ECE428/528 Programmable ASIC Design

Tutorial for Basys FPGA Board and Vivado Design Flow

Yuanzhi Zhang

Email: yzzhang@siu.edu

Department of Electrical and Computer Engineering

Introduction:

This tutorial demonstrates how to implement an FSM 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.

```
"1000000" when "0000"          //0
"1111001" when "0001"          //1
"0100100" when "0010"          //2
"0110000" when "0011"          //3
"0011001" when "0100"          //4
"0010010" when "0101"          //5
"0000010" when "0110"          //6
"1111000" when "0111"          //7
"0000000" when "1000"          //8
"0010000" when "1001"          //9
“1111111” when  “non of above” //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.

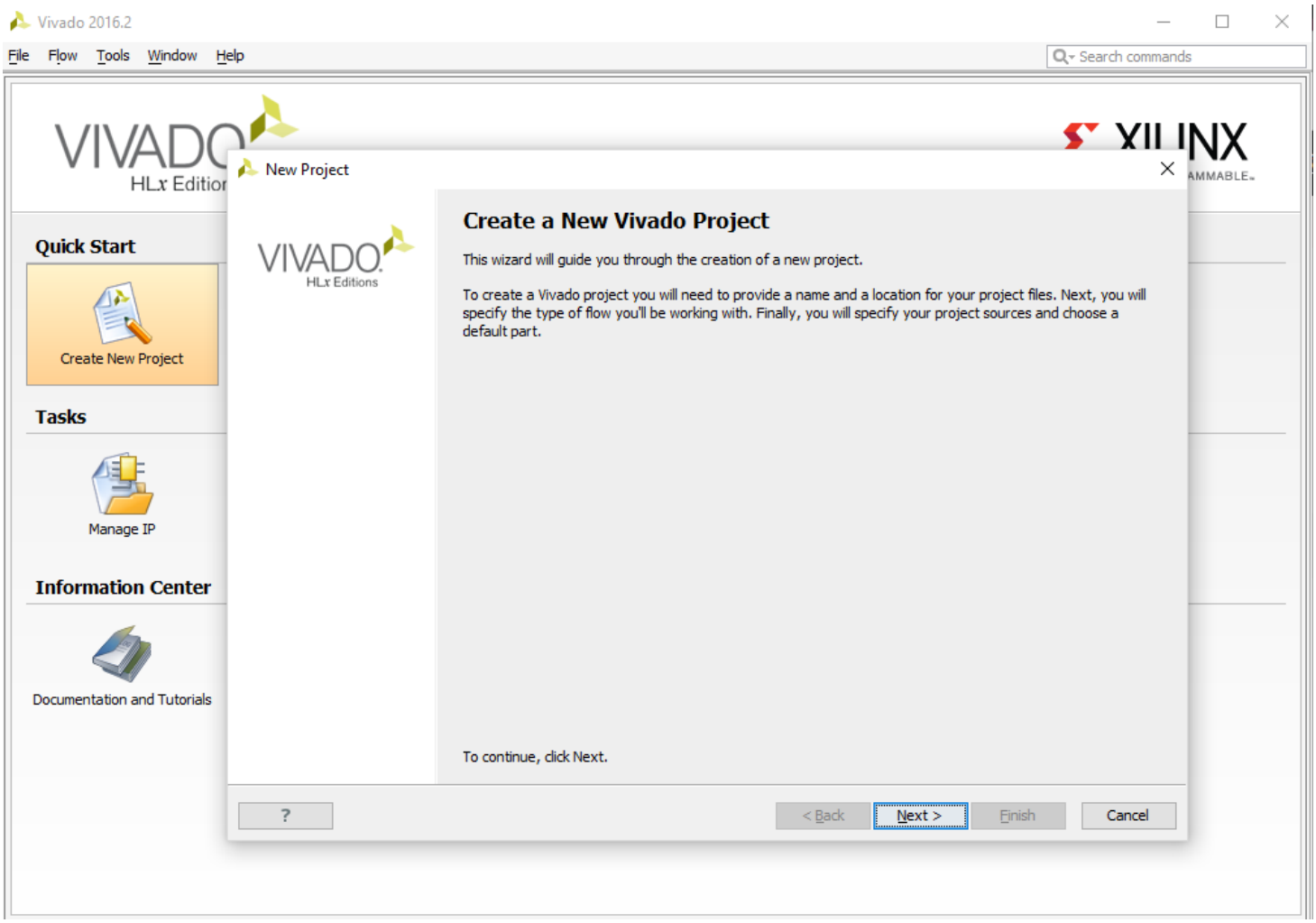

Fig. 1 Create a new project.

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”.

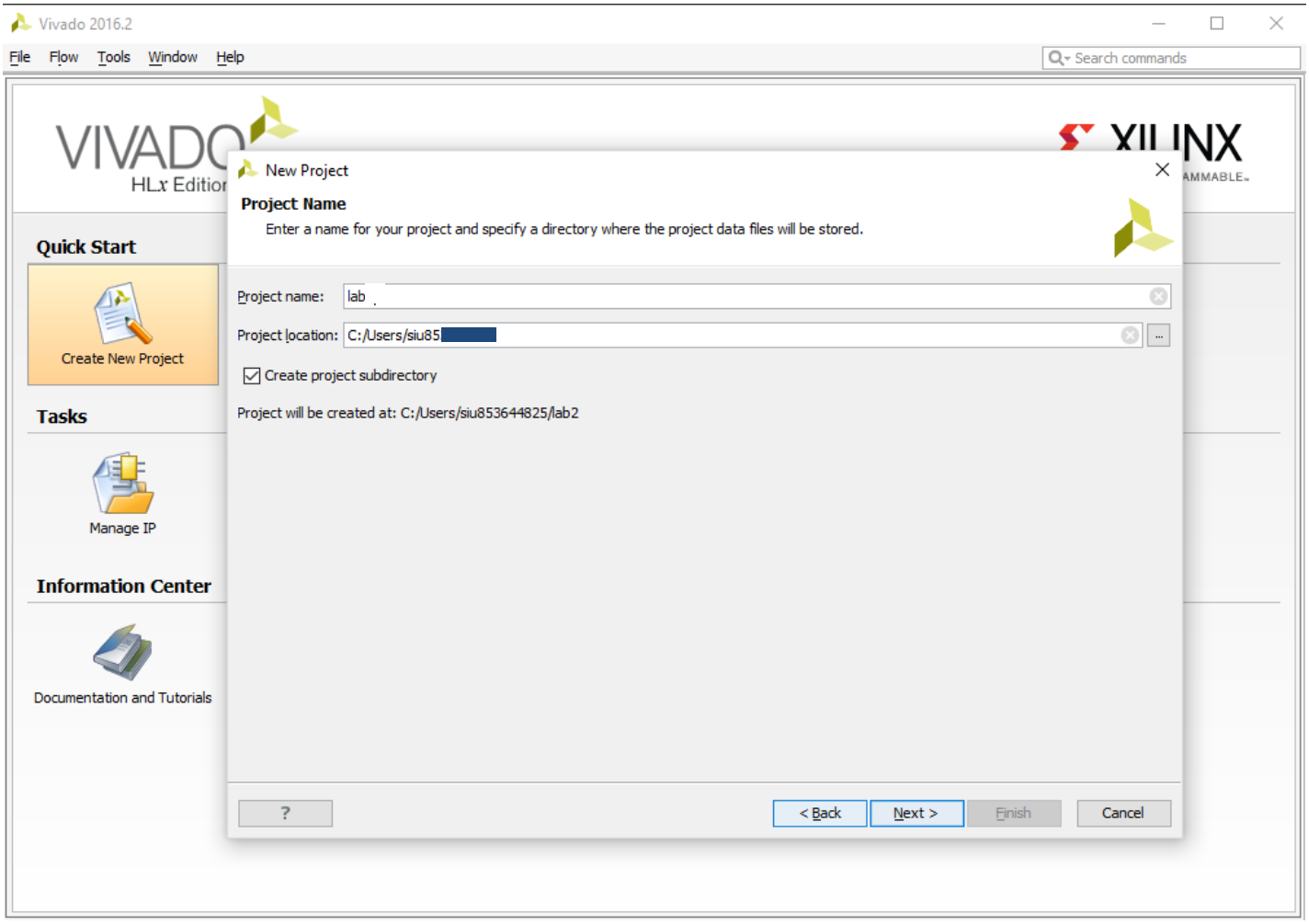

Fig. 2 Enter project name.

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

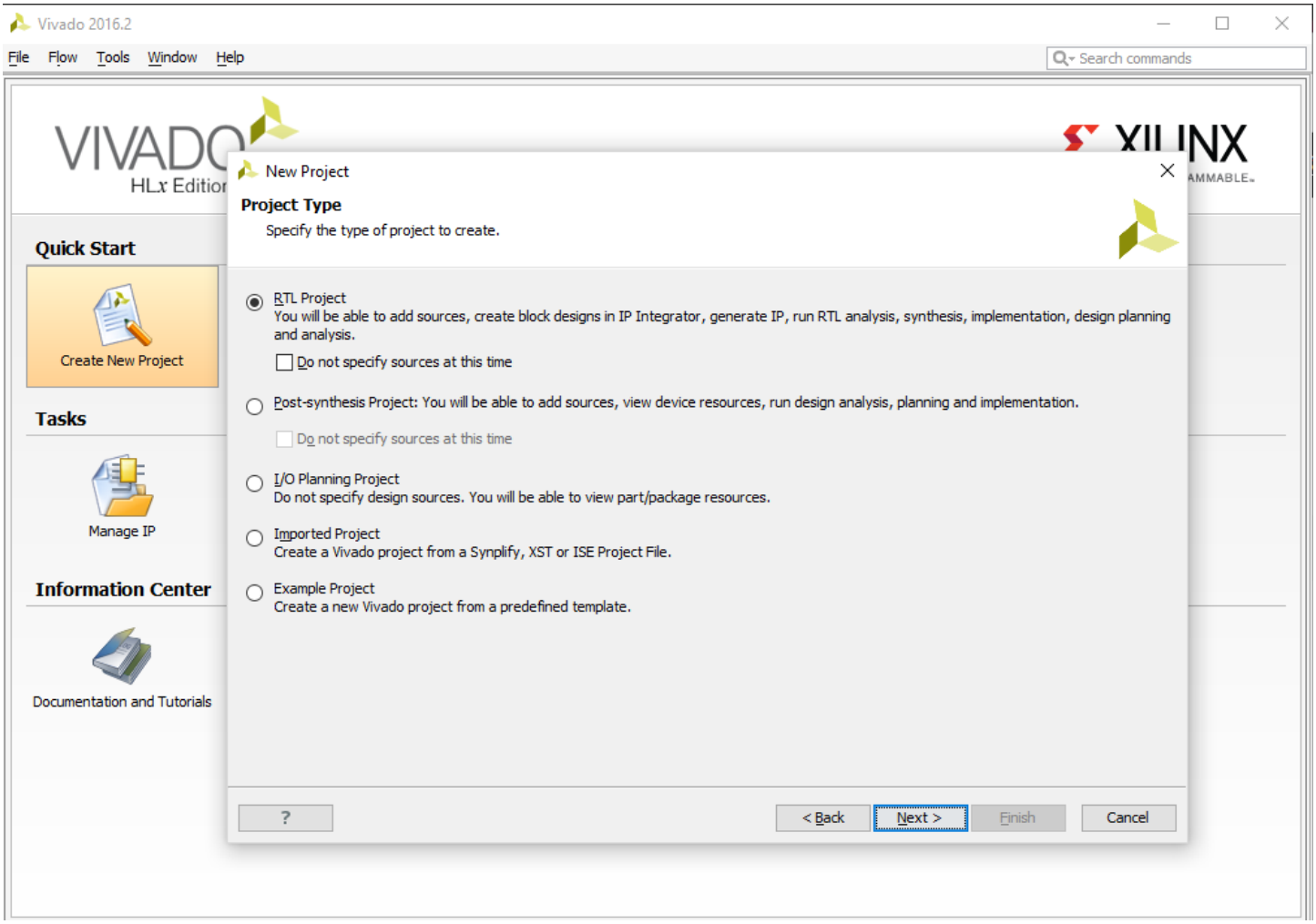

Fig. 3 Select project property.

Click “Next” unless you have source files. You can also create a source file in this step.

