

Tutorial on
FPGA Design Flow
based on
Xilinx ISE Webpack
and ModelSim

ver. 2.0

Prepared by Marcin Rogawski, Ekawat (Ice) Homsirikamol, and Dr. Kris Gaj

The example codes used in this tutorial can be obtained from

http://ece.gmu.edu/coursewebpages/ECE/ECE448/S09/experiments/448_lab3.htm.

The current version of the tutorial was tested using the following tools:

CAD Tool

- Xilinx ISE Webpack Version : 9.1

Synthesis Tool

- ISE&Webpack Synthesis&Implementation Version : 9.1
 - Synplicity Synplify Pro Version : 8.6

Implementation Tool

- Xilinx ISE/WebPack Version : 9.1

FPGA Board

- ## - Celoxica RC10

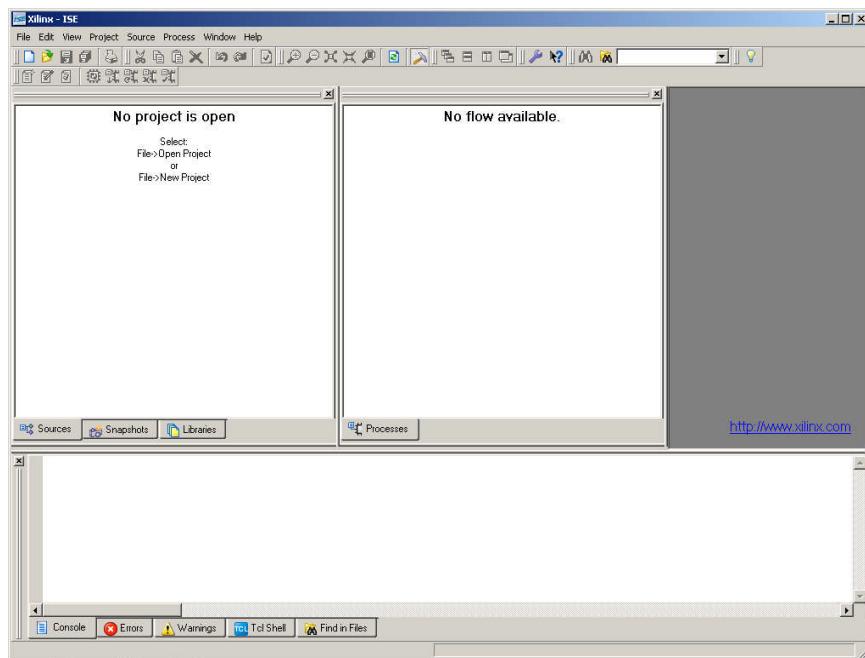
Table of Contents

1.	Project Settings	4
2.	Behavioral Simulation	7
3.	Pin Assignment	9
4.	Synthesis and Implementation	11
4.1	Synthesis	11
4.1.1	Synthesis with Xilinx XST	11
4.1.2	Synthesis with Synplify Pro	13
4.2	Translate	15
4.3	Post-Translate Simulation	16
4.4	Map	18
4.5	Place and Route	19
4.6	Post Place and Route Simulation	20
4.7	Implementation Reports	21
4.6	Bit Stream Generation	22
5.	Uploading Bitstream to FPGA Board	23

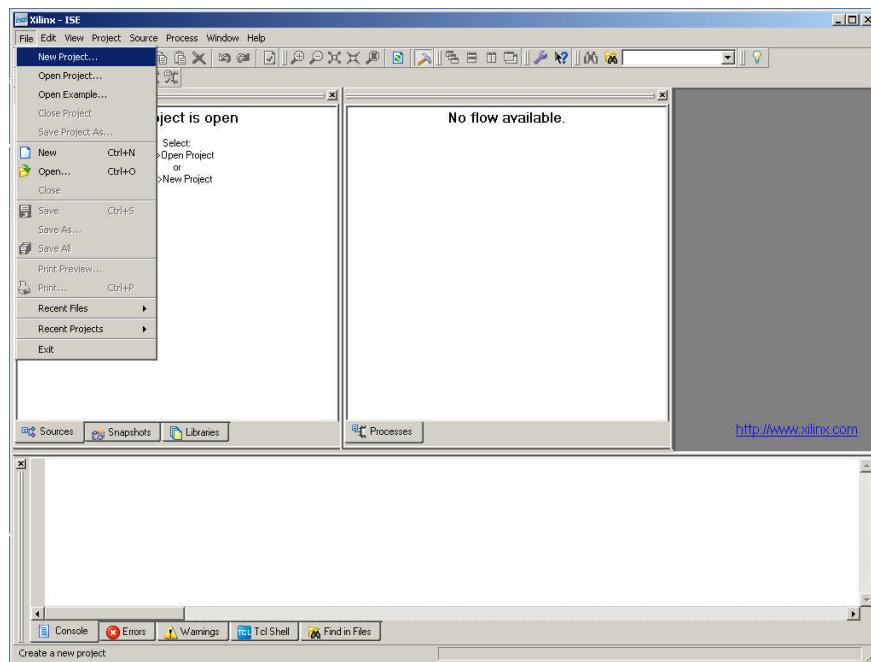
1. Project Settings

To Start Xilinx ISE go to:

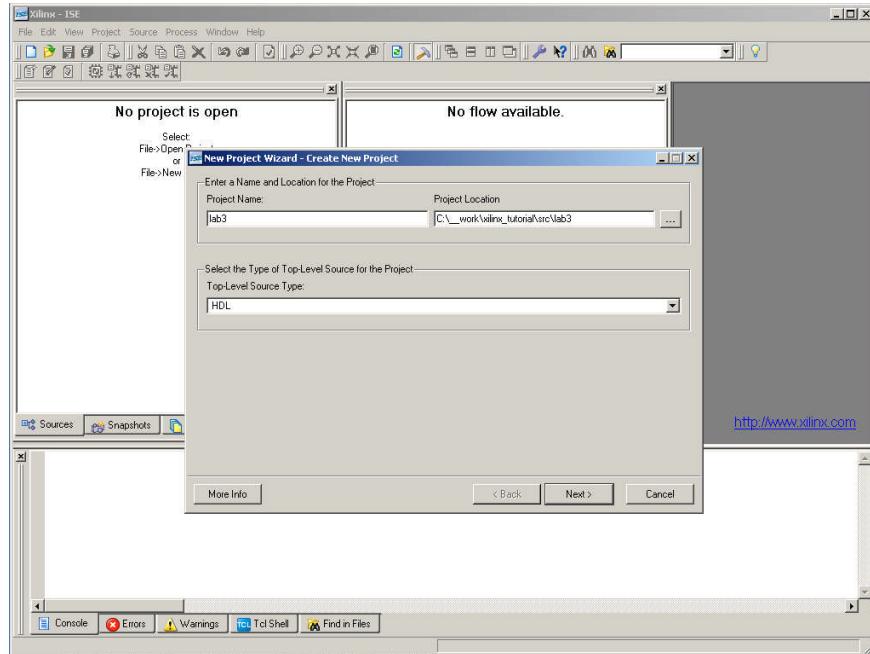
Start->All Programs-> VLSI Tools->Xilinx ISE 9.1i->Project Navigator



To start new Project go to: menu File->New Project

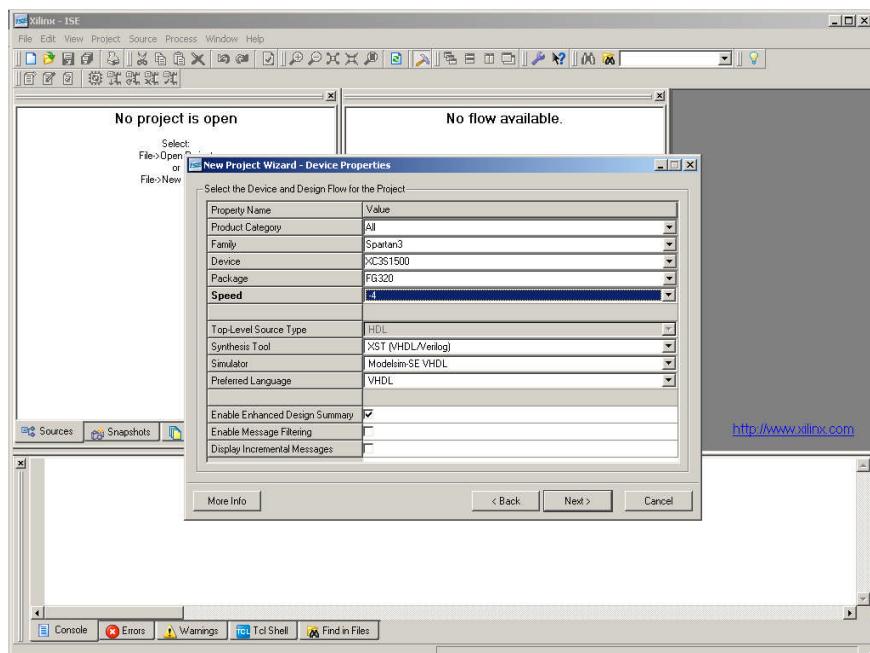


Specify Project Name and location for project files

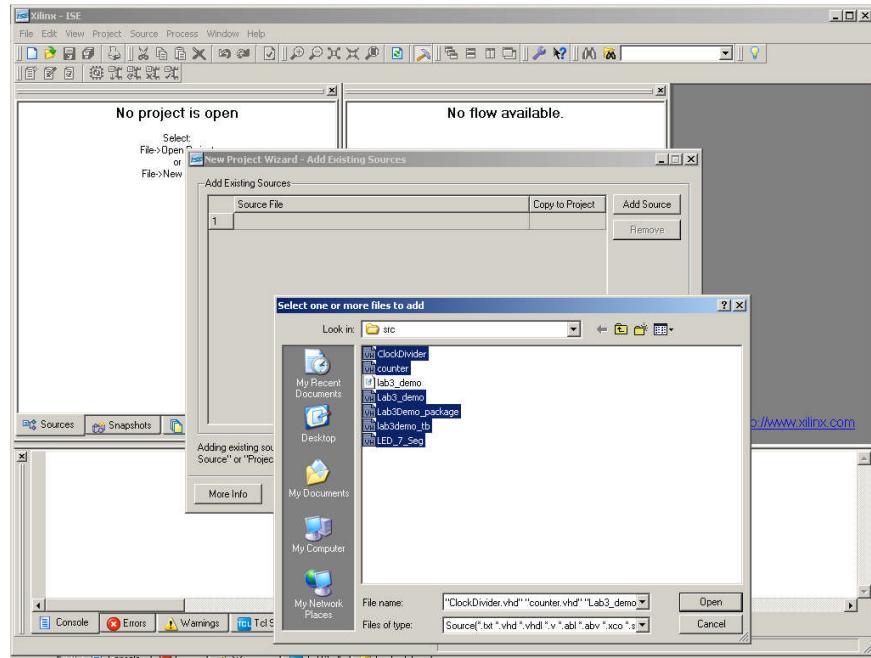


Click **Next** and specify

Family : Xilinx9x Spartan3
Device : 3s1500
Package : fg320
Speed Grade : -4
Simulator : ModelSim SE VHDL



