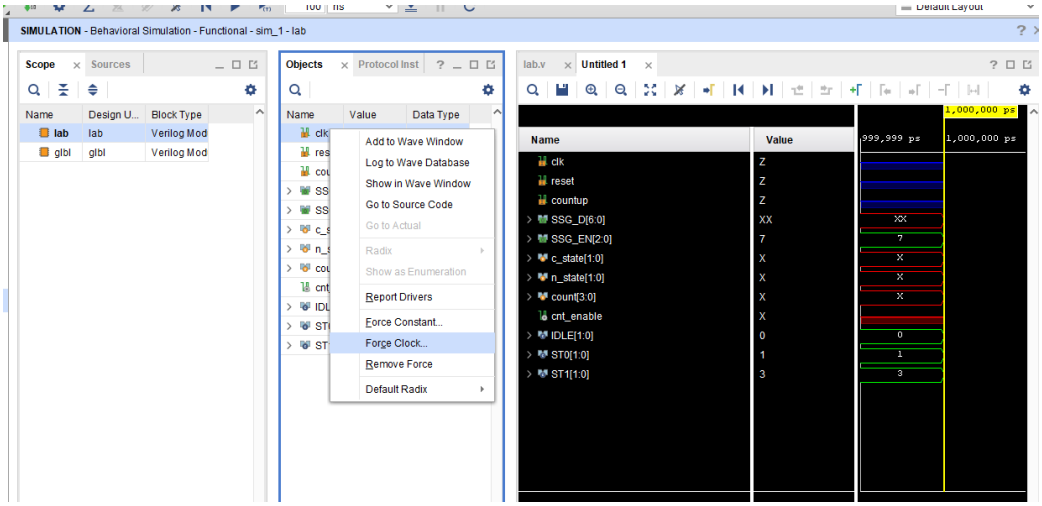


Fig. 11 Waveform window

Instead of writing a test bench, we first use the internal signal generator to provide stimulus for simulation. In the panel showing signal names, we right click clk signal and select “Force Clock”. In the popup window, we specify the clock period as 10 ns.



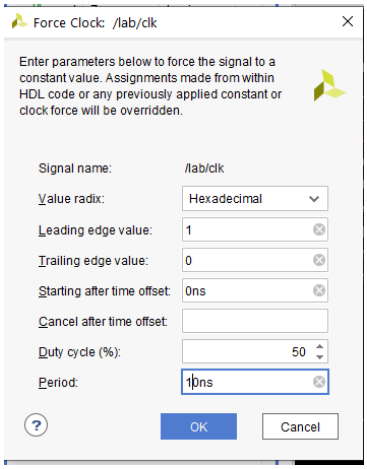


Fig. 12 Setting clock signal

Similarly, set countup signal as clock with period of 60 ns and assign 5ns delay as shown below.

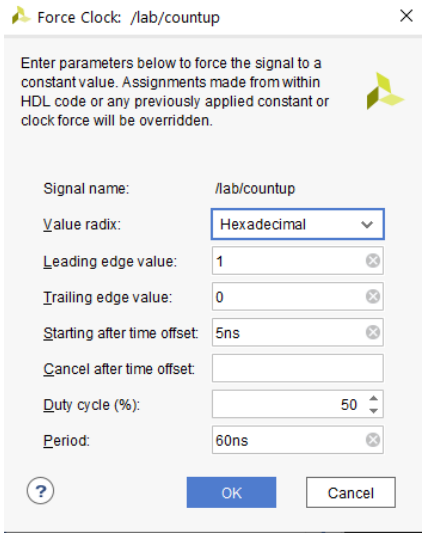



Fig. 13 Setting countup signal

In the panel showing signal names, right click reset signal and select “Force constant”. In set popup window, set its value to 1 (to reset the counter)

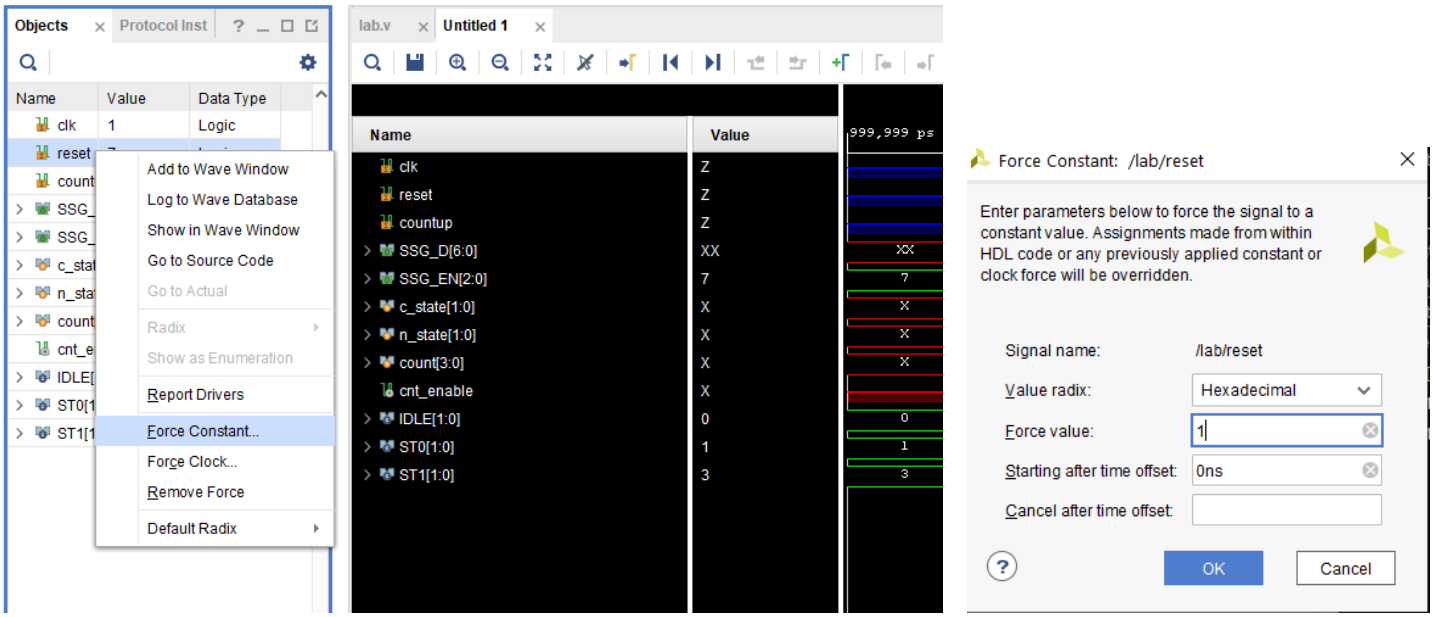



Fig. 13 assert reset signal

Click run icon to run simulation for 100 ns as shown below. Then, set the reset to 0 using force constant as shown Figure 15. Then run simulation for 2 us and check if the design behaves as expected by checking the simulated waveforms (as shown in Figure 16).