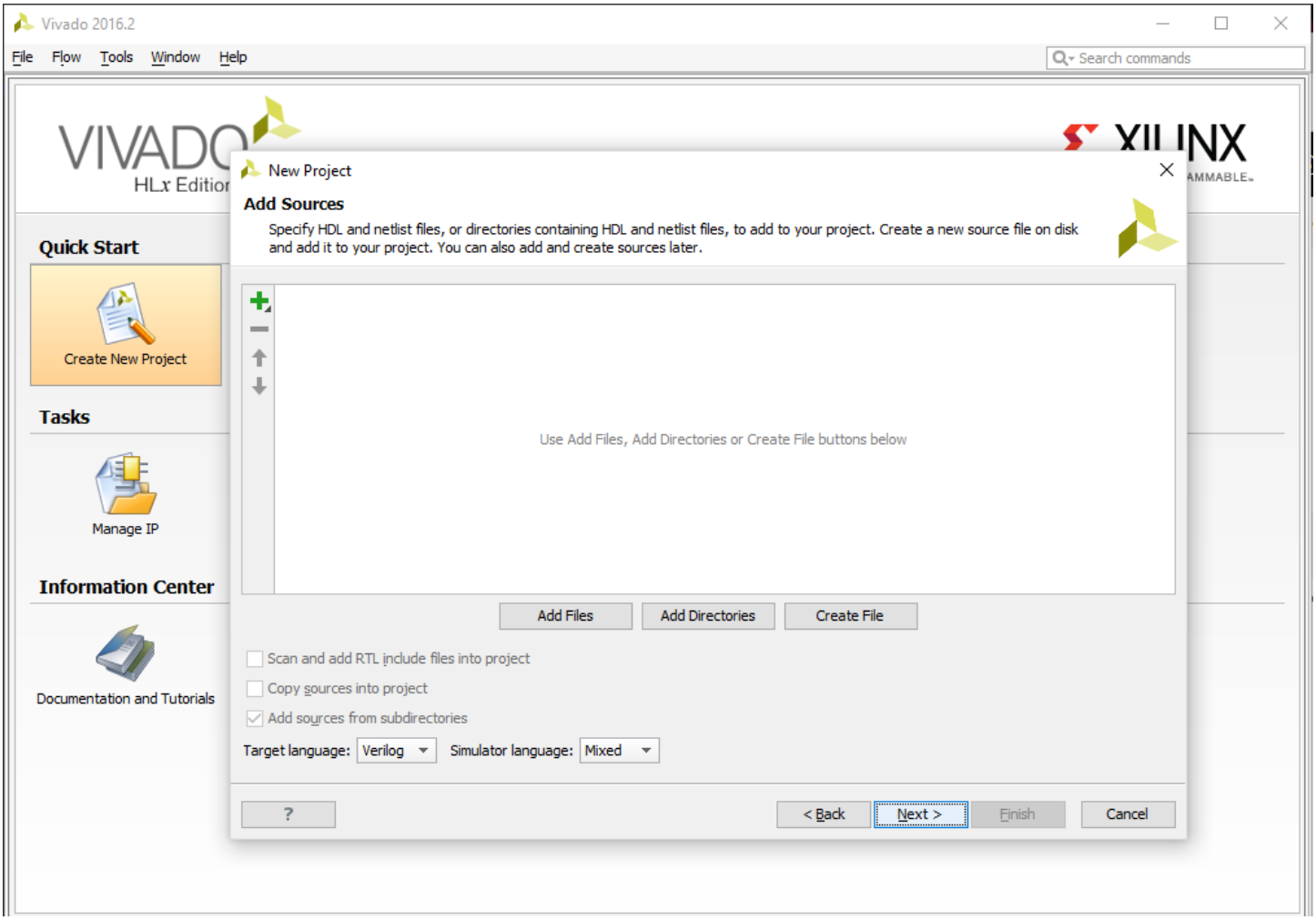

Fig. 4 Add and create source files.

Continue to click “Next” until the page for loading user constraints is popped up as shown in Fig. 5.

You can download the constraint file user.xdc from D2L and then add it to your project.

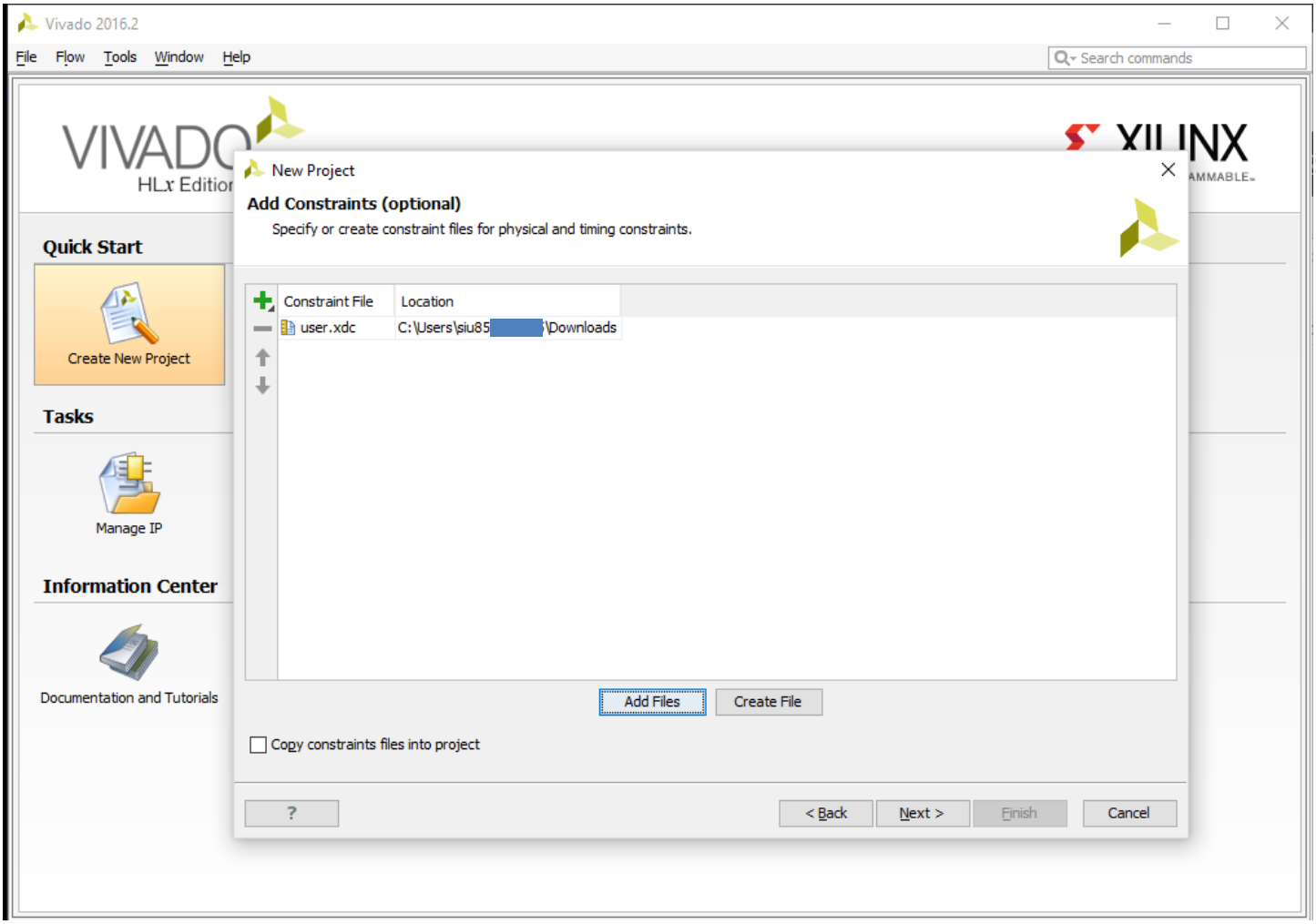

Fig. 5 Add user constraint file.

Select the correct Xilinx FPGA. The information about Basys 3 can be found online. Or you can simply configure according to Fig. 6. Then click “Next” until “Finish”.

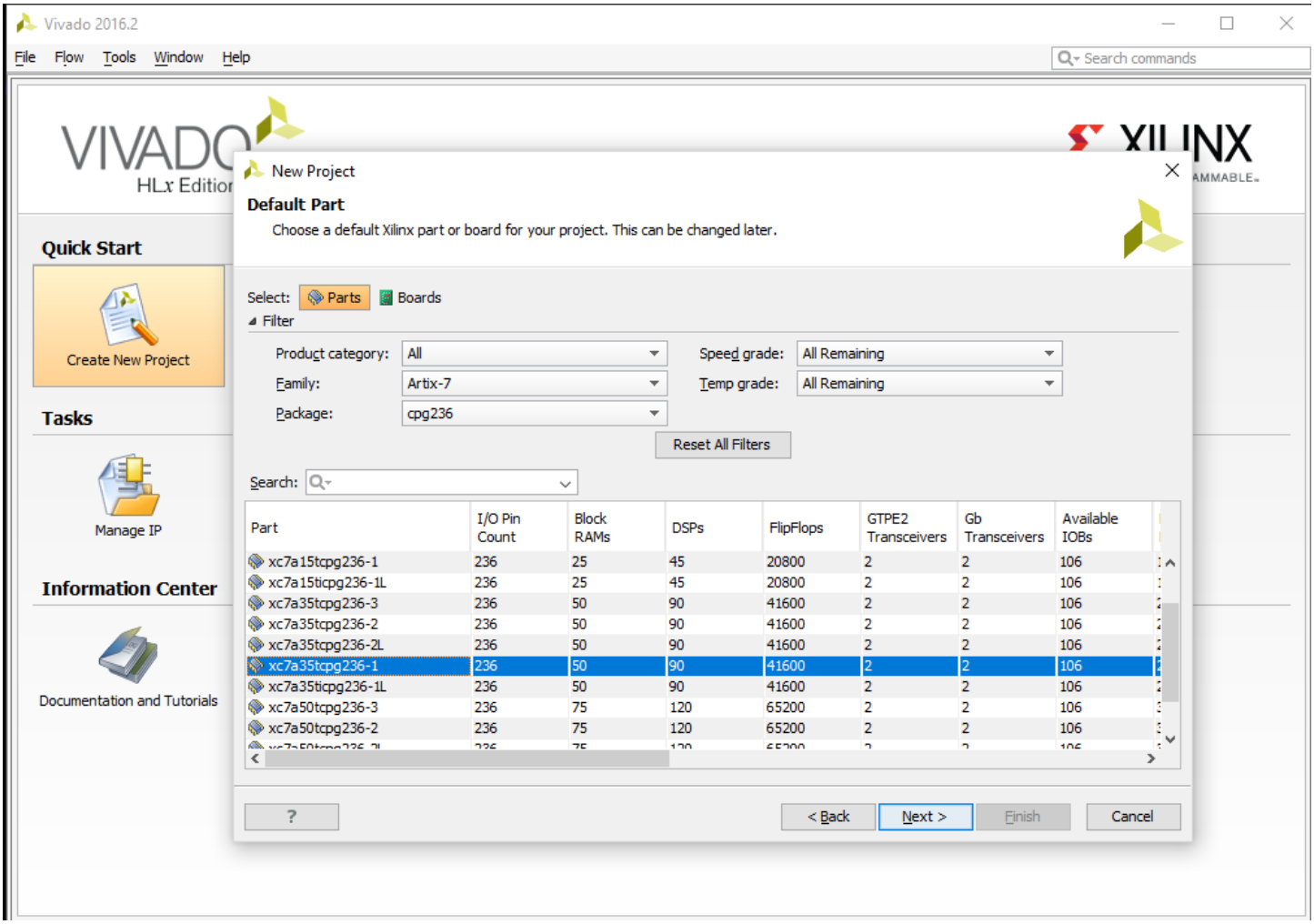

Fig. 6 FPGA part selection.

Add and create source files:

Go to “Project Manager” and select “Add Sources”. A pop-up window is shown as Fig. 7.

Select the correct item and go “Next”.