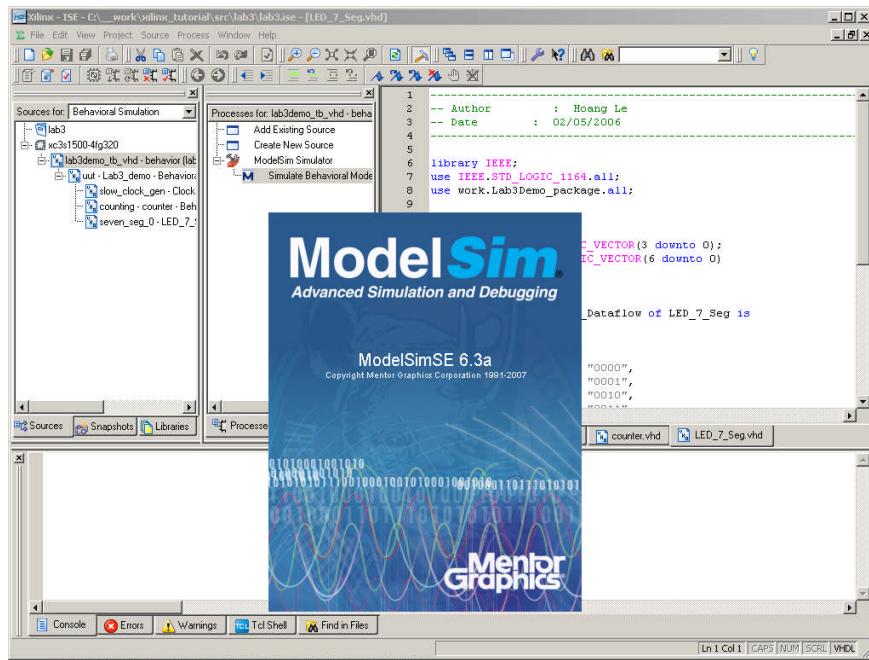
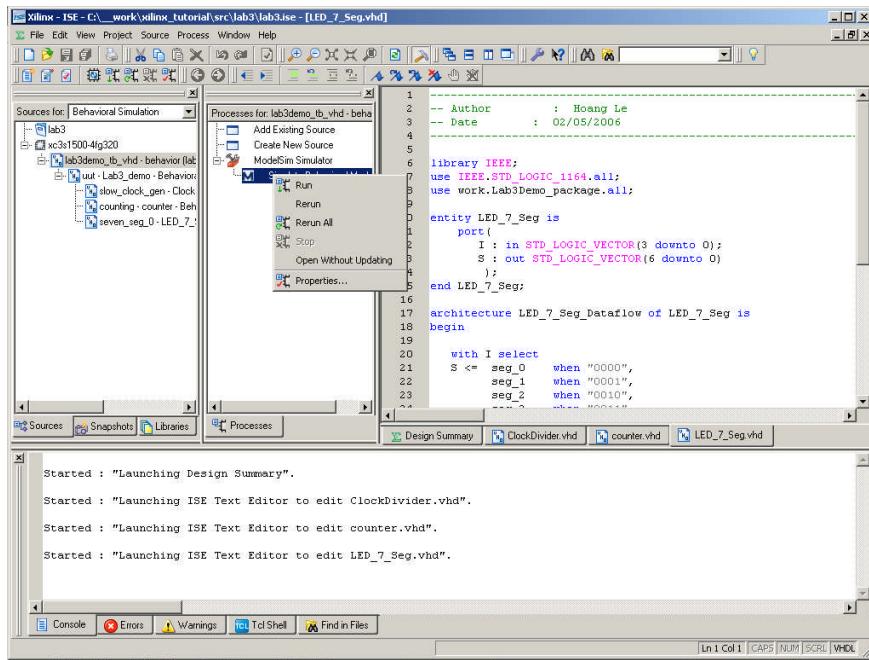
Click **Next** and then **Add Source**. Choose files for the project.



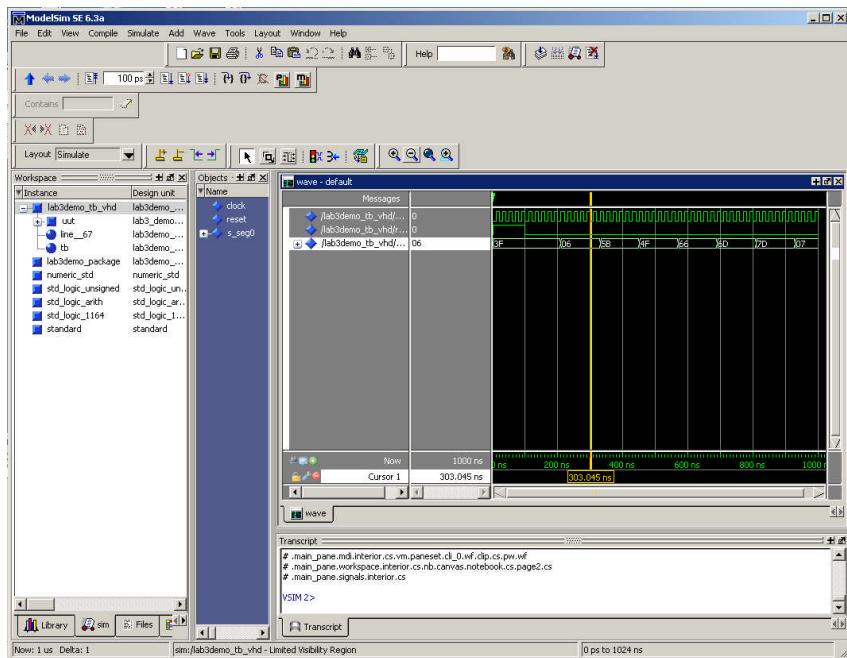
Then click **Open** and **Finish**. Your project has been defined.