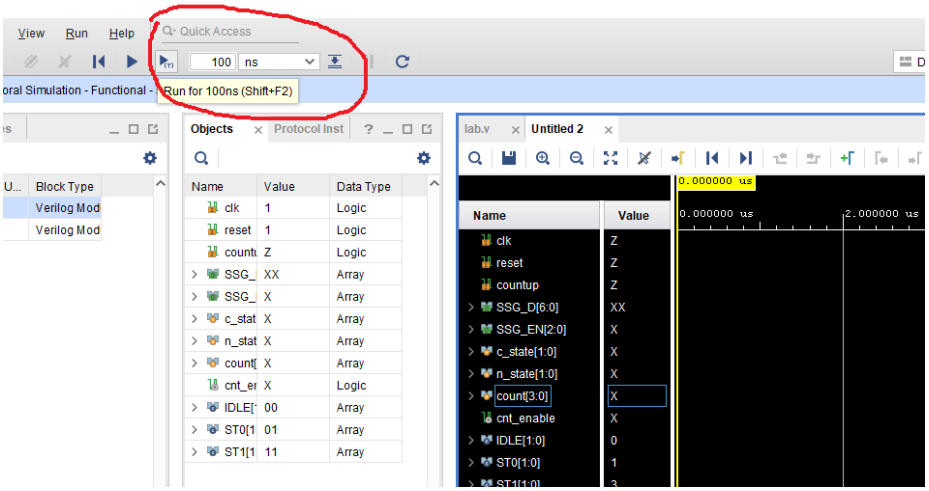



Fig. 14 Run simulation for 100 ns to reset the counter

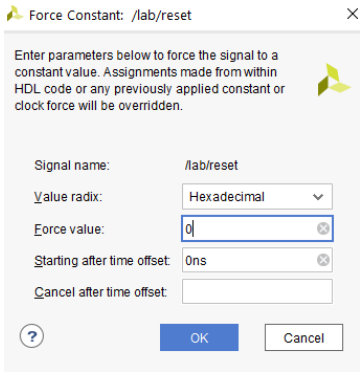



Fig. 15 dessert reset signal

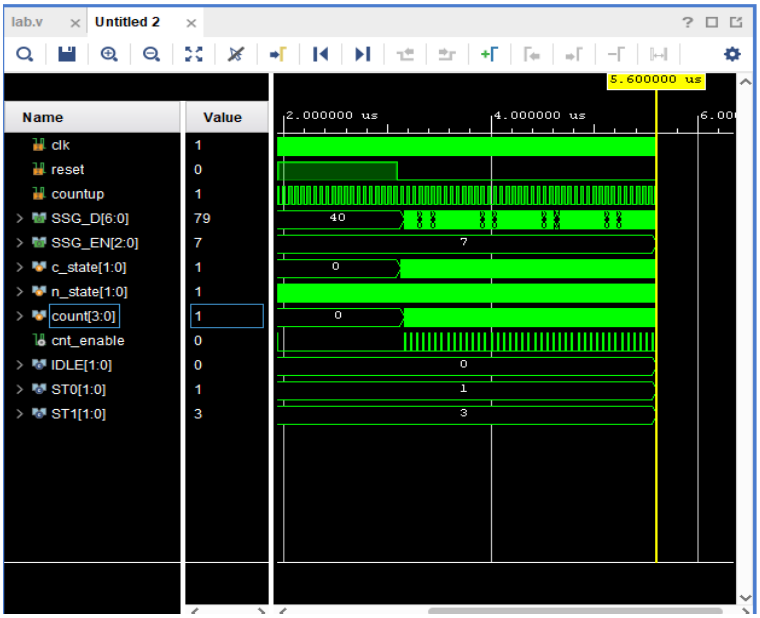



Fig. 16 Simulated waveform

# Perform Logic Synthesis

Logic synthesis generates netlist from RTL (Register Transfer Level) codes. In the Flow Navigator panel, click “Run Synthesis”. Use the default setup in the popup window and click OK to start synthesis process. It will take some time to complete the task. Once it done, a popup window will appear. It is shown on the right side of the figure below. In the popup window, we may select “Open synthesized Design” to view “Schematic” and synthesis reports.

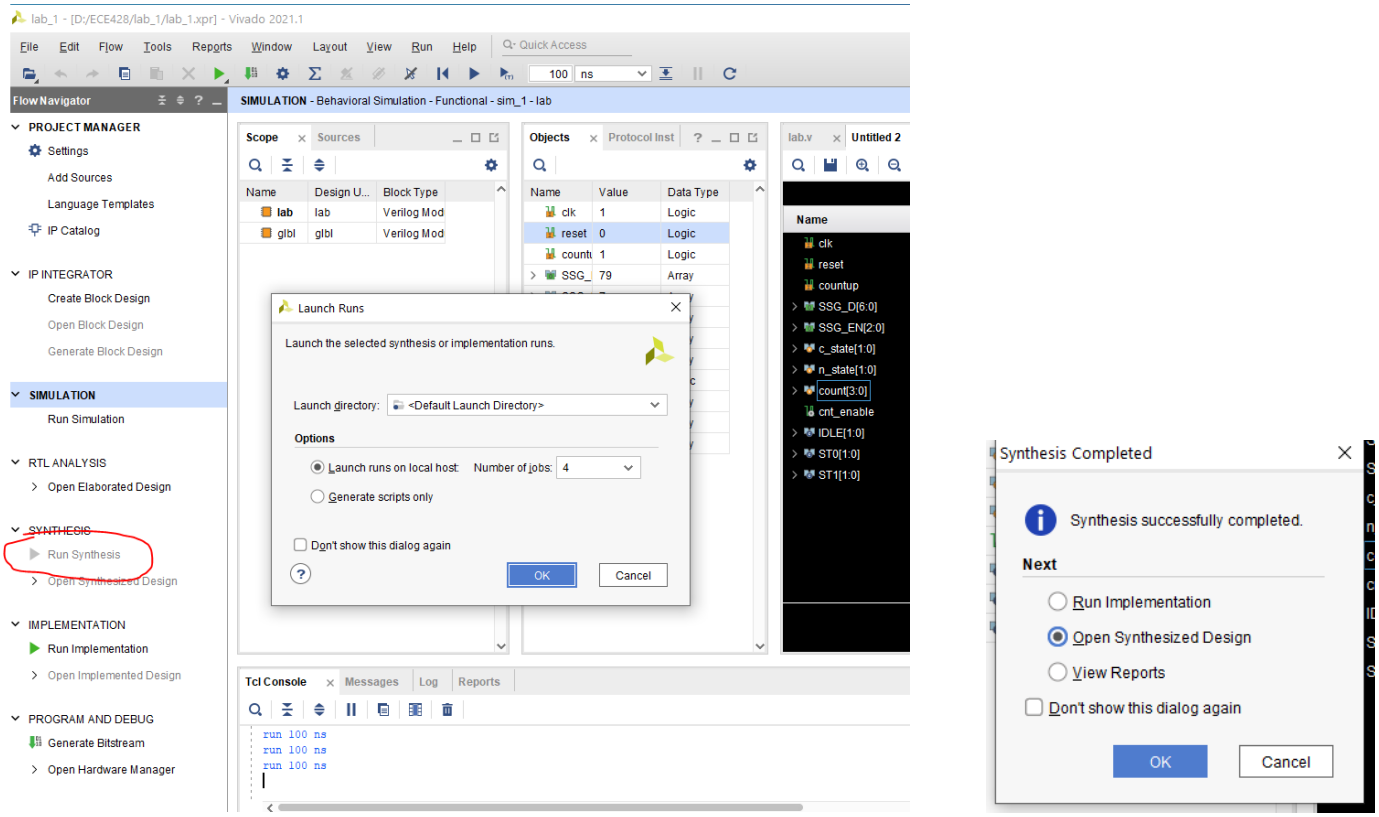



Fig. 16 Running logic synthesis

You will be able to see the project summary, schematic, device layout, and original RTL code. This is shown in Figure 17. Also, in the Flow Navigator panel, under SYNTHESIS→Open Synthesized Design, several reports are available to review. This is shown in Figure 18.