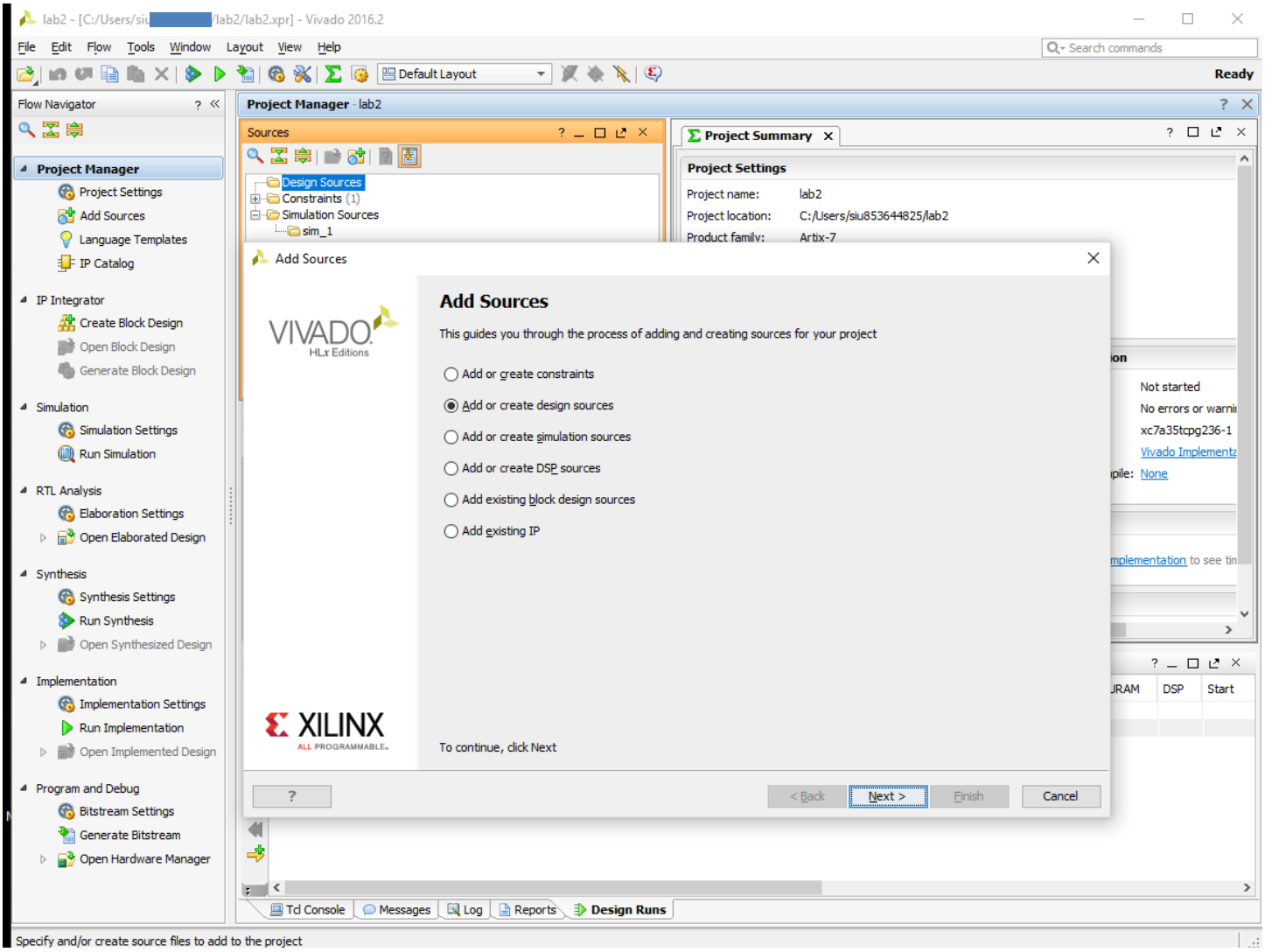

Fig. 7 Create a source file.

Create a Verilog file with file name. If you prefer VHDL file, then change file type to “VHDL”. The Verilog file used in this tutorial can be downloaded from D2L. The file name is lab.v.

Then click on “OK”.

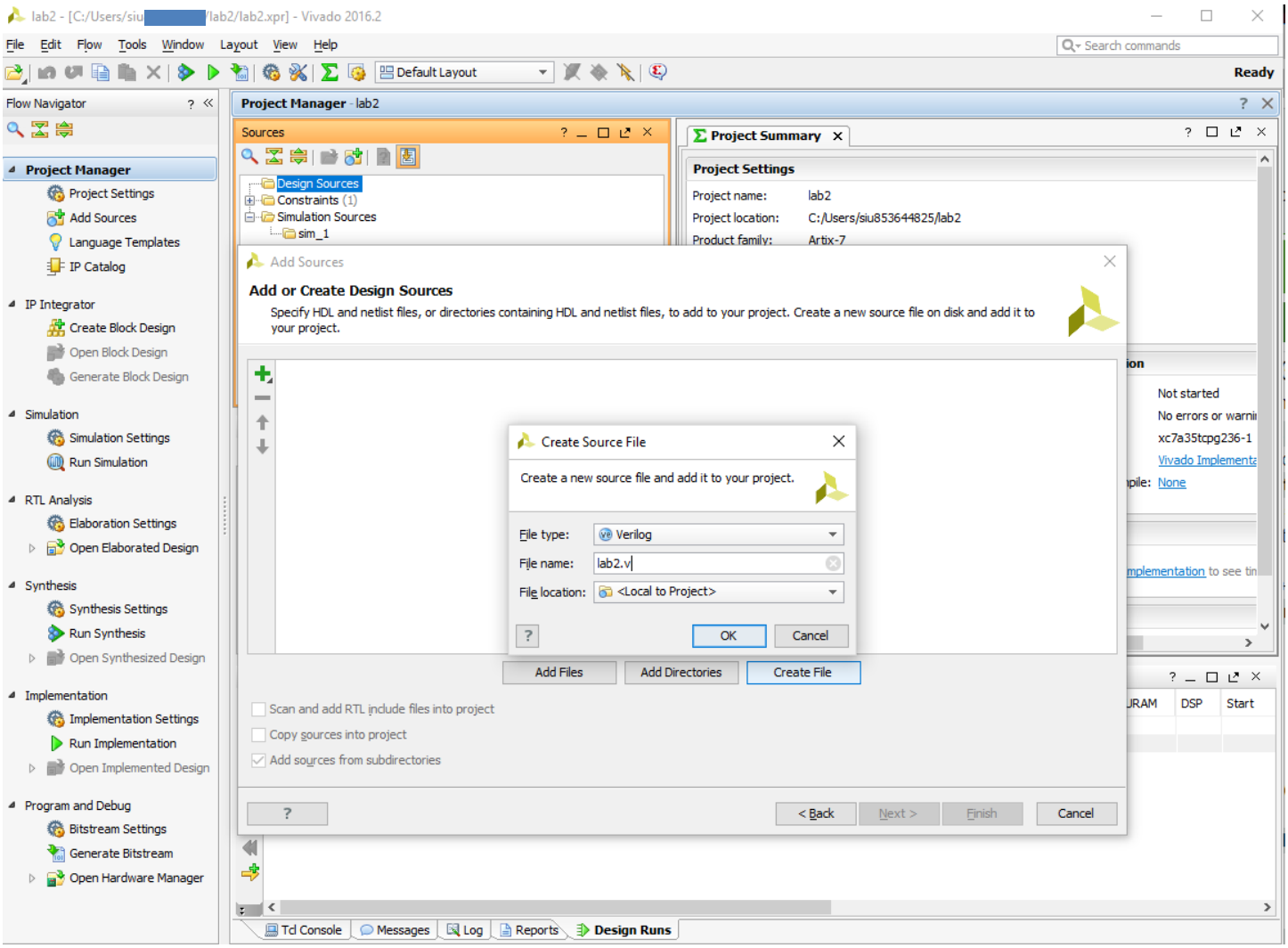

Fig. 8 Create a source file.

Configure the I/O signals for your source file created on last page. If you use an existing source file, skip this step. You can also define IO ports later for your own file by skipping here. **Note that SSG_EN is actually 3-bit signal, while it's one bit in Fig. 9. Modify it to be a 3-bit output signal, which is used to turn off 3 segment led displays.**

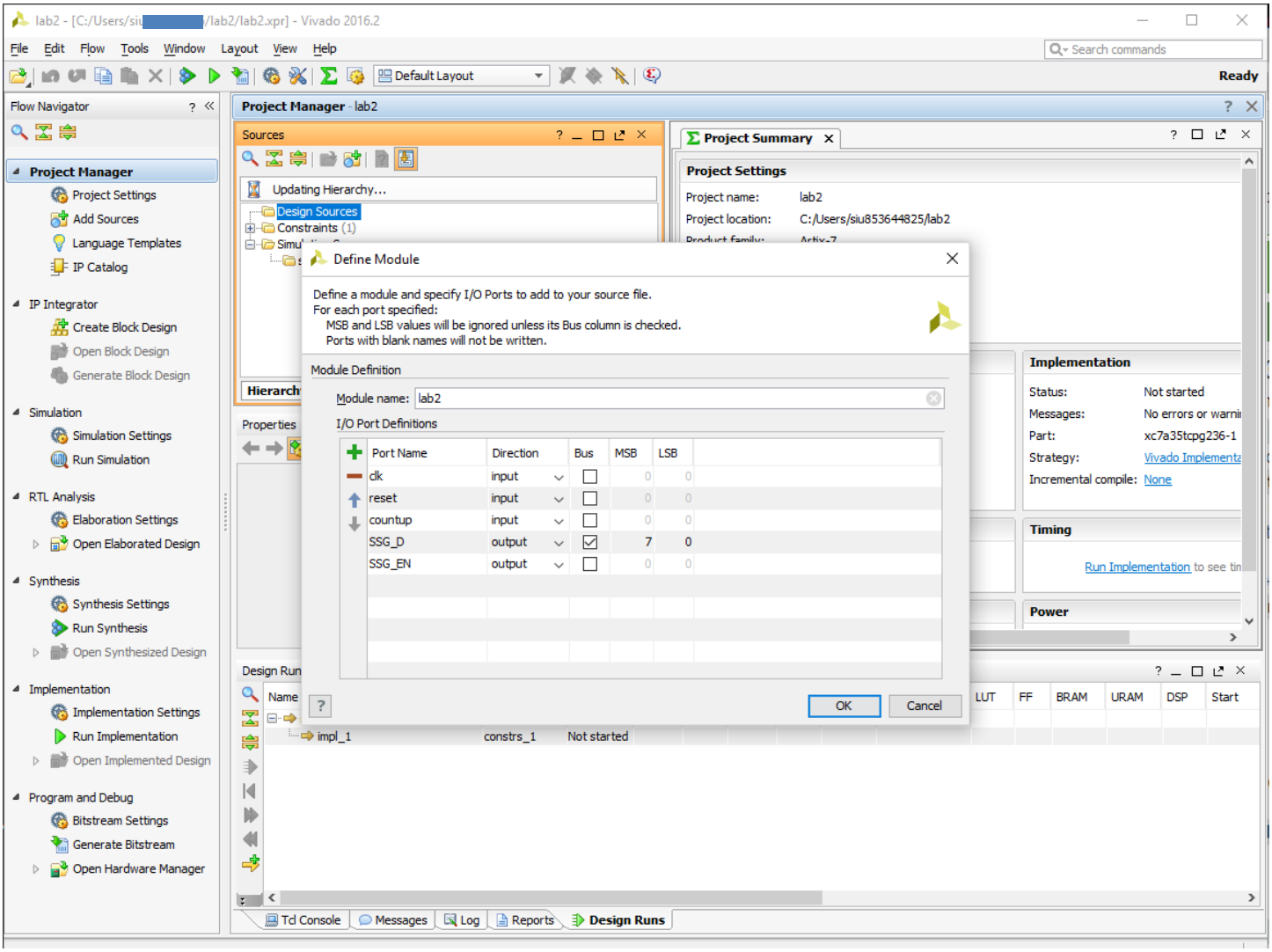

Fig. 9 I/O configuration page.

Behavioral Simulation:

Run “Behavioral Simulation” as shown in Fig. 10. A new simulation window will be opened.

Instead of writing a test bench, here we adopt the internal signal generator to provide stimulus for simulation. The stimulus is generated as shown in Fig. 11- Fig. 13. Waveforms can be find in Fig. 14. You can configure those input signals as you like, as long as the configuration is capable of finding functional errors.

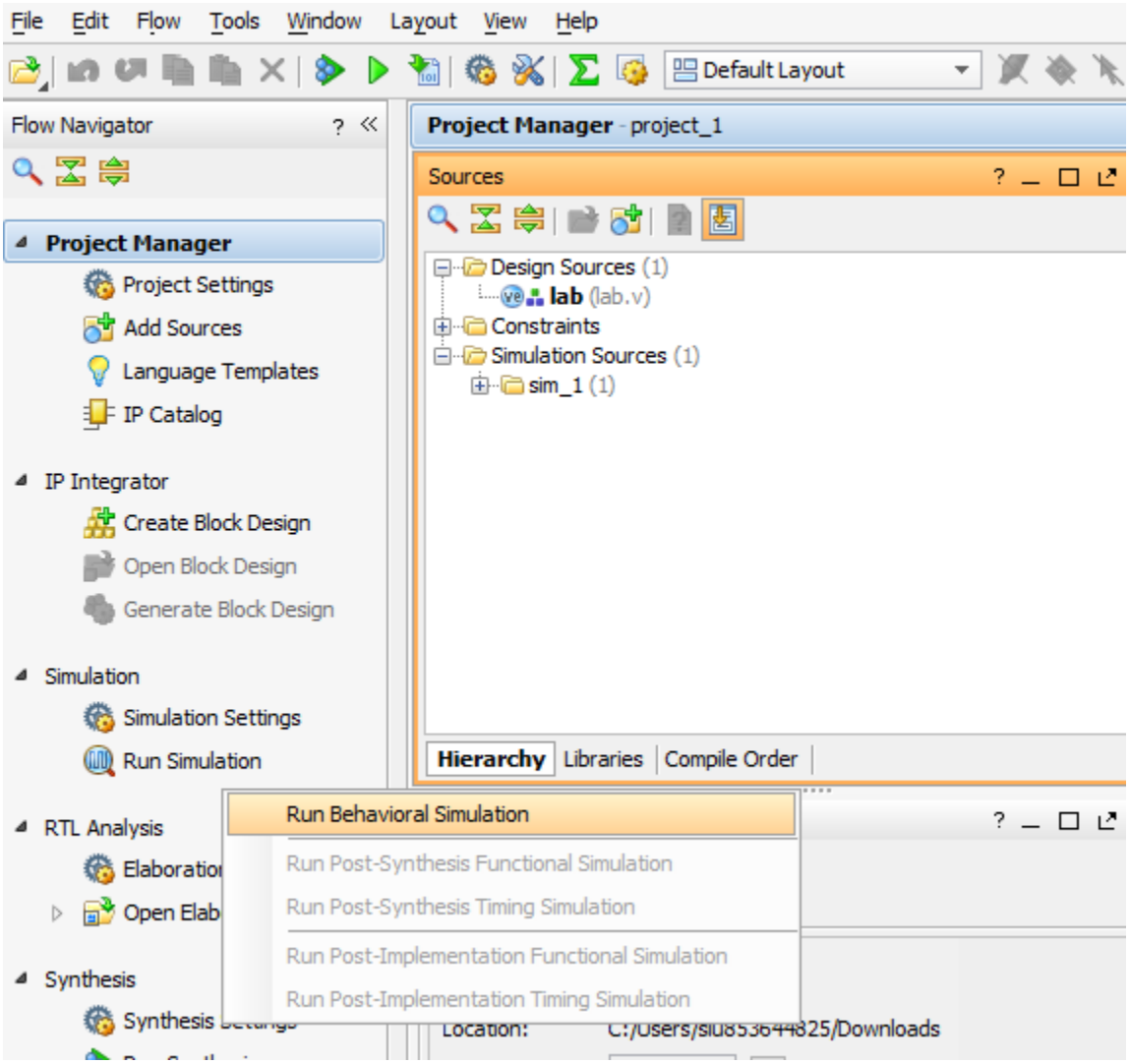

Fig. 10 Behavioral Simulation

Select “Force Constant” to generate a constant signal for “reset”, while using “Force Clock” to generate clock signal for “clk” and “countup”.