2. Behavioral simulation

In the menu **Sources for** we choose **Behavioral Simulation**. The testbench for this design was set as a top level. In the menu **Processes for** choose **ModelSim Simulator**, right-click and choose **Run**.



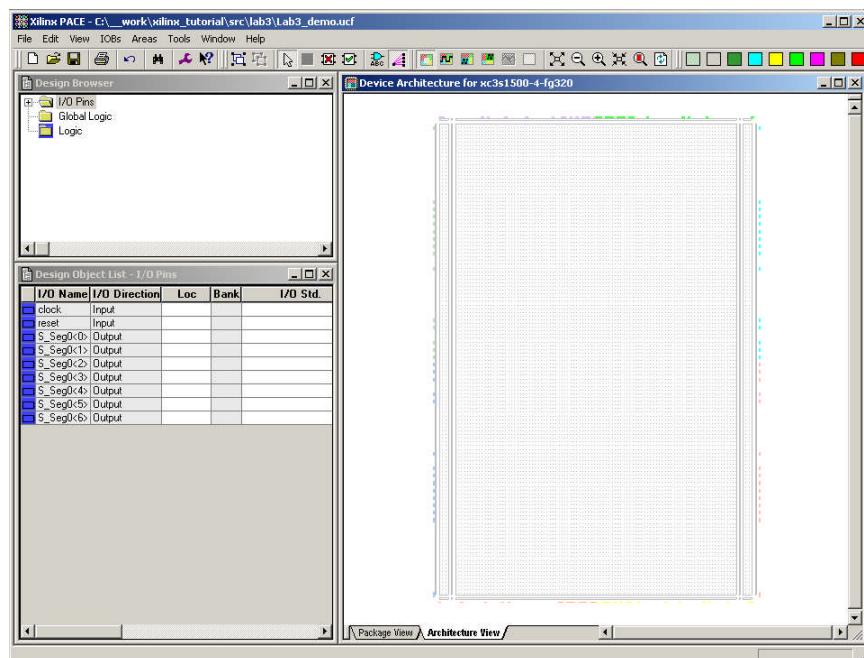
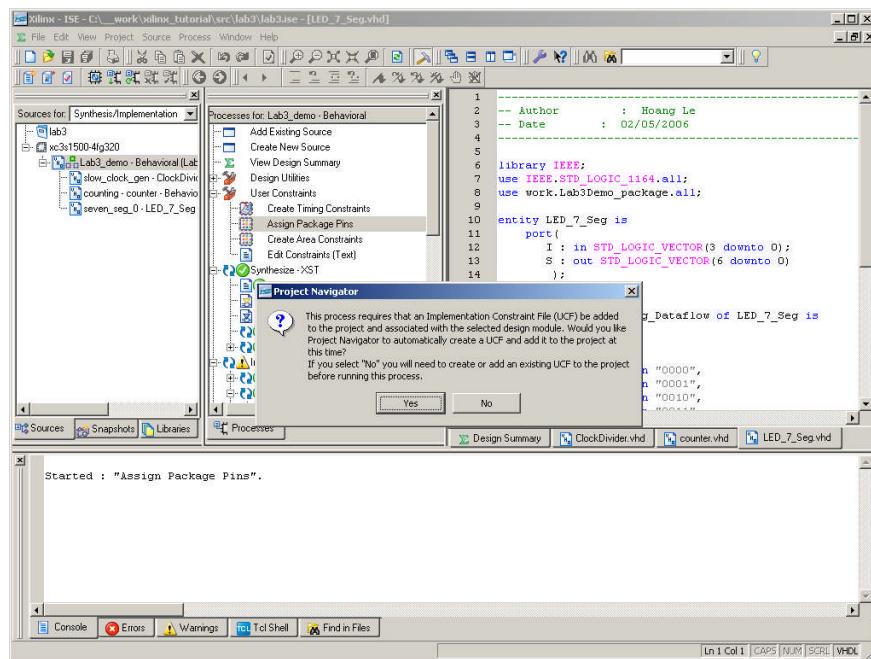
You will see a picture similar to the one above.