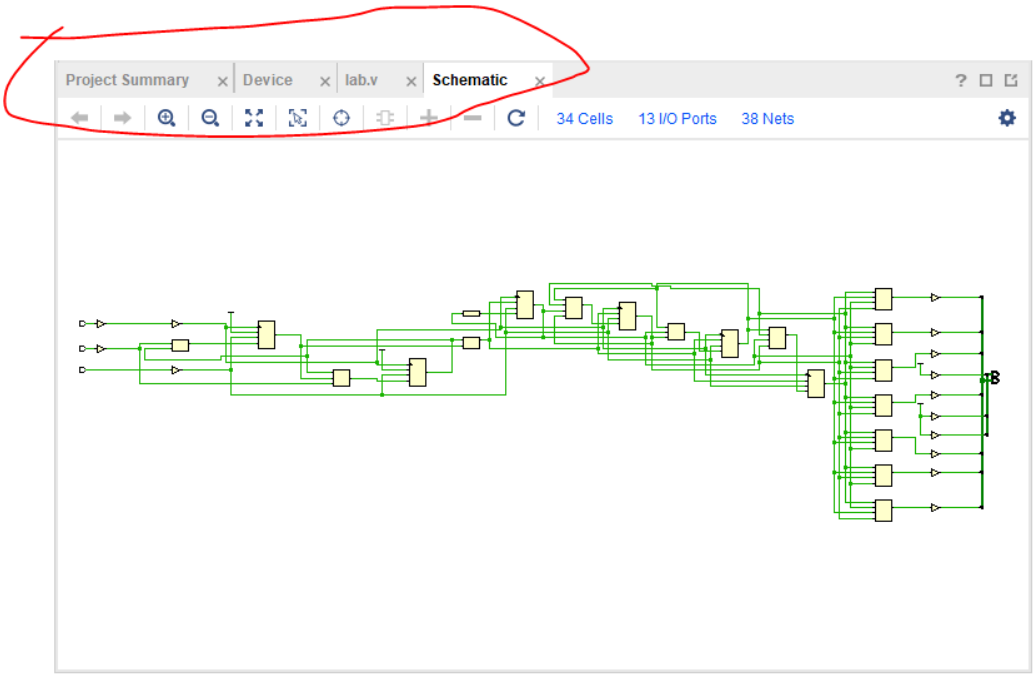



Fig. 17 Schematic of the synthesized circuit

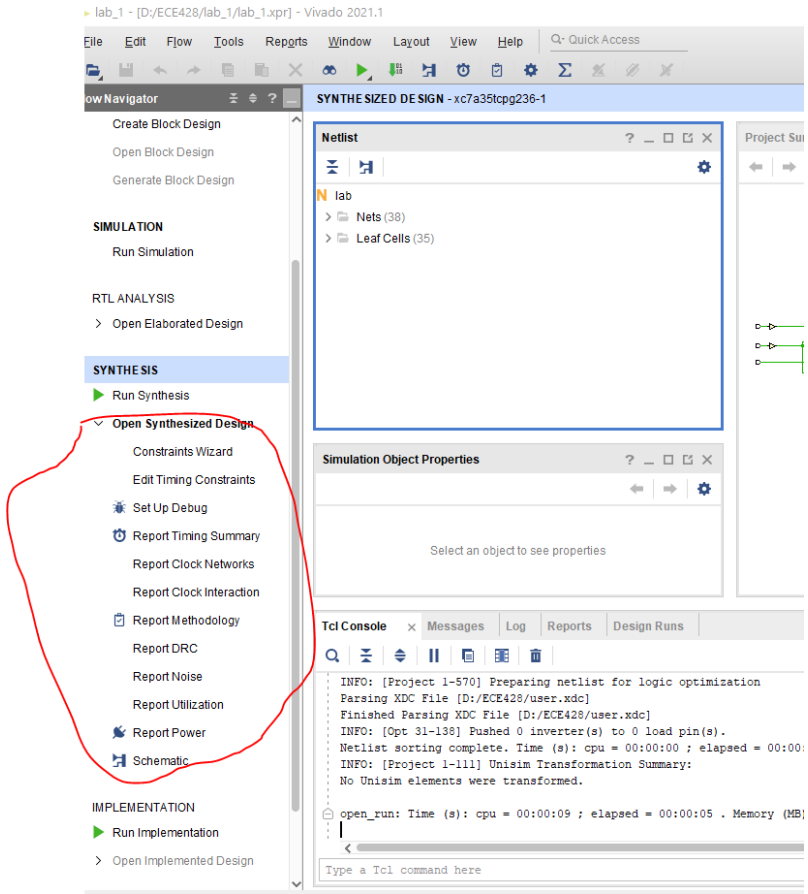



Fig. 18 Synthesis reports

## Post-Synthesis Timing Simulation

There are some differences between post-synthesis timing simulation and behavior simulation. For behavior simulation, timing information is not considered. All logic cells and data paths are considered as ideal, indicating no delay will be incurred when a signal is passing through. However, this is definitely not the real case, where delays existing everywhere along the logic paths. Post-synthesis timing simulation (PSTS) does functionality check utilizing synthesized circuit and delay information of logic cells from a certain library used in synthesis. Therefore, PSTS provides a more accurate and valuable timing analysis than behavior simulation.

Once you completed synthesis, you can run post-synthesis simulation as shown below. There are two options: one is for function simulation and the other is for timing simulation. We will select timing simulation. Previously, these two options are unavailable.

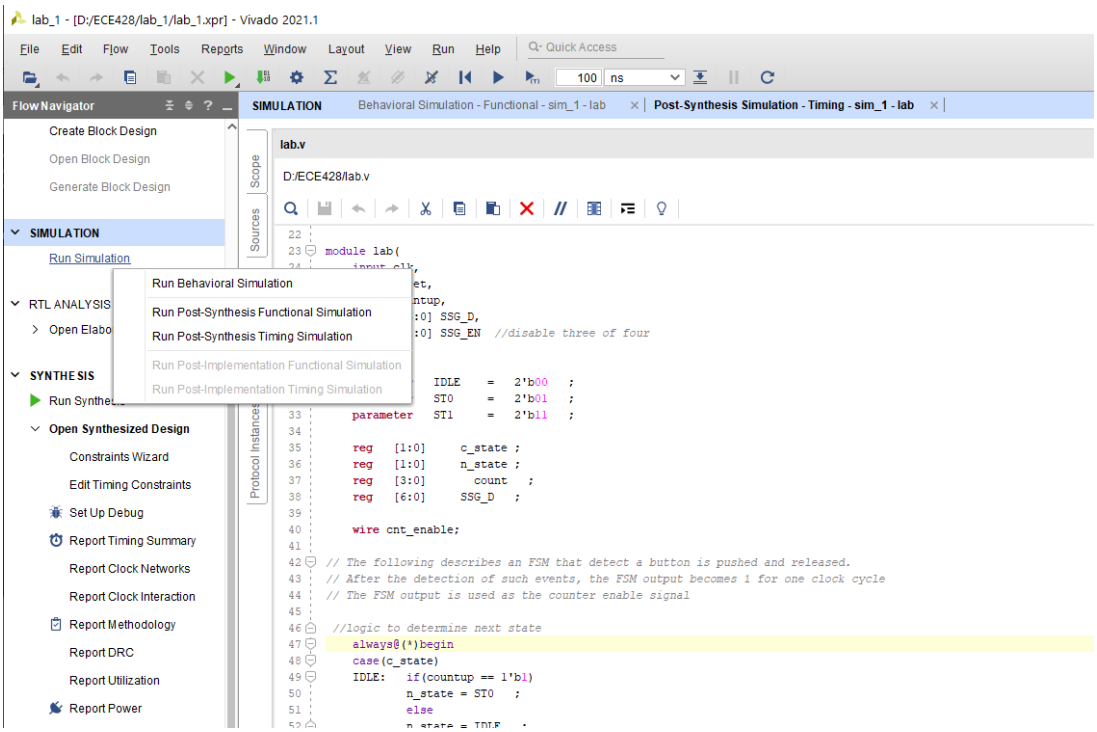



Fig. 19 Run post synthesis simulation

The setting of input signals is similar to behavior simulation. However, the resulting waveforms are a slightly different as shown in Fig. 20, where we can find signal edges are not aligned perfectly. This is caused by inherent cell delays along logic paths.