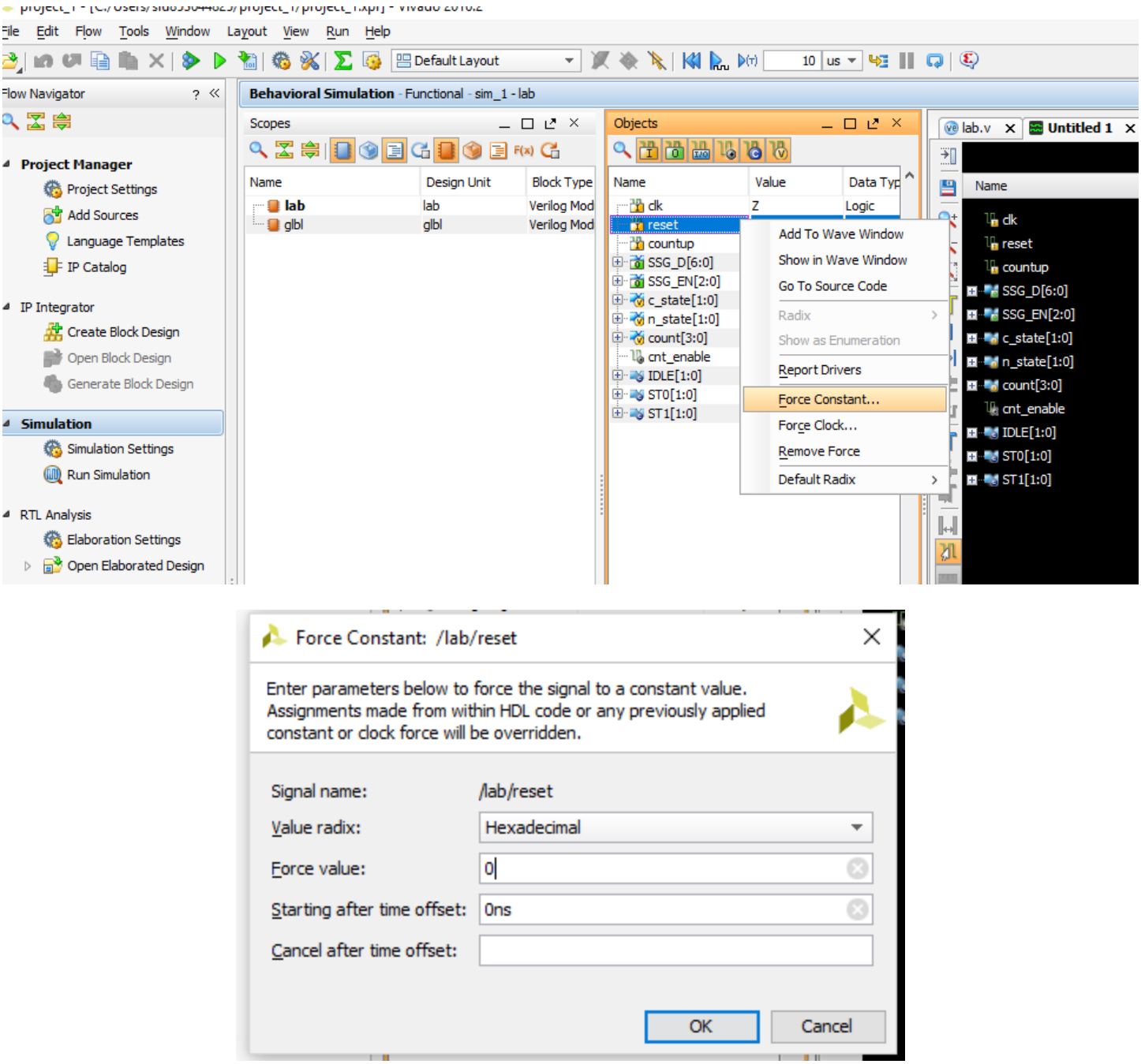

Fig. 11 Configure “reset” signal in simulation.

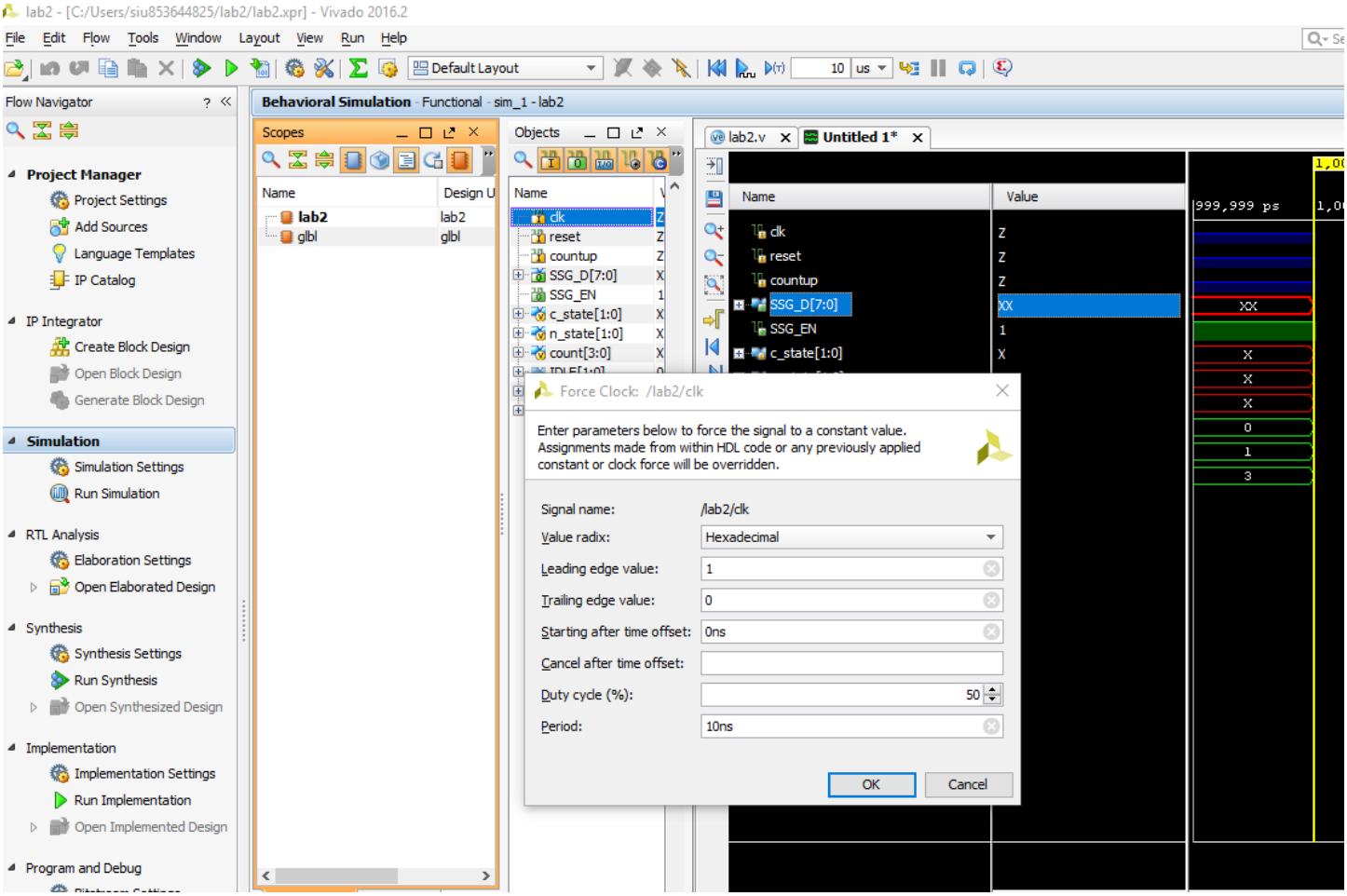

Fig. 12 Configure “clk” to be a 100 MHz clock.

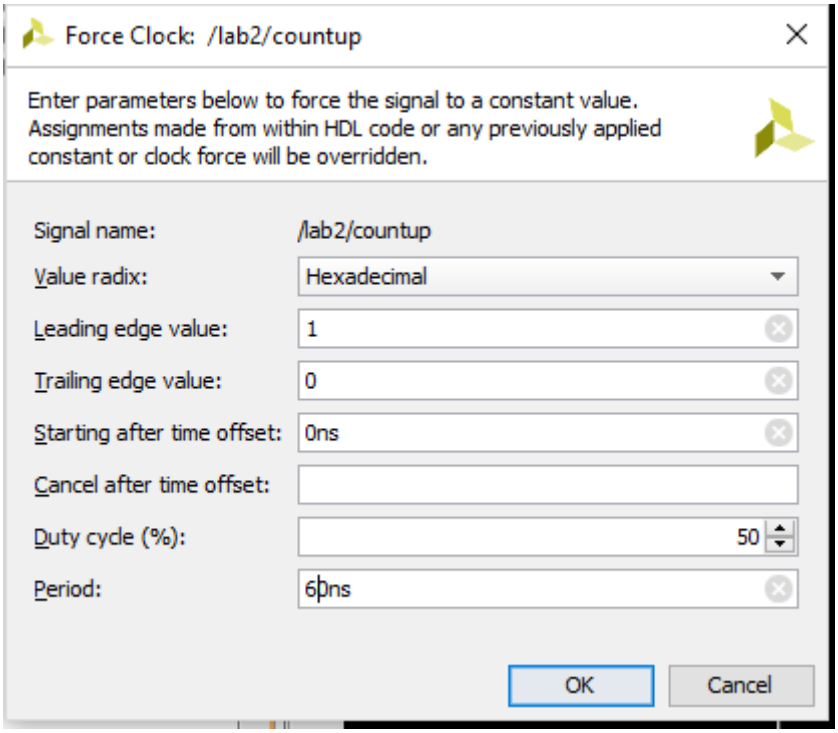

Fig. 13 Configure "countup" signal in simulation.

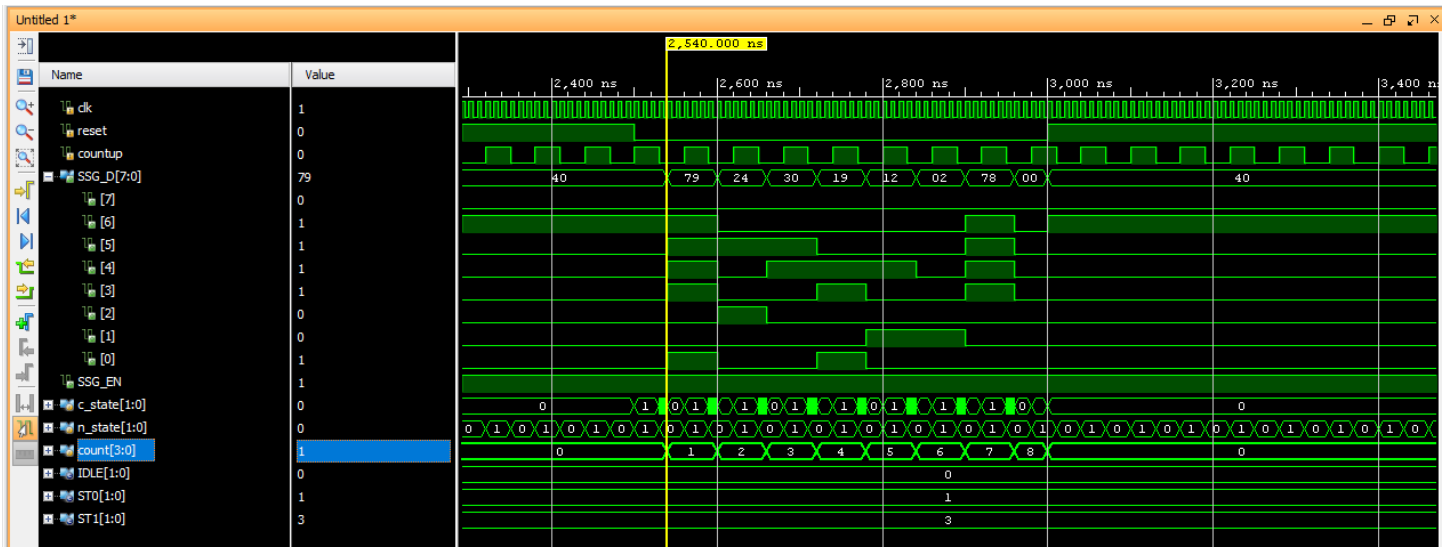

Fig. 14 Simulation waveforms.