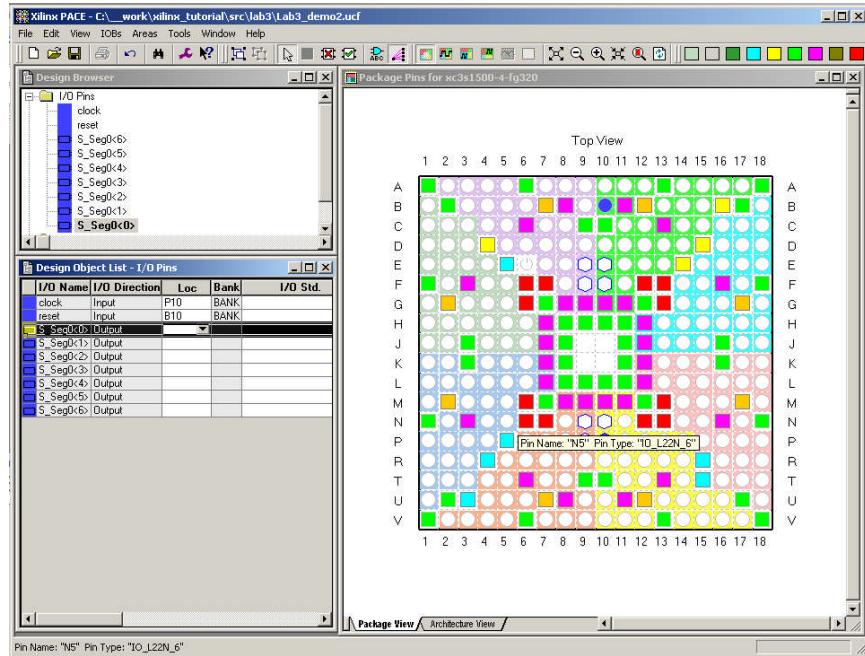


Finally you should be able to observe waveforms for your design.

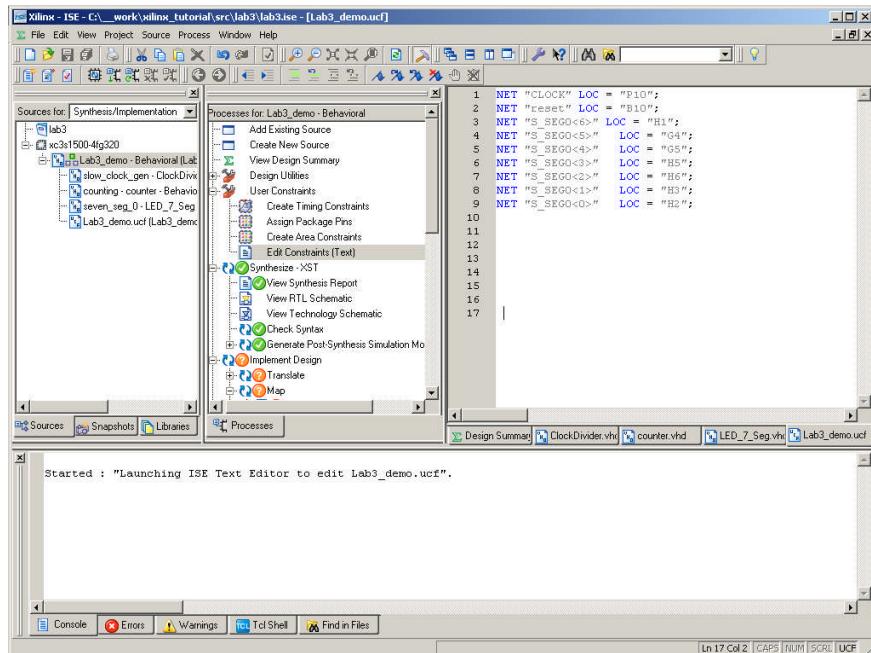
3. Pin Assignment

Go to menu **Sources for** and change this option to **Synthesis/Implementation**. Click on **Assign Package Pins**. The User Constraint File (UCF) will be created if you decide to assign design port names to the physical pins. Columns I/O Names and I/O Descriptions represent ports from your design. The Loc column should be used to input the location of the corresponding pin in the FPGA device.





We can specify Pin Assignments by going to **User Constraints** menu. We can assign design port names to the physical pins of a chosen device by option **Edit Constraints (Text)**. Keyword **NET** is for a port name assignment and keyword **LOC** is for a physical pin assignment.