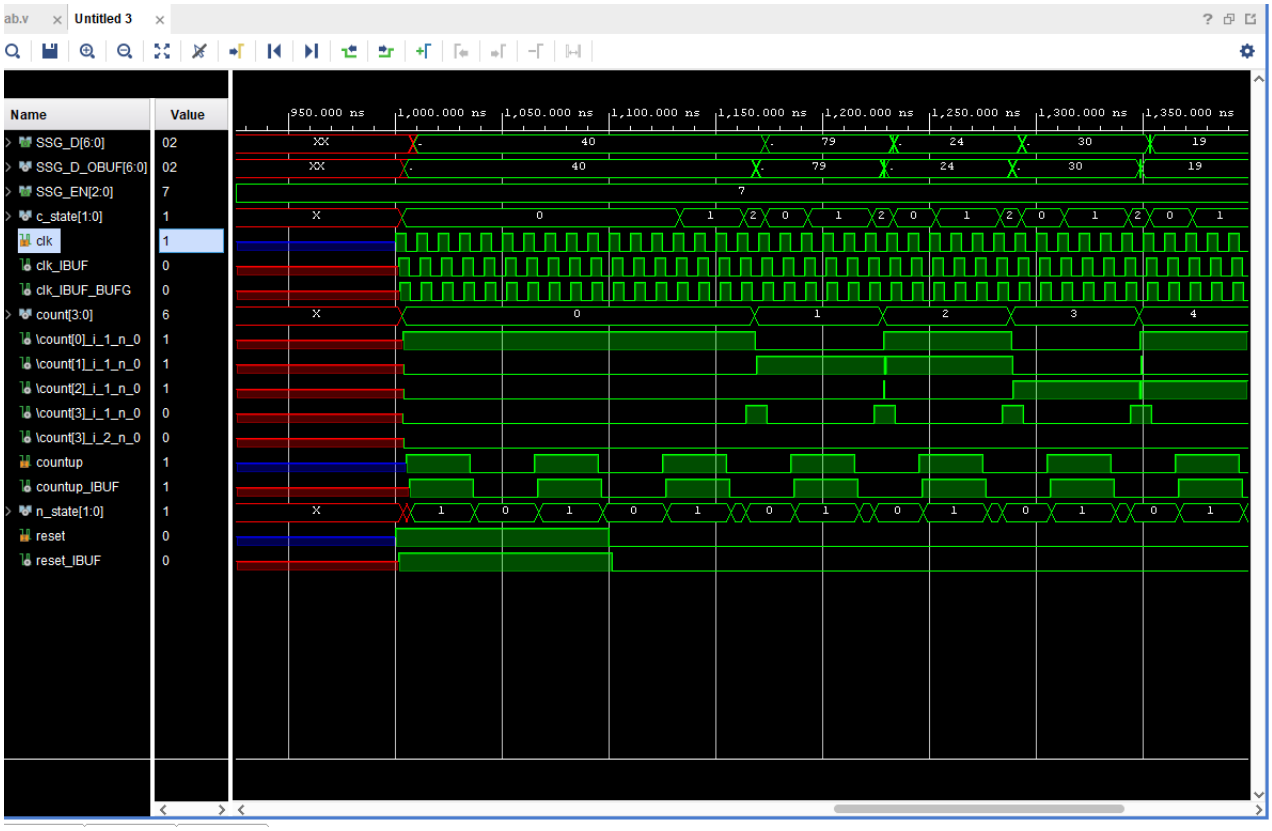



Fig. 20 Post synthesis simulation results

## Implementation:

The next step is to implement the synthesized circuit on the FPGA. It is also referred to as placement and routing (P&R). First, we need assign circuit inputs and outputs to FPGA Io pins. In the user constraint file, we already made such assignments. Fig. 21 shows how clock signal clk is assigned to FPGA pin W5 and specify it using CMOS logic standard with power supply of 3.3V.

```
## Clock signal

set_property PACKAGE_PIN W5 [get_ports clk]

set_property IOSTANDARD LVCMOS33 [get_ports clk]

create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
```

Fig. 21 Clock pin assignment.

In Flow Navigator panel, under IMPLEMENTATION, click Run Implementation. Use the default setting in Launch Runs window. Click OK. Once the implementation process is complete, a small popup windows appears as shown in the right side of the figure 22.

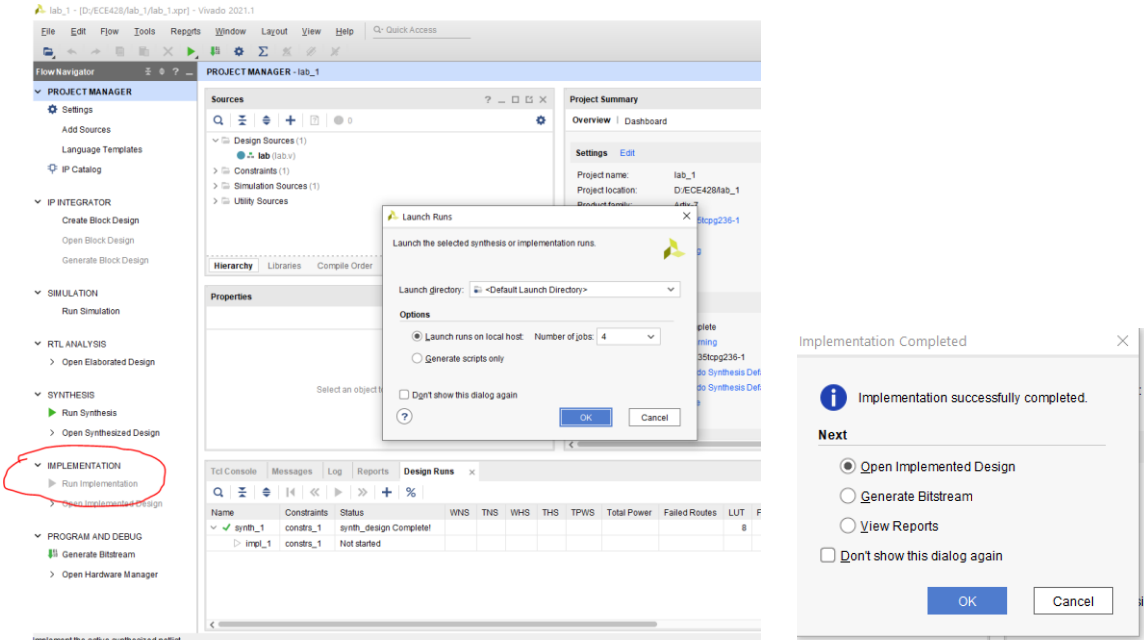



Fig. 22 Run Implementation

Select Open Implemented Design in the popup window and click OK. A panel showing FPGA layout appears and it show where the implemented circuit is located as shown in Fig. 23. On the upper right corner, you may select what types of information to show.