Post-Synthesis Timing Simulation

There is some differences existing 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.

The setting of input signals are similar to behavior simulation. However, the resulting waveforms are a slightly different as shown in Fig. 16, where we can find signal edges are not aligned perfectly. This is caused by inherent cell delays along logic paths.

Moreover, Vivado also provides Post-Implementation Timing Simulation (PITS), which runs simulation after implementation that maps your design to the desired device. PITS will consider more timing information than PSTS. For example, logic line delay between two cells will also be considered. Interconnect delay is the dominant delay if the technology of fabrication is smaller than 90nm. In order to run PITS, you have to finish implementation first. The process of PITS is similar to PSTS and not introduced here.

Note: PSTS can only be activated after you have finished synthesis as shown Fig. 15.

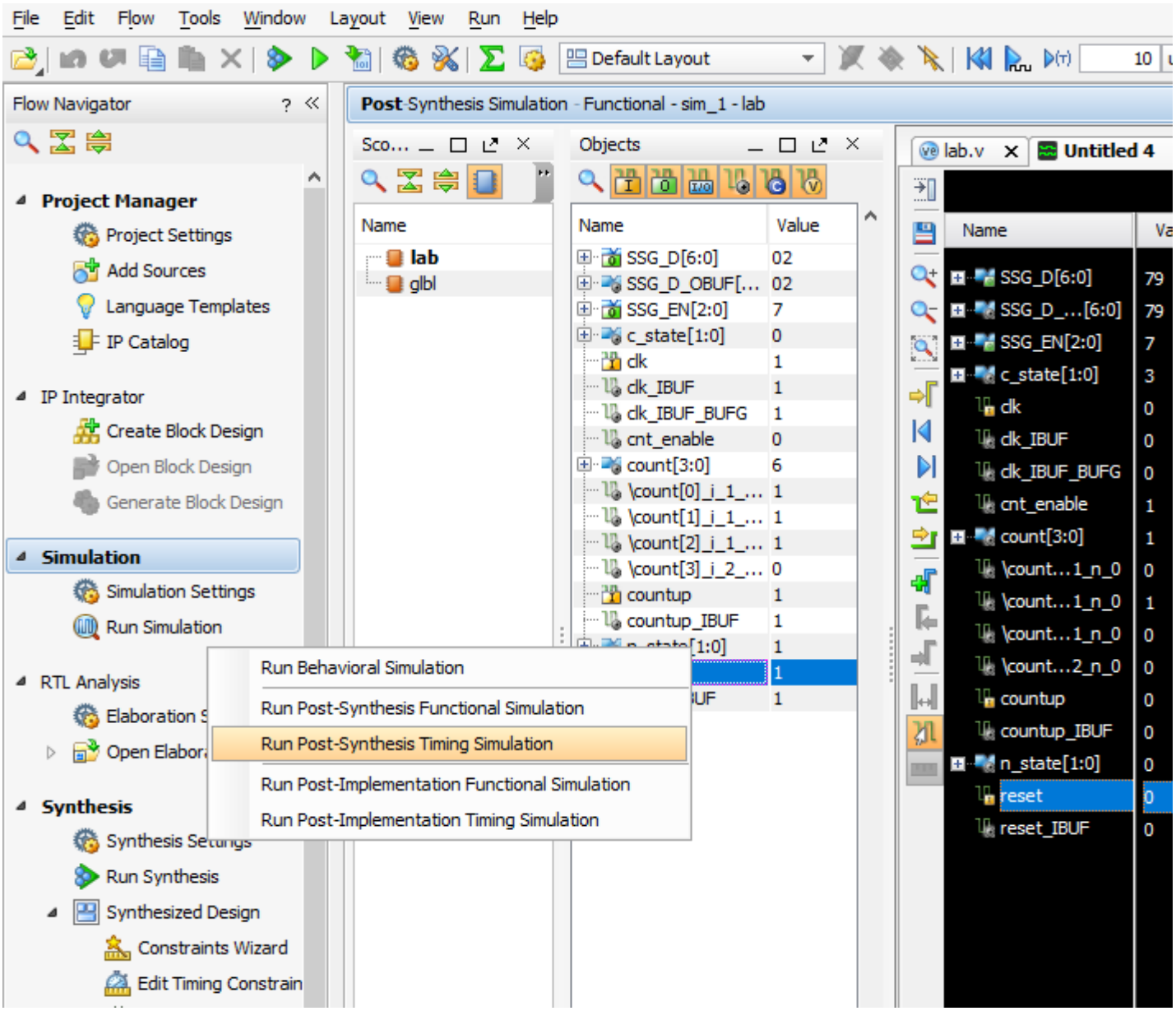

Fig. 15 Post-Synthesis Timing Simulation

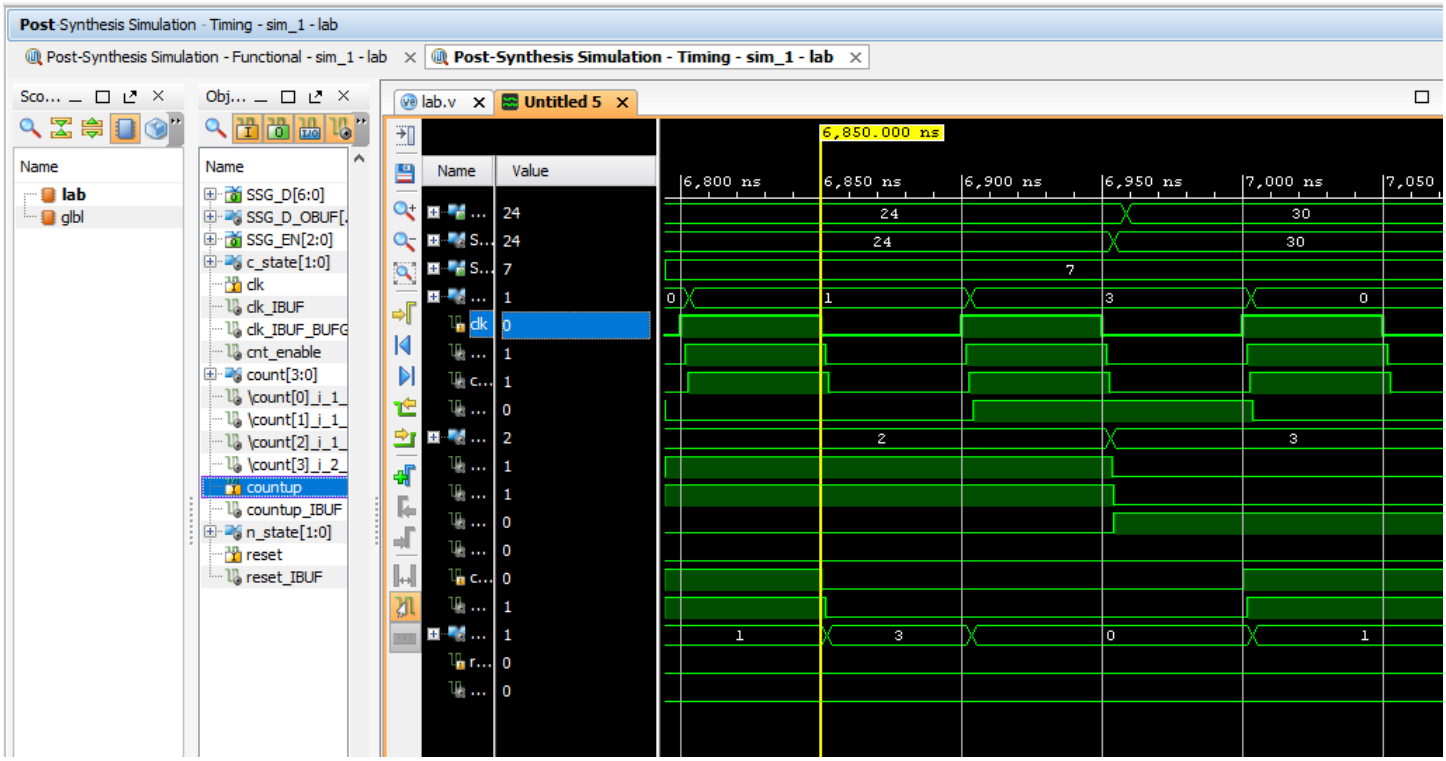

Fig. 16 PSTS waveforms

Synthesis and implementation:

In the flow navigator on the left side of Vivado, find “Synthesis” and click on “Run Synthesis”, as shown in Fig. 17.

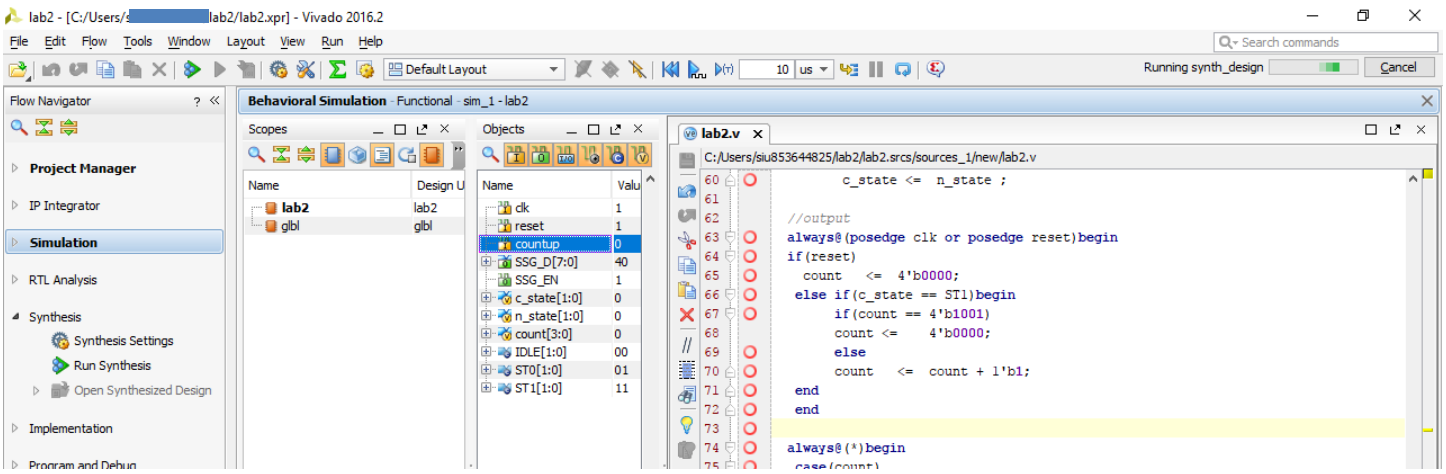

Fig. 17 Synthesize your design after simulation.

A window will be popping up as shown in Fig. 18 after successful synthesis. If you have some errors during synthesis, correct the errors and re-run synthesis. Select "Run Implementation" and click on "OK".

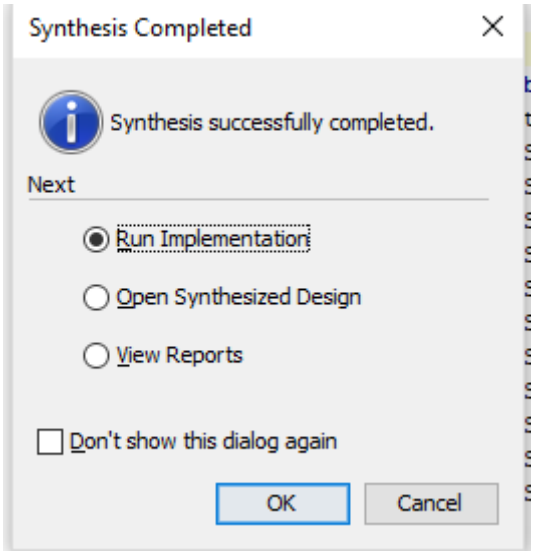

Fig. 18 Pop-up window after successful synthesis.

Select "Open Implemented Design" after successful implementation as shown in Fig. 19.