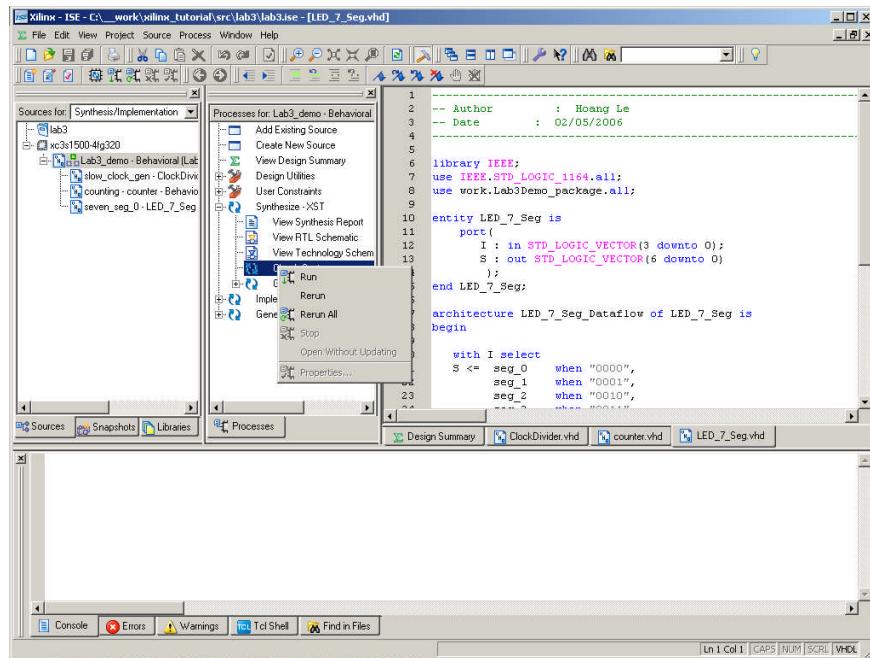


4. Synthesis and Implementation

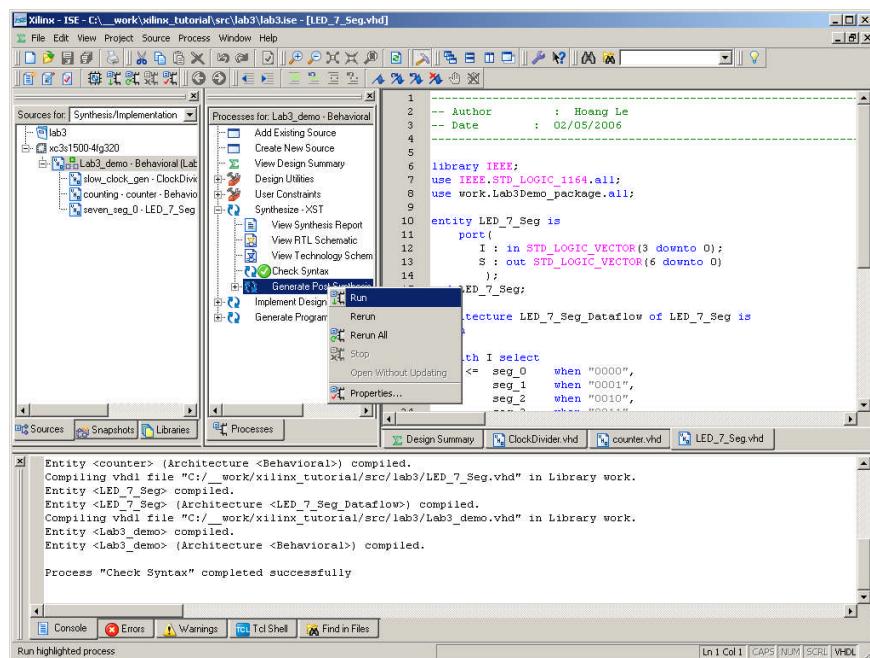
4.1 Synthesis

4.1.1 Synthesis with Xilinx XST

Go to the menu **Sources for** and change this option to **Synthesis/Implementation**. Click **Check Syntax** to check if vhdl sources are properly coded.



When you right-click on **Generate Post-Synthesis Simulation Model** then pop-up menu appears. Choose **Run** to start synthesis (There is no option to simulate post-synthesis netlist for Xilinx ISE WebPack).



```

-- Author : Hoang Le
-- Date : 02/05/2006

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use work.Lab3Demo_package.all;

entity LED_7_Seg is
    port(
        I : in STD_LOGIC_VECTOR(3 downto 0);
        S : out STD_LOGIC_VECTOR(6 downto 0)
    );
end LED_7_Seg;

architecture LED_7_Seg_Dataflow of LED_7_Seg is
begin
    with I select
        S <= seg_0 when "0000",
        seg_1 when "0001",
        seg_2 when "0010",
        seg_3 when "0011"
end architecture;

```

Synthesis and generation of Post-Synthesis Simulation Model were completed successfully and we can start Implementation part.

```

Process "Synthesize" completed successfully

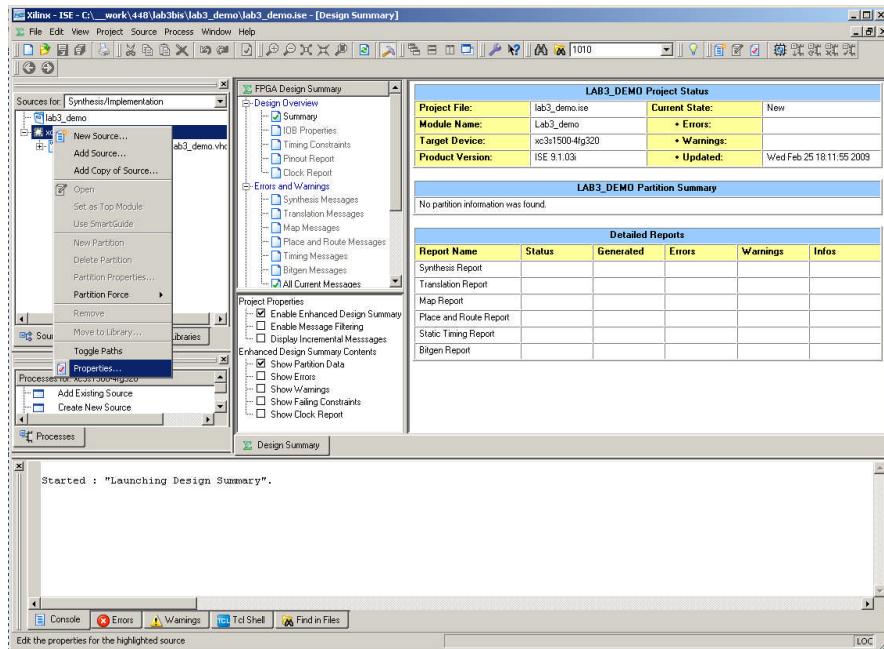
Started : "Generate Post-Synthesis Simulation Model".
INFO:NetListWriters:635 - The generated VHDL netlist contains Xilinx UNISIM
simulation primitives and has to be used with UNISIM library for correct
compilation and simulation.

Process "Generate Post-Synthesis Simulation Model" completed successfully

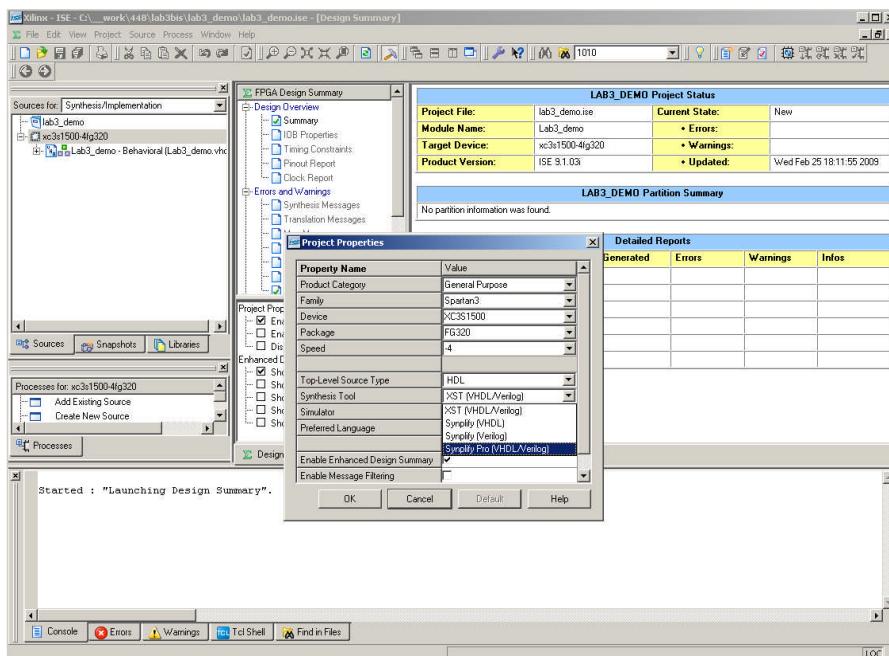
```