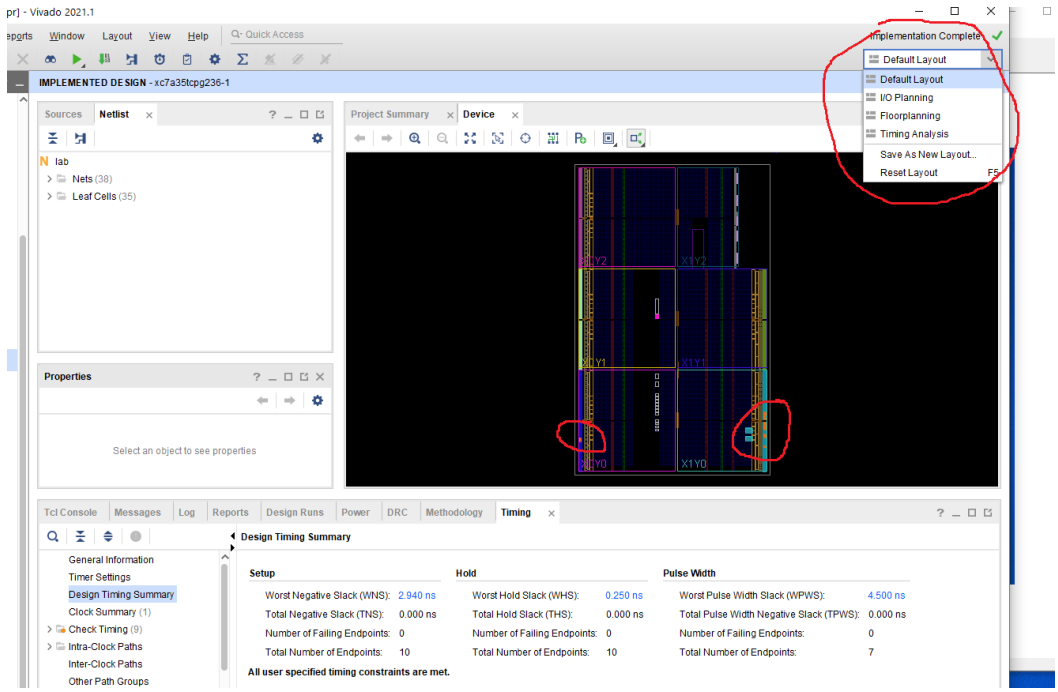



Fig. 23 Device layout

The FPGA resources used to implement the circuit are highlighted on the device layout as marked by the red cycles. Click on the “Show Cell Connections” icon in the “Device” tab and click “Show All”. If you zoom in

these areas, you will see the input and outputs are assigned to I/O pins, LUTs are used to implement the circuits. Click the components used in the circuits, it also shows connects for the select components as shown in Fig. 24.

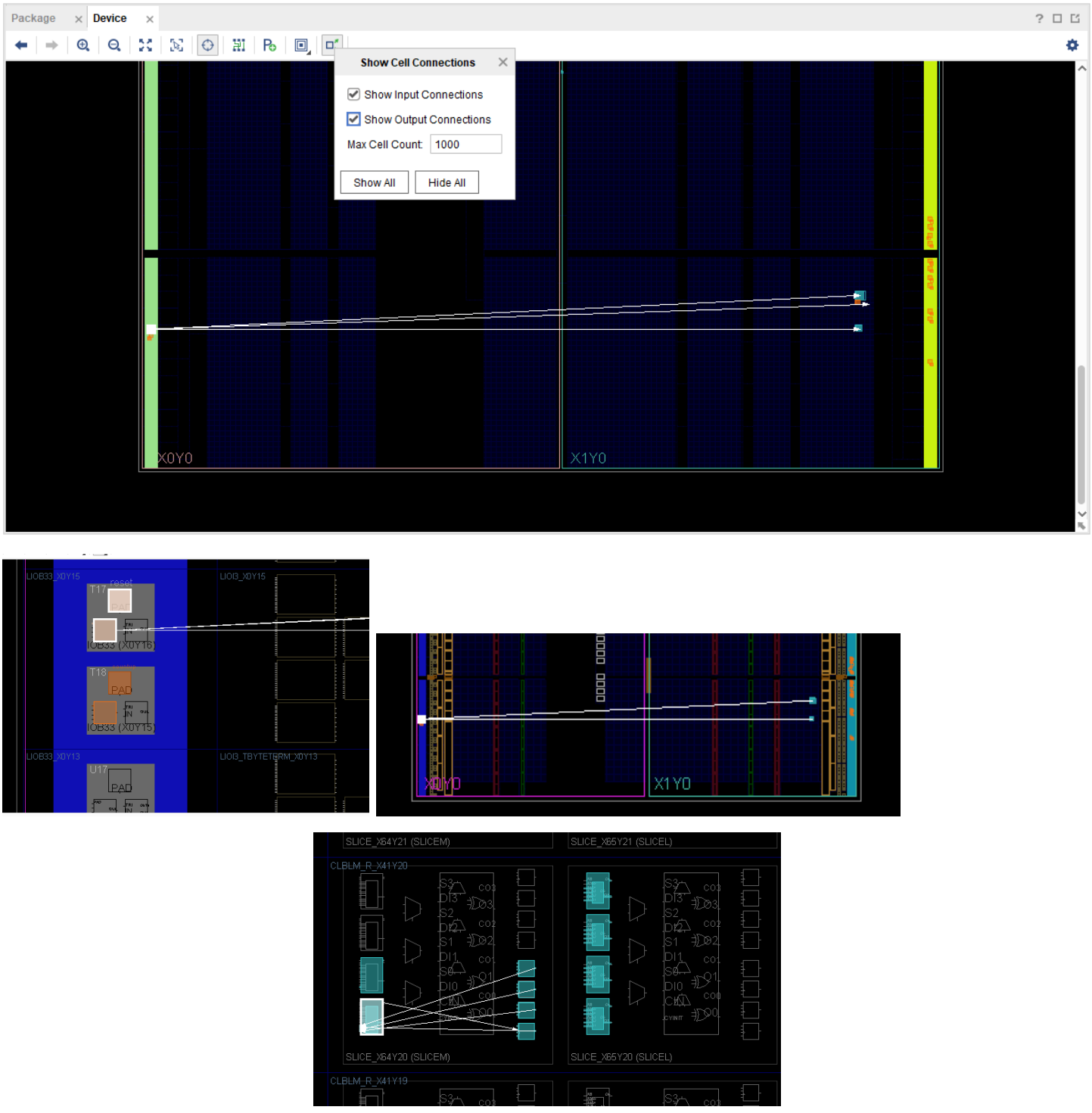



Fig. 24 Locating components used in the design

You can also select “I/O planning” from the upper right corner of the panel. It will show “I/O ports” and in the bottom panel and you can re-assign signals to different pins (different from the ones specified in user constraint file). This is shown in Fig. 25.