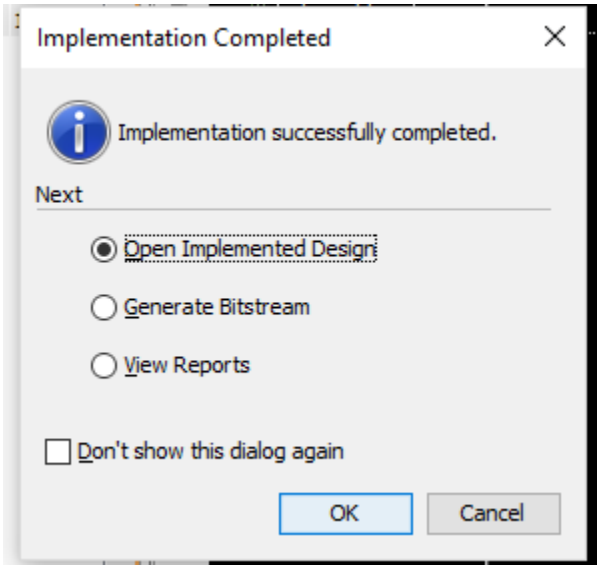

Fig. 19 Open implemented design.

Open the design after implementation as shown in Fig. 21.

Go to “Layout -> IO planning”, select “IO Planning” as shown in Fig. 20.

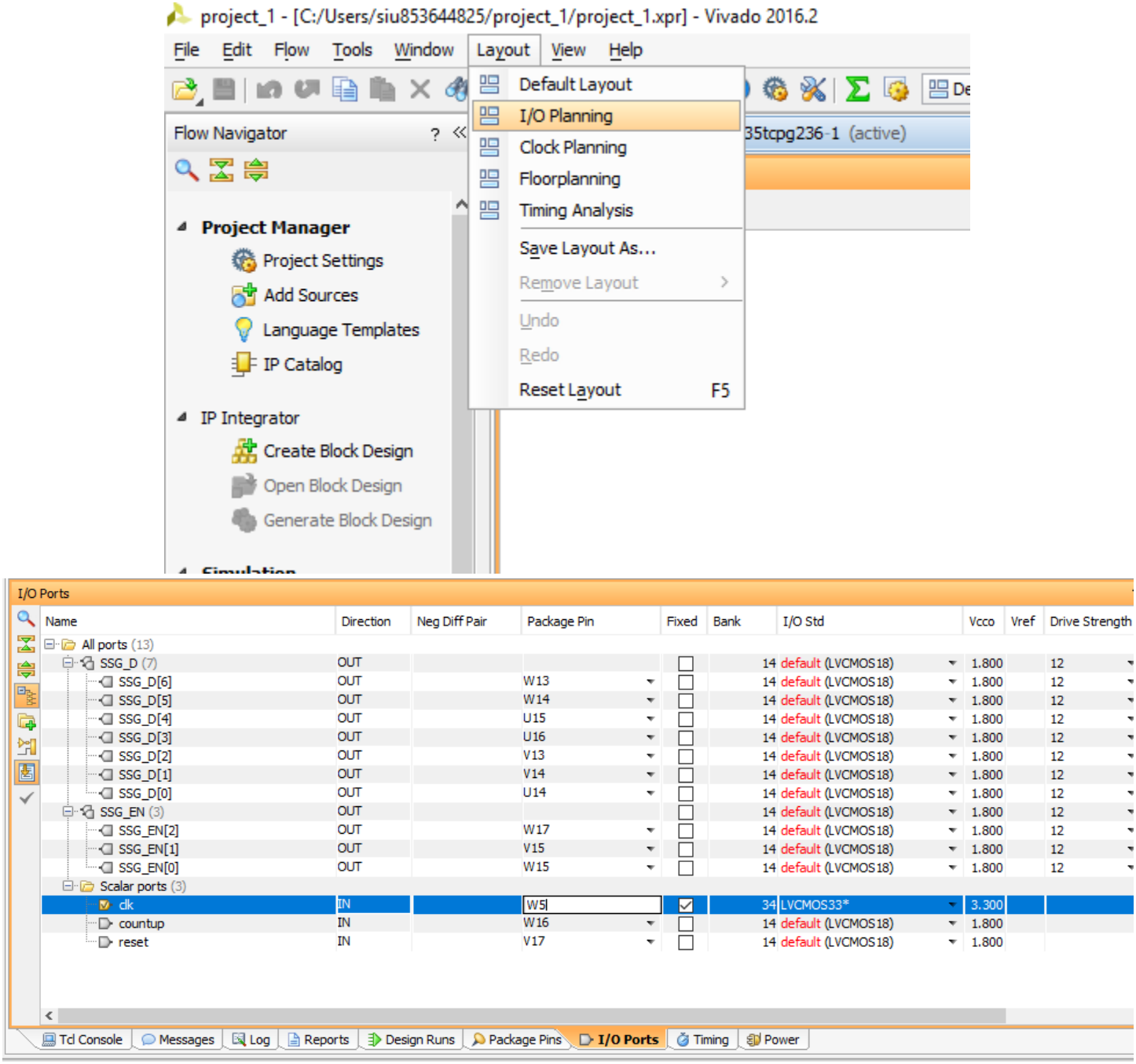

Fig. 20Manually IO assignment.

In this lab, we can also directly edit user constraint file instead of using the GUI for I/O assignment as shown in Fig. 22-25. Directly editing XDC file is much more convenient.

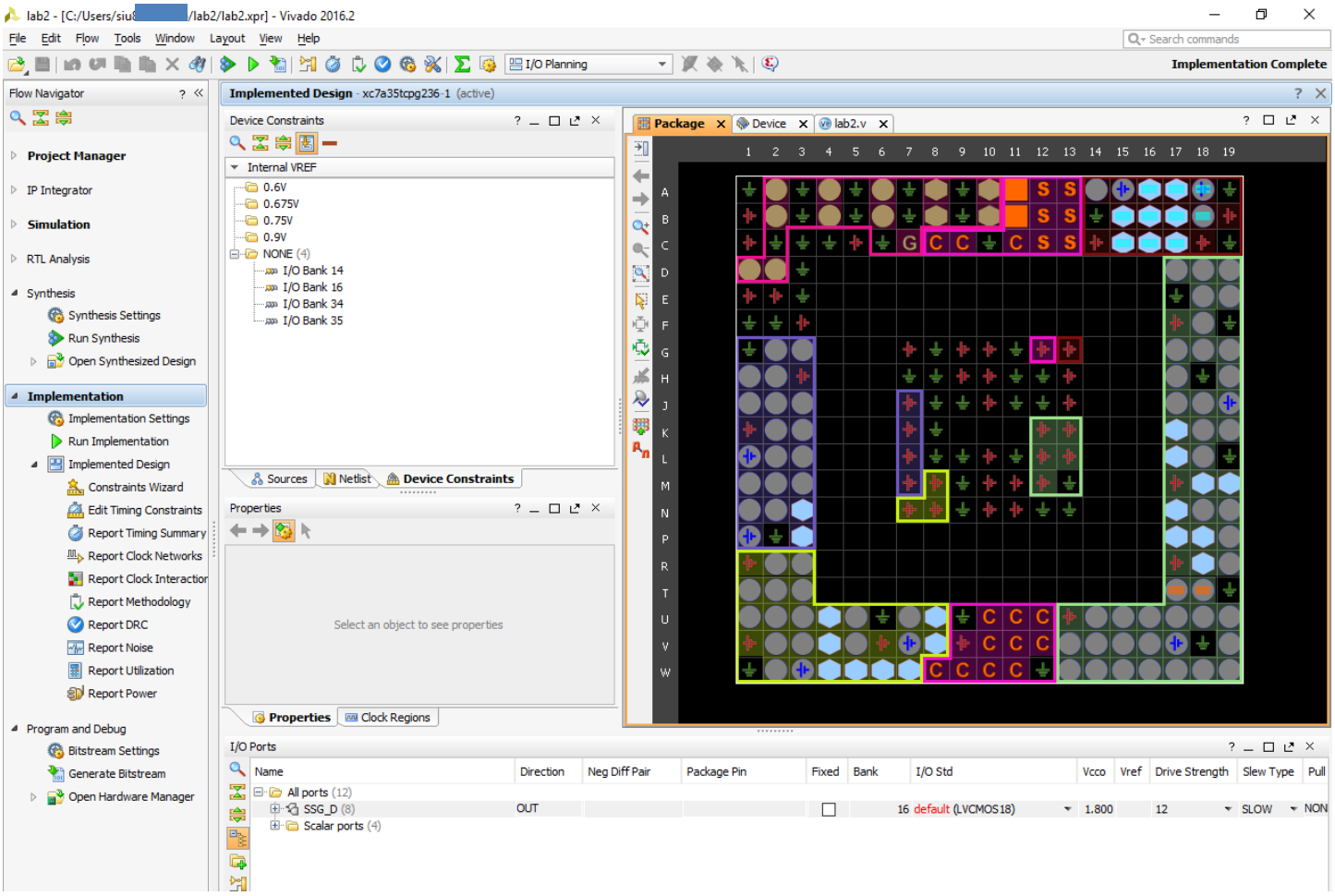

Fig. 21 I/O assignment.

To open “xdc” file, go to “Design Sources” and double click “user.xdc” under “Constraints”.

Note: XDC file has been loaded during project set-up in Fig. 5.

Use 100 MHz clock signal from pin “W5” and connect it with “clk”.

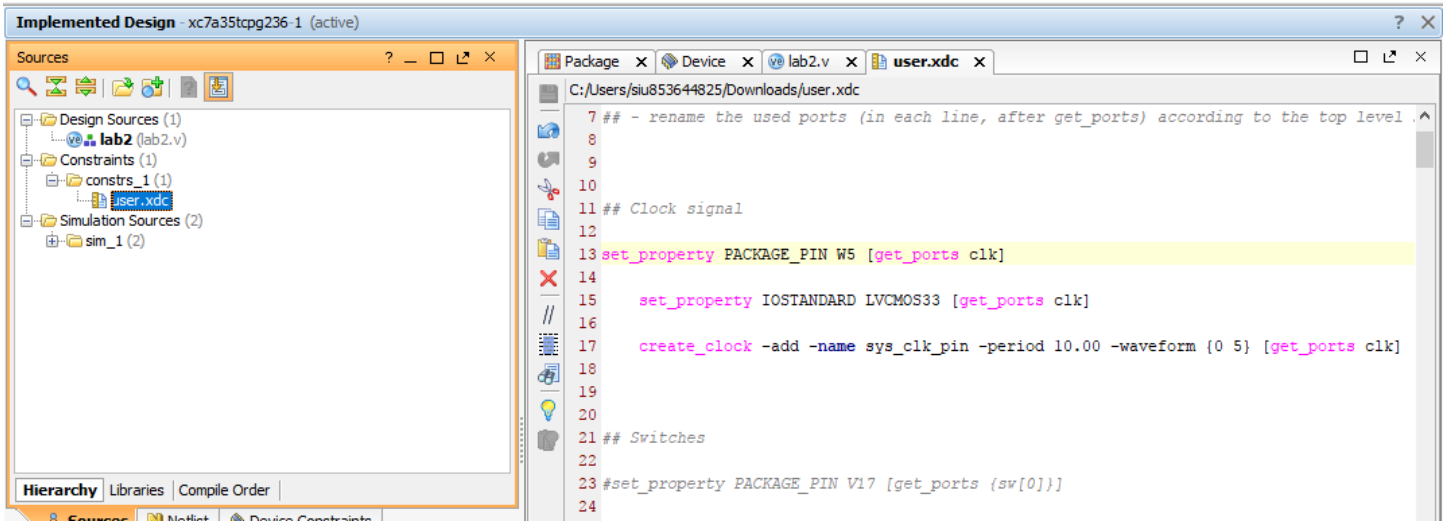

Fig. 22 Clock signal assignment.

Configure 7-segment LED display to the corresponding pins.