4.1.2 Synthesis with Simplify Pro

There is a possibility in Xilinx ISE to specify other vendors of synthesis software. Go to **Sources** for menu, right-click on the name of the chosen design. Pop-up menu should appear. Choose **Properties**.



The **Project Properties** menu should be available now. Change Synthesis Tool to Simplify Pro (VHDL/Verilog). Click **OK** – your design will use Simplify Pro for synthesis.



When Synthesis process is completed, the report from synthesis becomes available. Some of the most important features of the design are the minimum clock period and the maximum clock frequency. We can find these two parameters in the report file from Synthesis. Please remember that the values of these parameters after synthesis are different than the values of the same parameters after implementation.

```

Xilinx - ISE - C:\_work\448\lab3bis\lab3_demo\lab3_demo.ise-[Lab3_demo.srr]
File Edit View Project Source Process Window Help
Sources for: Synthesis/Implementation
Top view: Lab3_demo
Requested Frequency: 204.4 MHz
Wire load mode: top
Paths requested: 0
Constraint File(s): C:\_work\448\lab3bis\lab3_demo\Lab3_demo.sdc
ON: MT195 | This timing report estimates place and route data. Please look at the place and route tim
ON: MT197 | Clock constraints cover only FF-to-FF paths associated with the clock..

Performance Summary
*****
Worst slack in design: -1.281

Processes for Lab3_demo - Behavioral
Starting Clock Requested Frequency Estimated Period Requested Estimated Slack Clock Type
Lab3_demo|clock 204.4 MHz 162.0 MHz 4.893 6.174 -1.281 inferred

Clock Relationships
*****
Clocks          | rise to rise | fall to fall | rise to fall
Starting       |           | constraint slack |           | constraint slack
Ending         |           |           |           |
Clocks          |           |           |           |
constraint slack |           |           |           |
constraint slack |           |           |           |
constraint slack |           |           |           |

Design Summary Lab3_demo.srr
Console Errors Warnings Tcl Shell Find in Files
Ln 26 Col 23 Ln yy Col xx