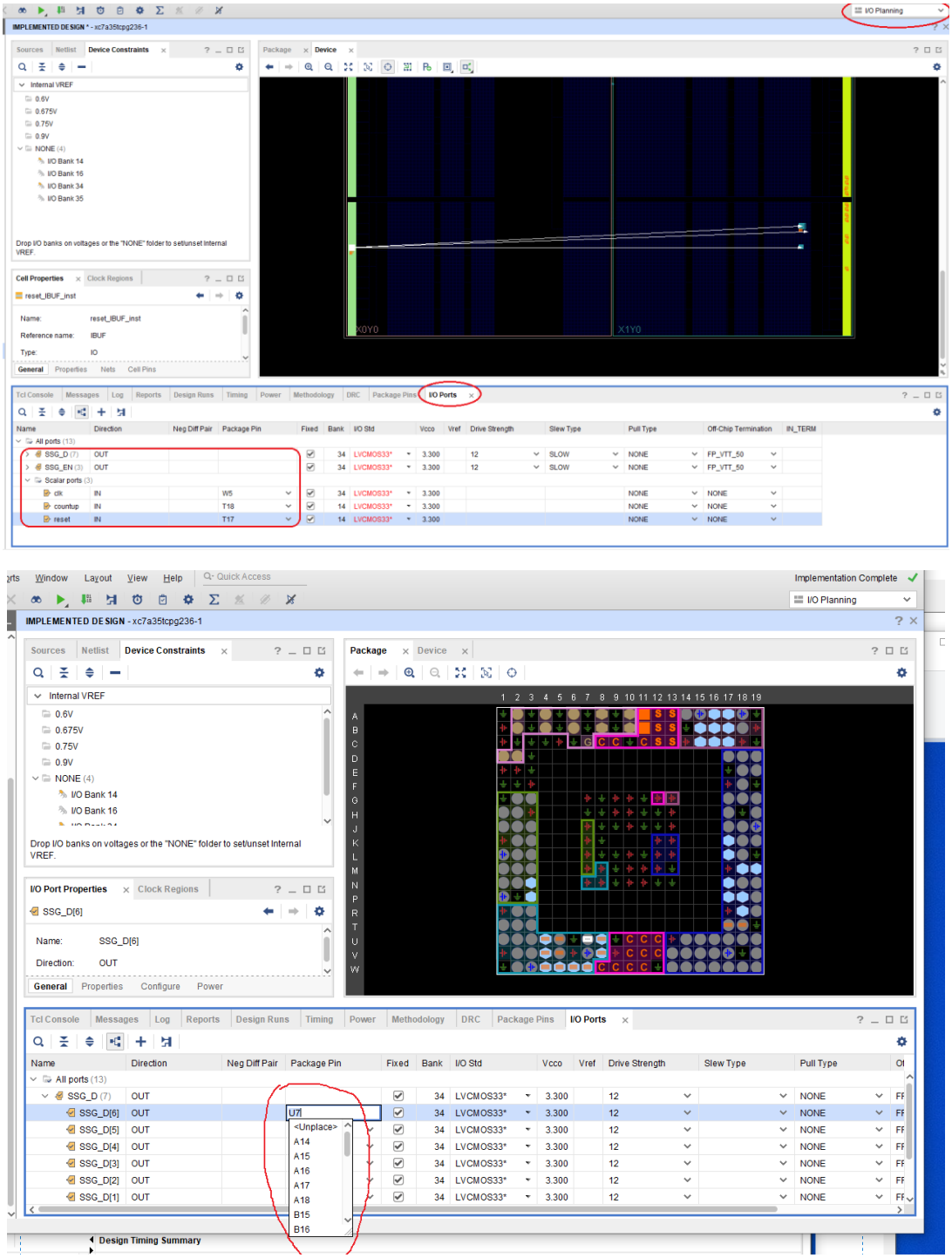



Fig. 25 I/O pin assignment

## Post-Implementation Timing Simulation

After implementation, more accurate information about delay caused by interconnect is available. We may perform simulation to check the timing again. In the Flow Navigator panel, under SIMULATION, click Run Simulation and select Run Post-Implementation Timing Simulation. Following the similar steps as discussed earlier to setup inputs and run simulation. The obtained waveform is shown in Fig. 26.

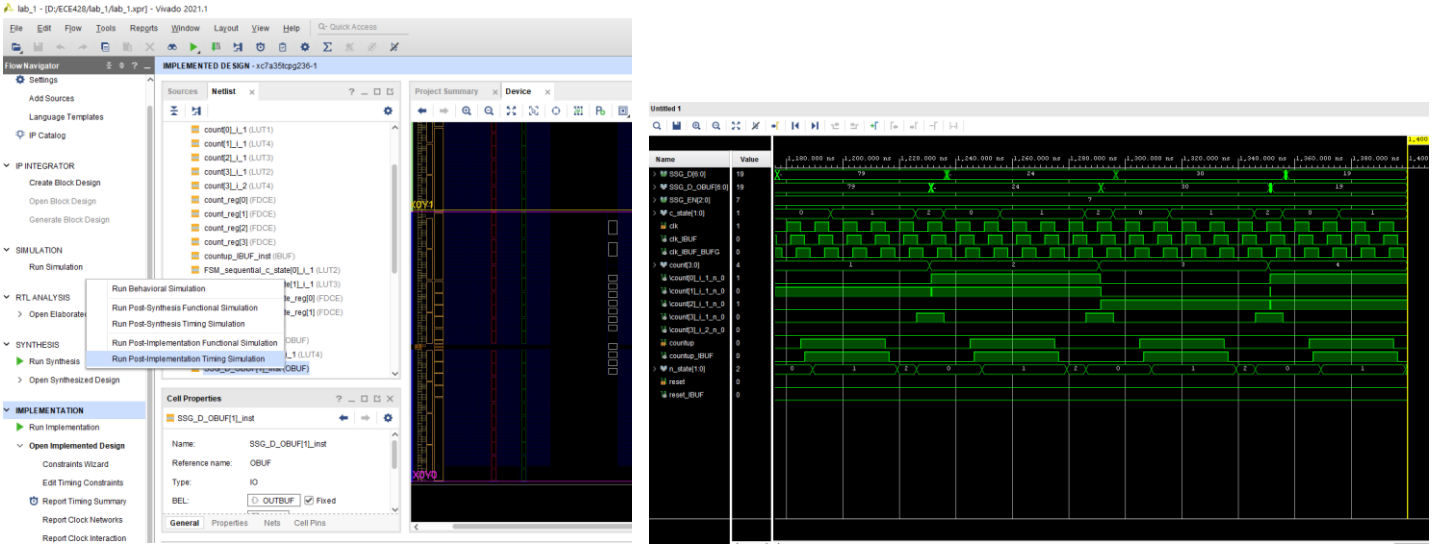



Fig. 26 Post-Implementation Simulation

## Generate Bitstream and program FPGA:

Once we verify the design via the aforementioned simulations, we can generate FPGA programming file. Go to "Program and Debug" at the bottom of the Flow Navigator. Click on "Generate Bitstream". A pop-up window will appear, click OK. A dialog with appear and show the successful completion of bitstream generation, select "Open Hardware Manager" and click OK as shown at the right in Fig. 27.