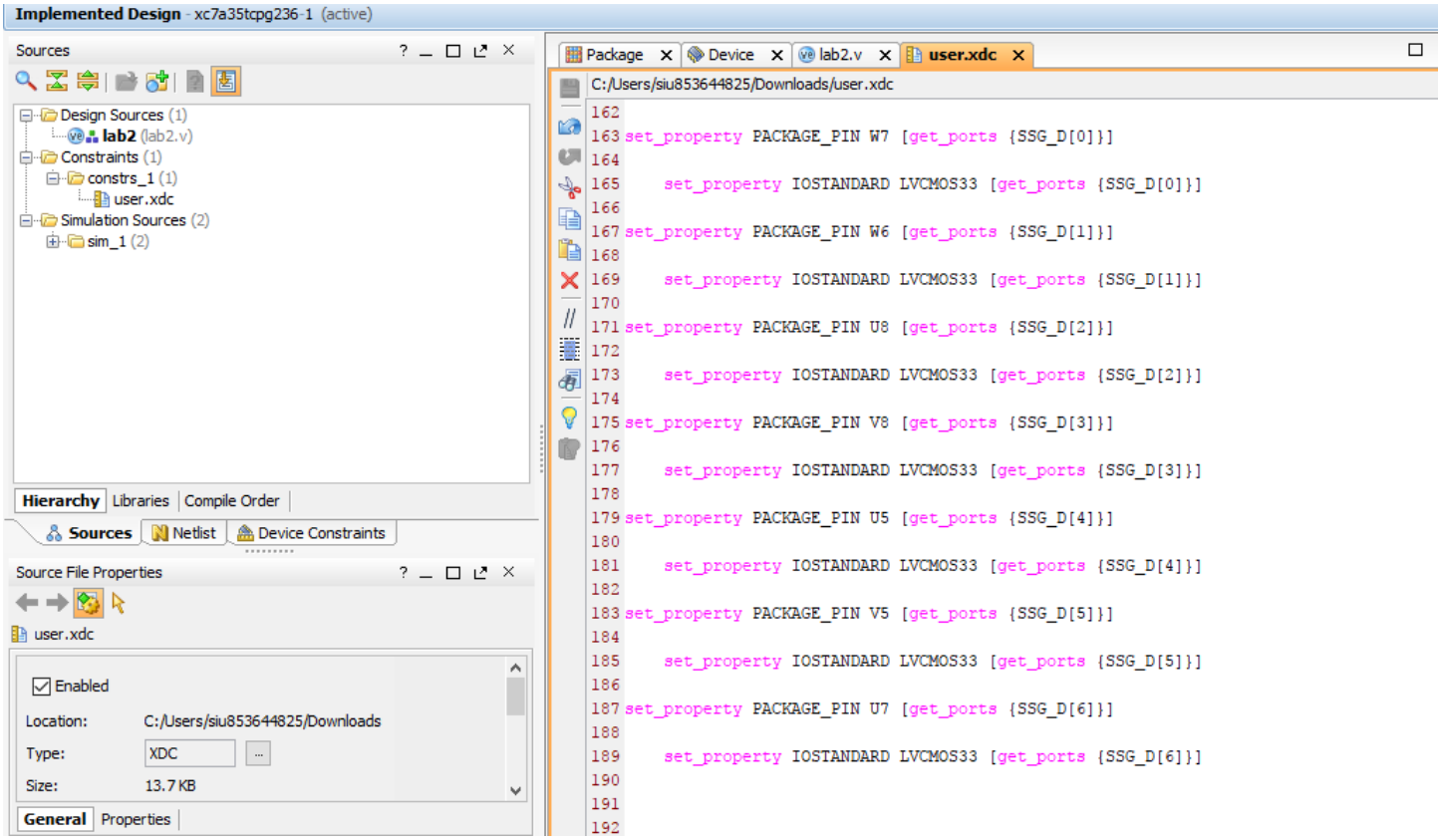

Fig. 23 Configure 7-segment LED display.

Connect three 7-segment display controlling signals. A “1” value will disable the corresponding segment display. In this lab, only one of four is used, thus other three will be disabled by assigning “1” to controlling signal.

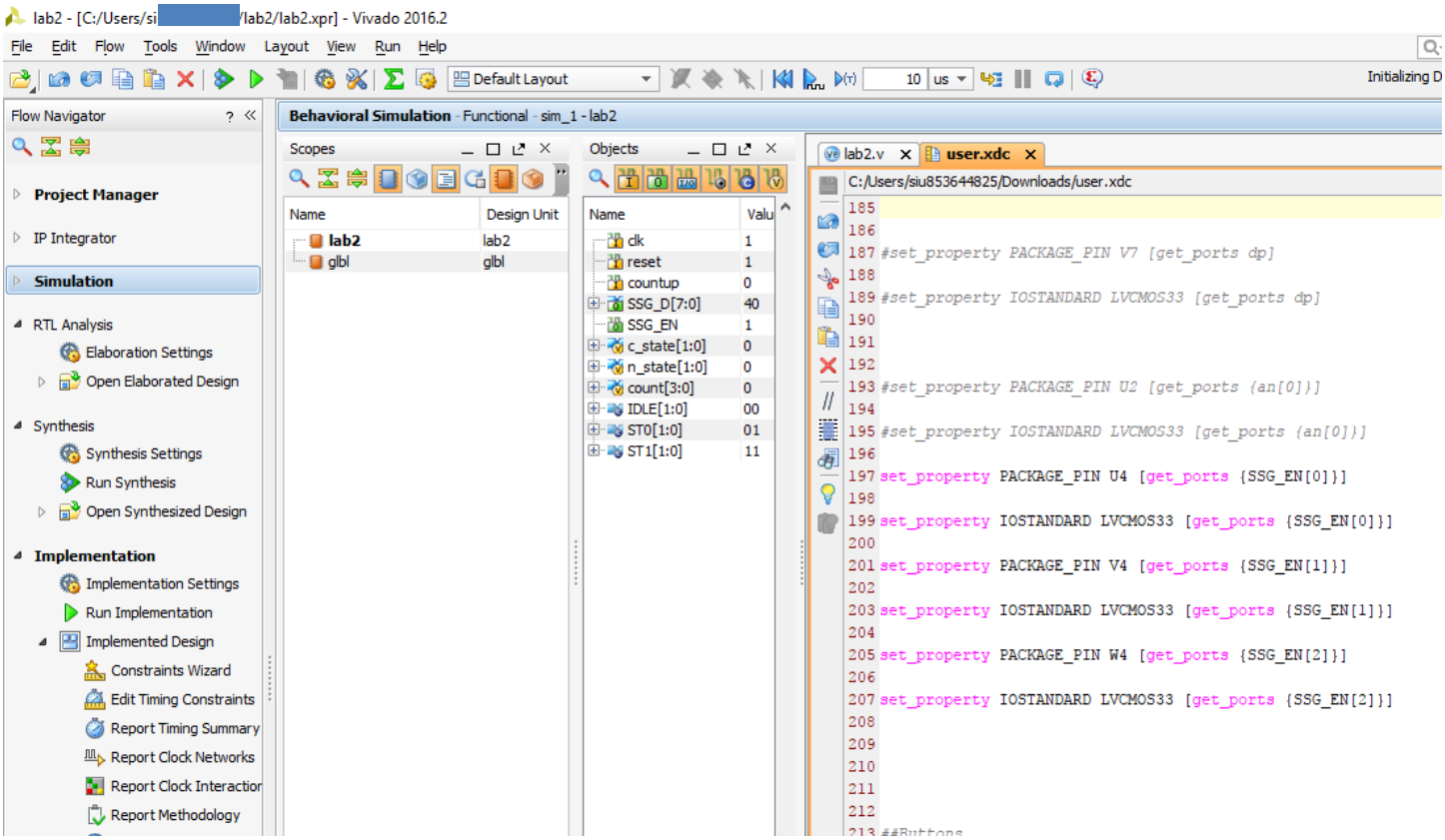

Fig. 24 Segment display control.

Configure “reset” and “countup” signal to buttons on the board.

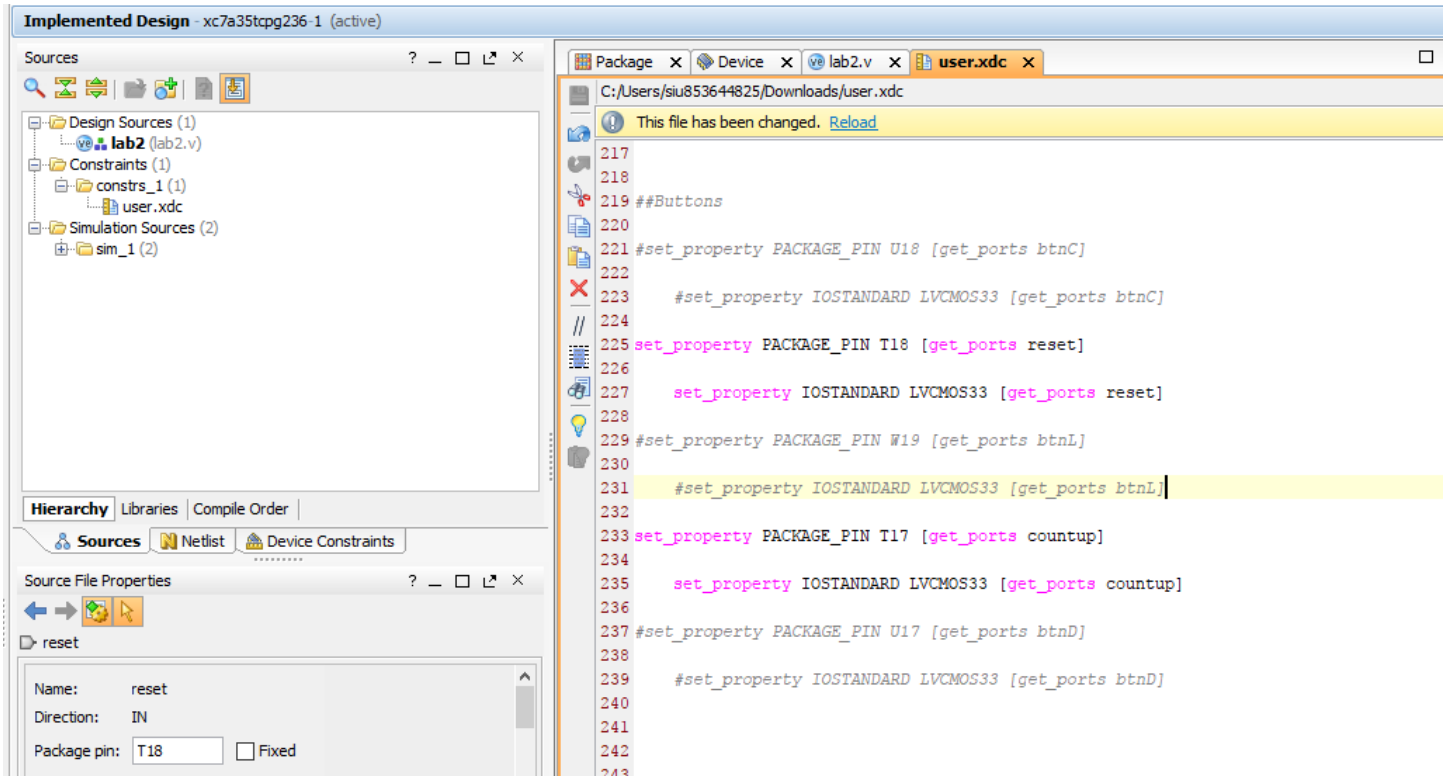

Fig. 25 Buttons for “reset” and “countup” signals.

Generate bin file and program FPGA:

Go to “Program and Debug” in Flow Navigator. Click on “Bitstream Settings” and select “bin_file” as the type for output as shown in Fig. 26 and Fig. 27.

Then click on “Generate Bitstream” as shown in Fig. 28.

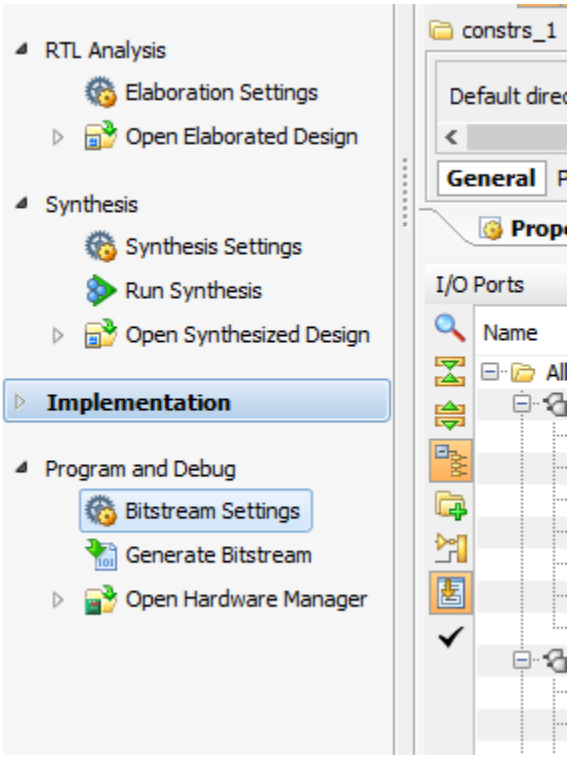

Fig. 26 Bitstream Settings

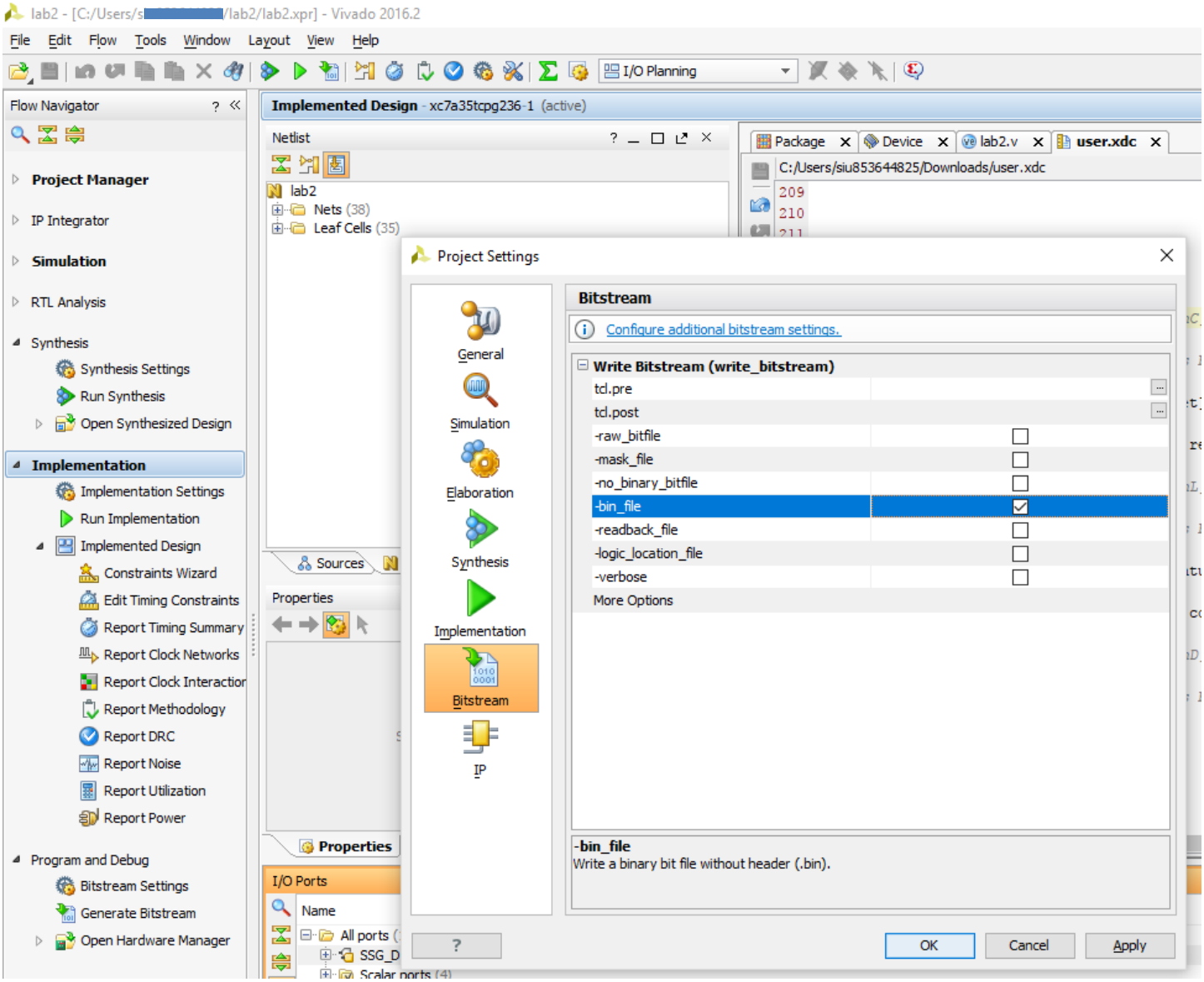

Fig. 27 Bitstream Settings.

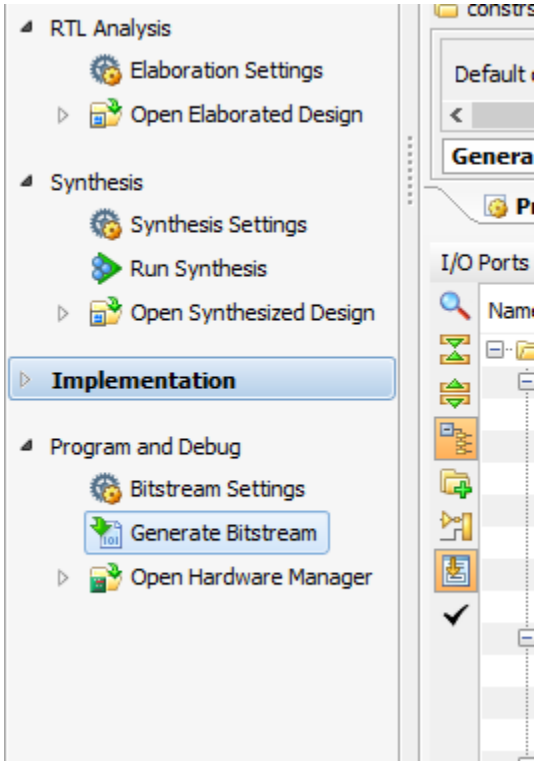

Fig. 28 Generate Bitstream

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

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

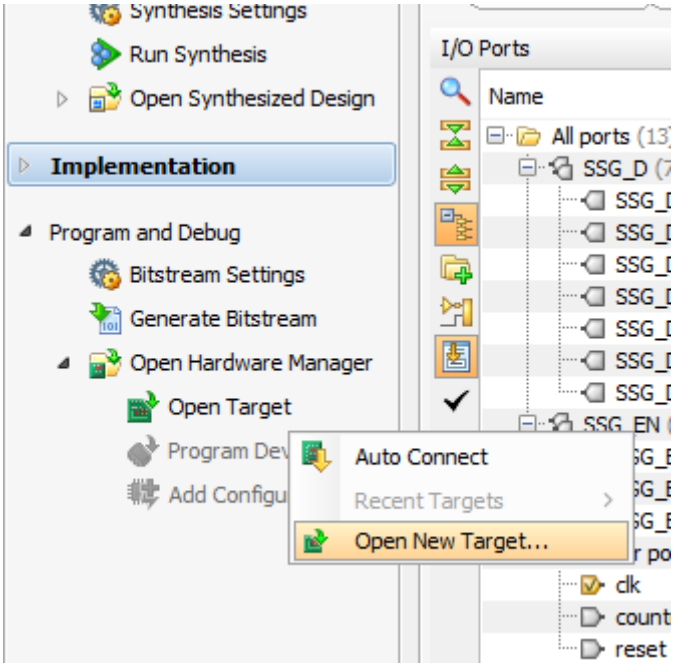

Fig. 29 Open New Target

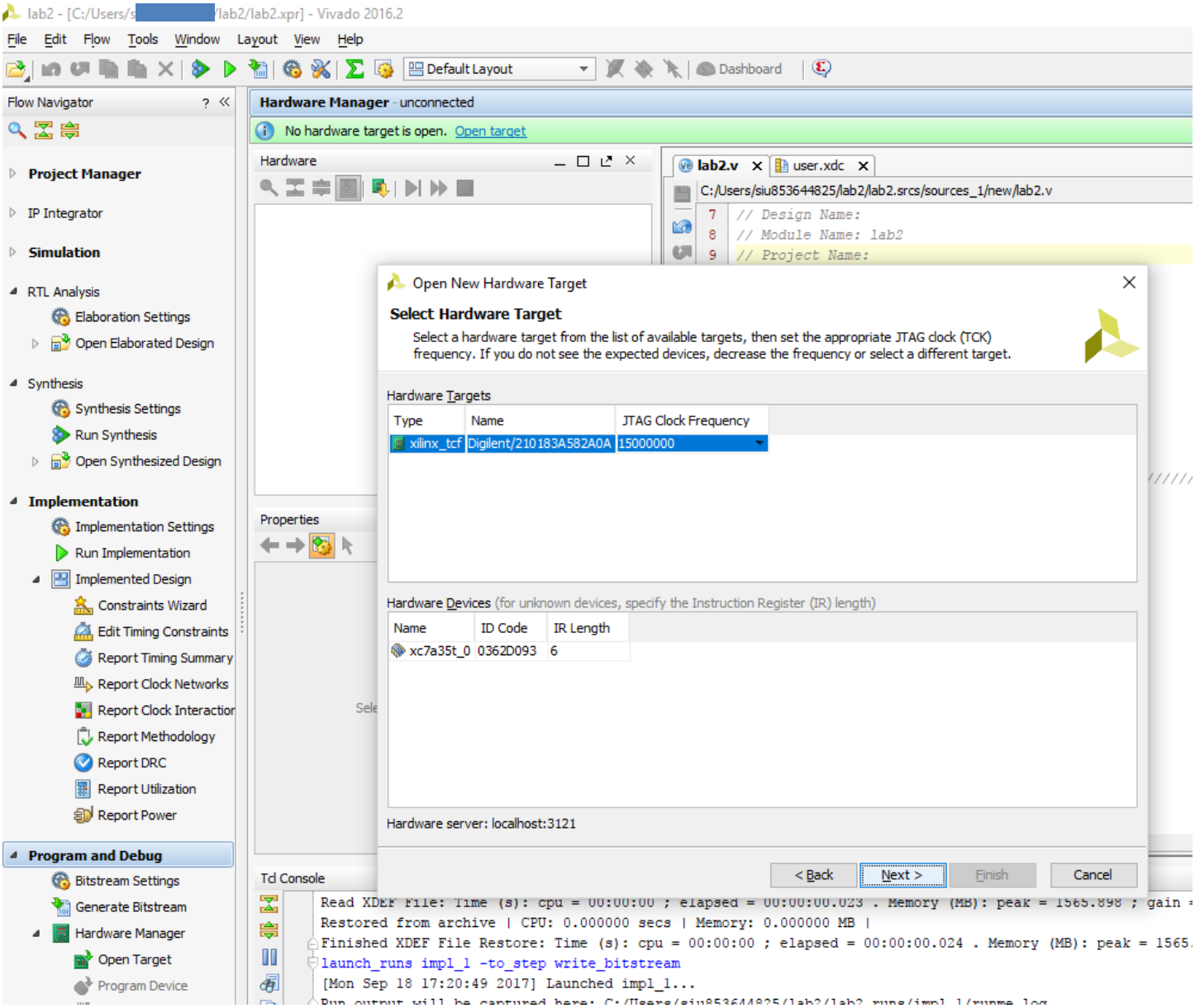

Fig. 30 Select the found device.

After successfully open 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.

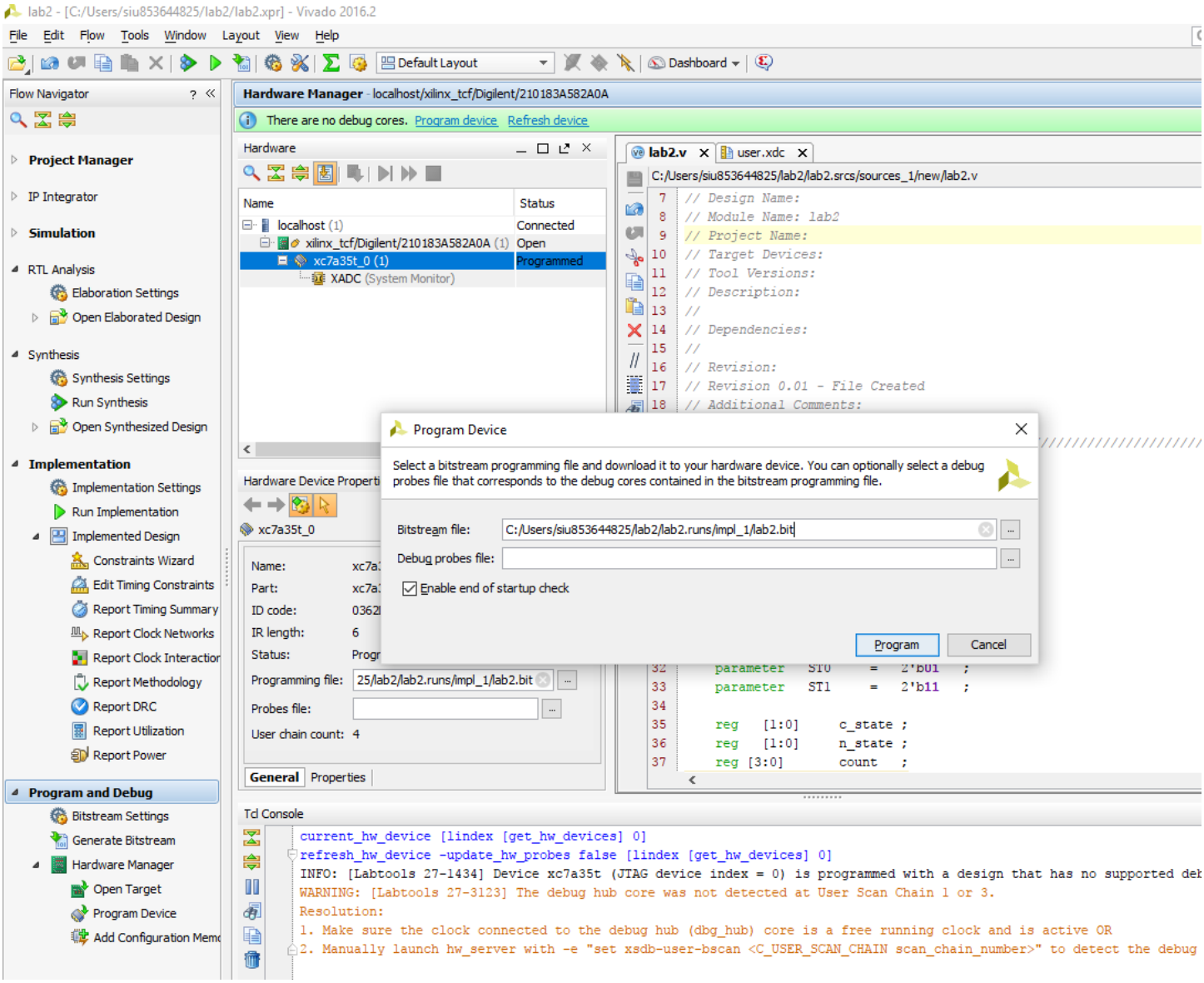

Fig. 31 Program your device.