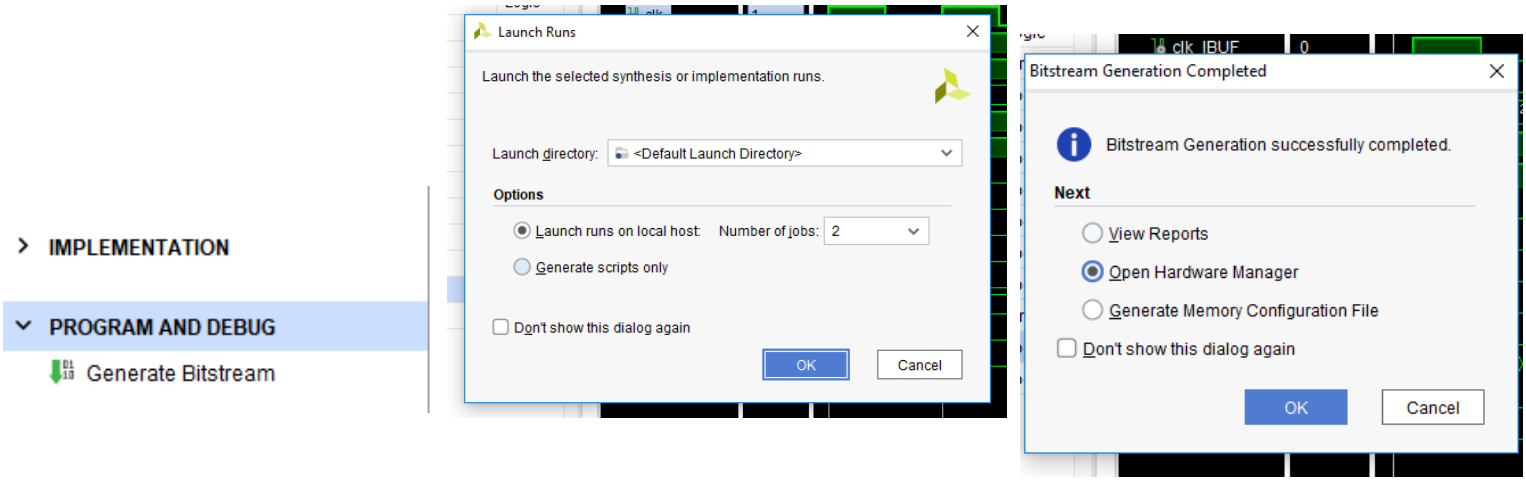



Fig. 27 Generate Bitstream

Once binary bit-stream file is generated successfully, go to “Open Hardware Manager” and click on “Open Target” then “Open New Target” as shown in Fig. 29.

Note: In order to find your board, you have to make sure USB cable is connected to you PC and the power supply of your board is turned ON. The result is shown in Fig. 30.

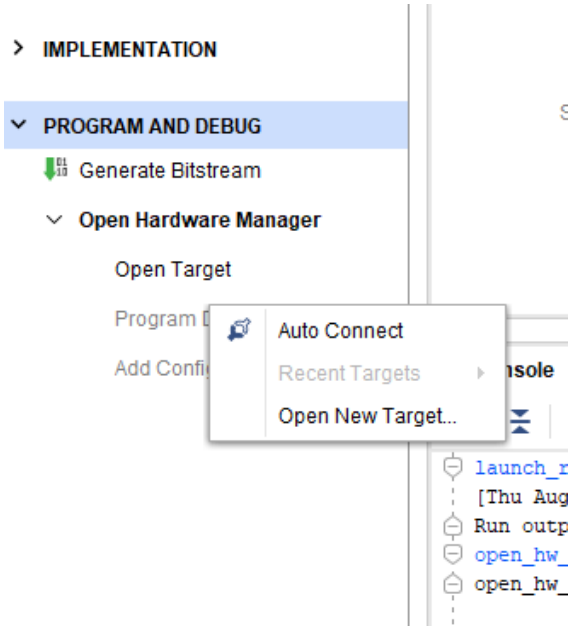
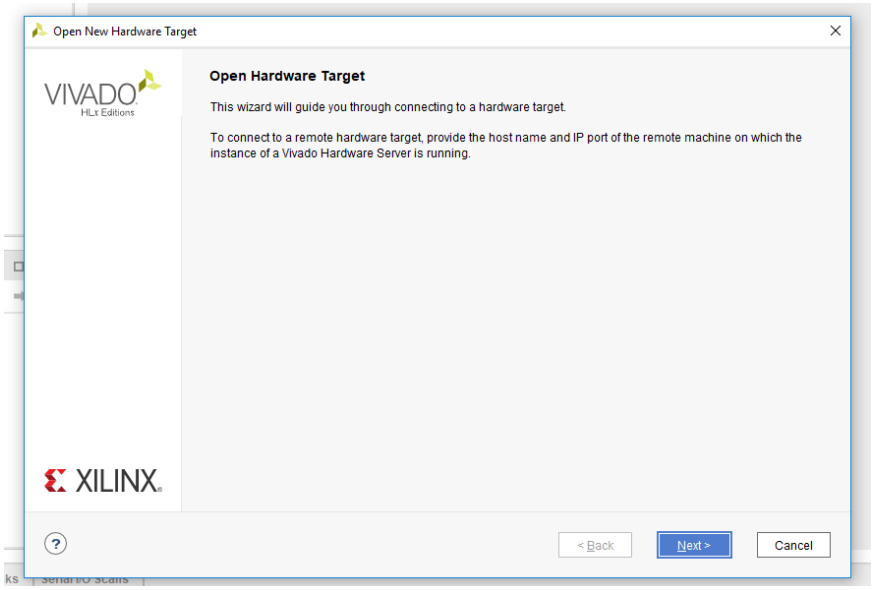
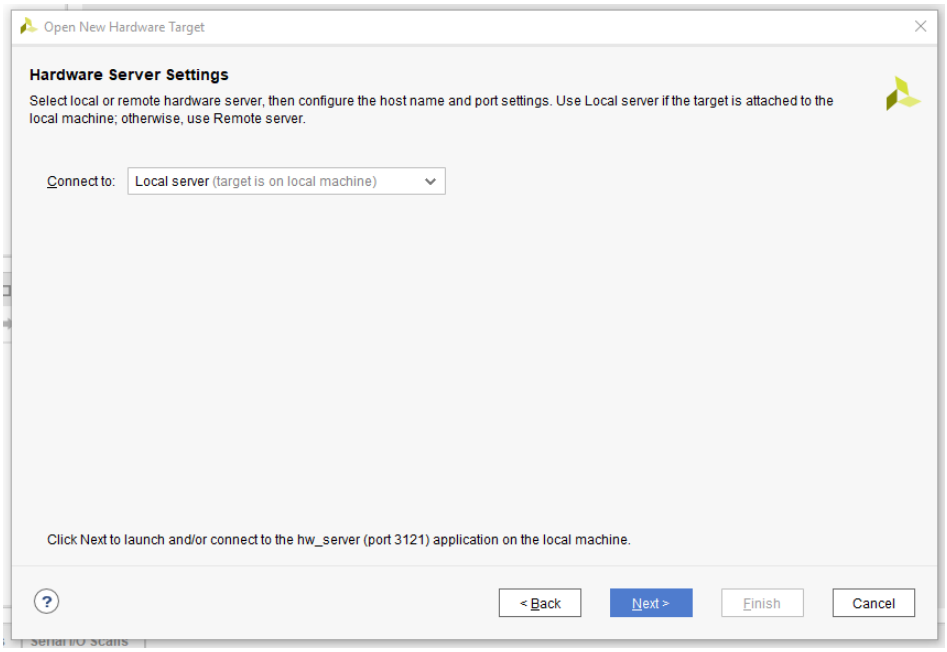



Fig. 29 Open New Target

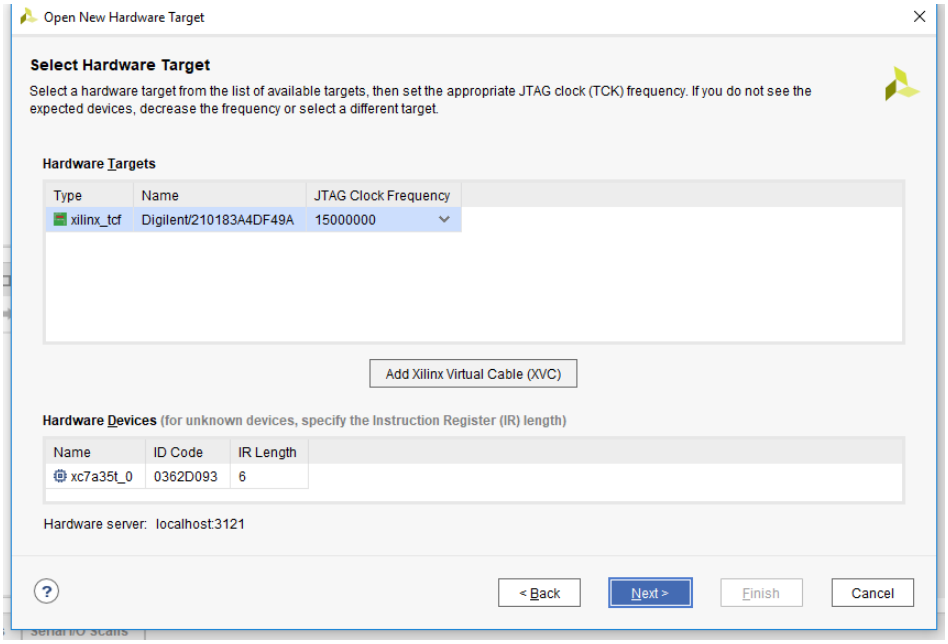
An “Open Hardware Target” window will popup, click NEXT as Fig. 30 (a). Then in “Hardware Server Settings” window, select “Local server (target is on local machine) option from the “Connect to:” dropdown and click NEXT as Fig. 30 (b). At the “Select Hardware Target” window you should see the Hardware Target with expected Hardware Device (xc7a35t_0), click NEXT and FINISH as Fig. 30 (c).




(a)




(b)




(c)

Fig. 30 Select the found device.

After successfully opening a device, click on “Program Device”, which should be active as long as your device is found.

Click on “Program” as shown in Fig. 31.

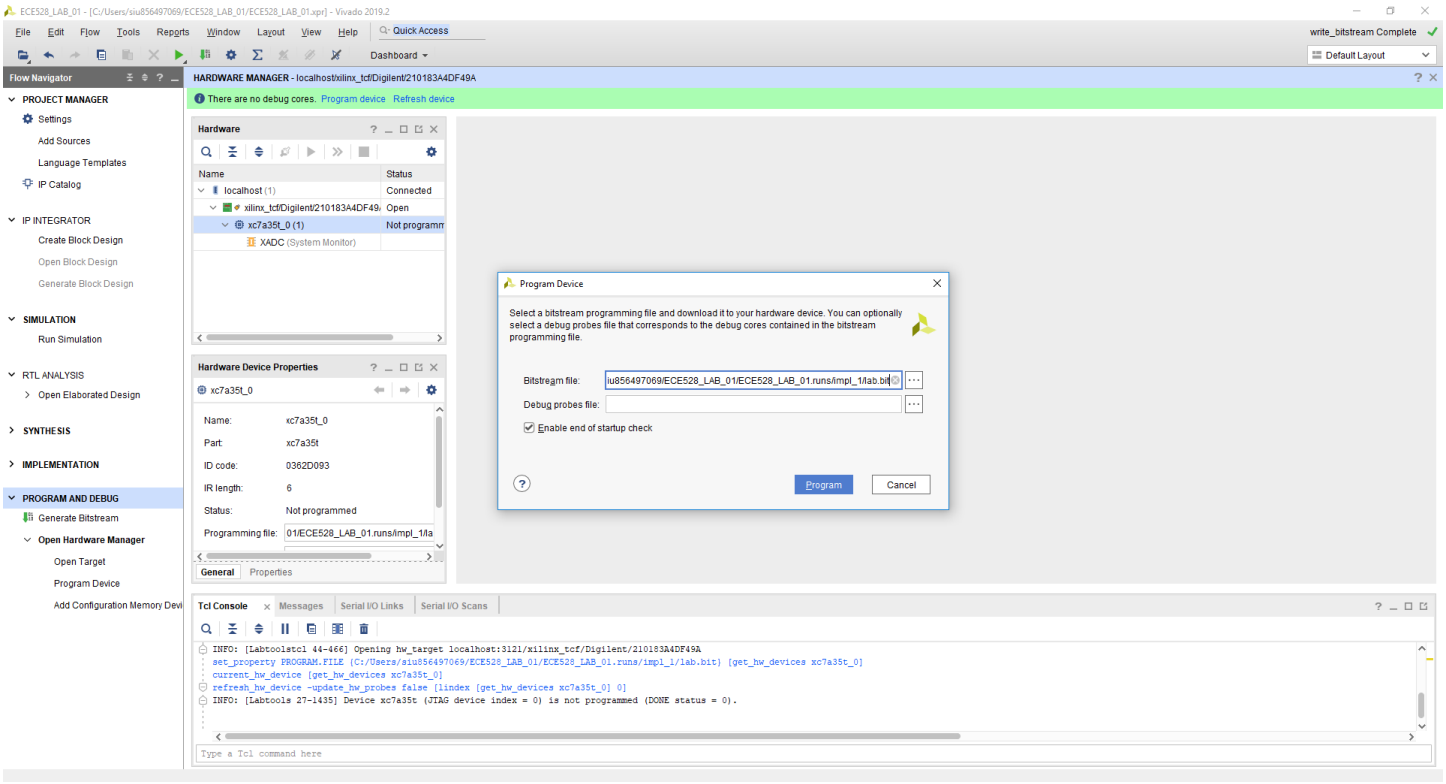
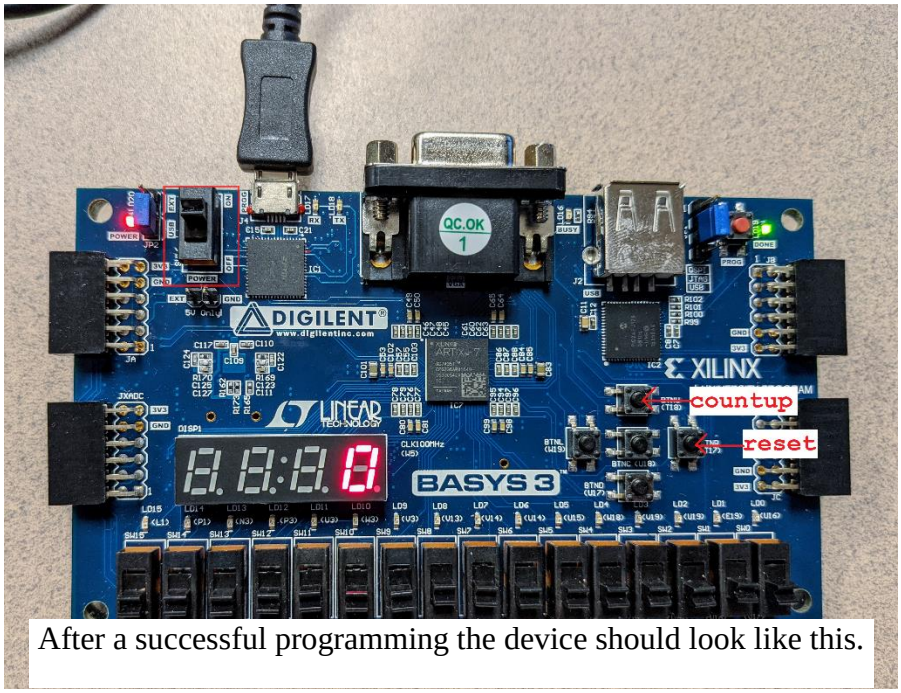



Fig. 31 Program your device.




After a successful programming the device should look like this.