```

The other important information we can find in this report is the amount of FPGA resources your design requires.

```

Xilinx - ISE - C:\_work\448\lab3bis\lab3_demo\lab3_demo.ise-[Lab3_demo.srr]
File Edit View Project Source Process Window Help
Sources for: Synthesis/Implementation
Mapping to part: xc3s1500fg320-4
Cell usage:
FDC      36 uses
GND      2 uses
MUXCY_L  30 uses
XORCY   25 uses
LUT1    25 uses
LUT2    17 uses
LUT3     5 uses
LUT4    25 uses

I/O ports: 9
I/O primitives: 8
IBUF     1 use
OBUF     7 uses
BUFGP   1 use

I/O Register bits: 0
Register bits not including I/Os: 36 (0%)

Global Clock Buffers: 1 of 8 (12%)

Total load per clock:
Lab3_demo|clock: 36

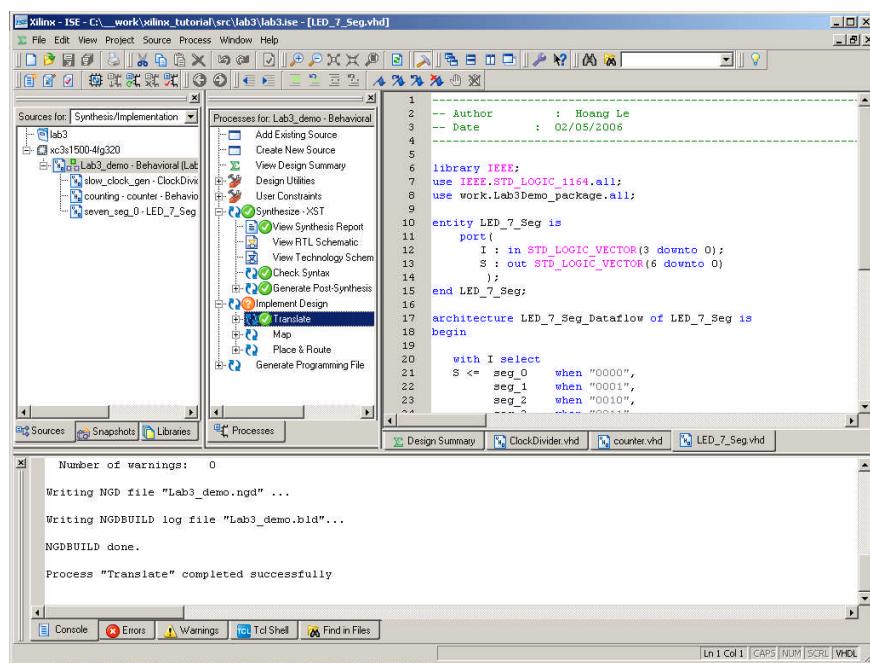
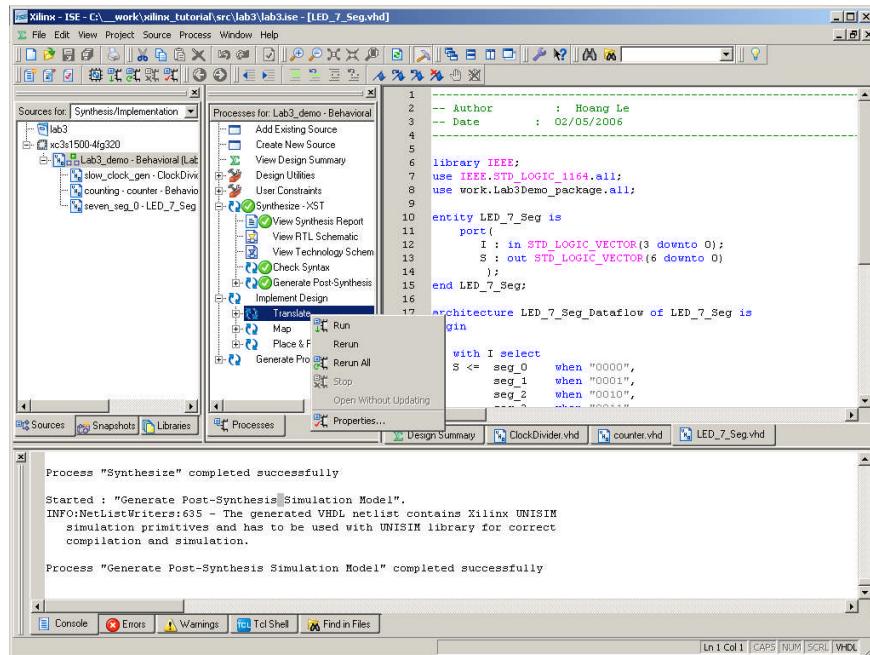
Mapping Summary:
Total LUTs: 72 (0%)

Mapper successful!
Process took 0h:00m:03s realtime, 0h:00m:03s cputime
# Wed Feb 25 18:14:38 2009
#####
Design Summary Lab3_demo.srr
Console Errors Warnings Tcl Shell Find in Files
Ln 26 Col 23 Ln yy Col xx

```

4.2 Translate

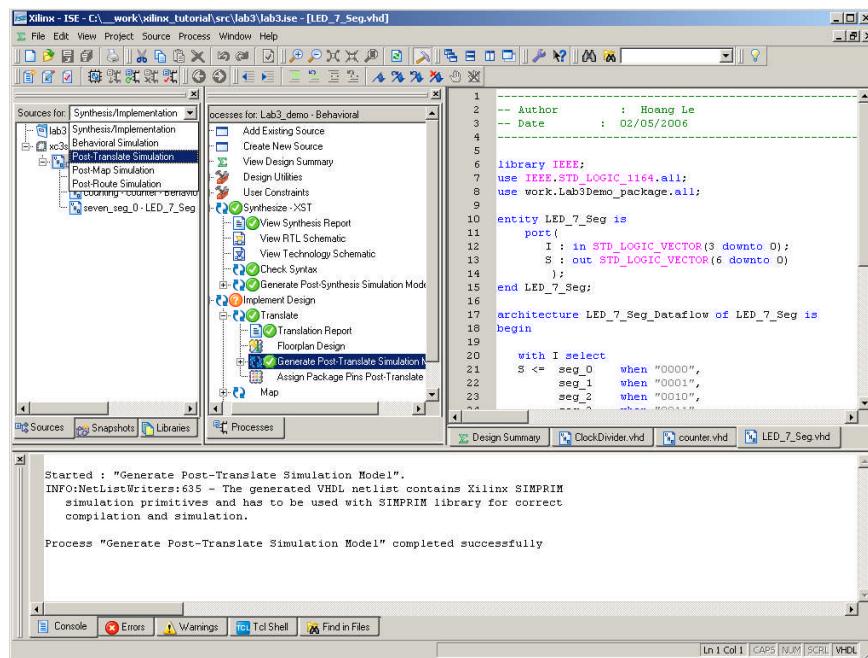
The first step in the Xilinx Design Flow for implementation is Translate. Under **Implement Design** option, choose **Translate**, and then **Run**.



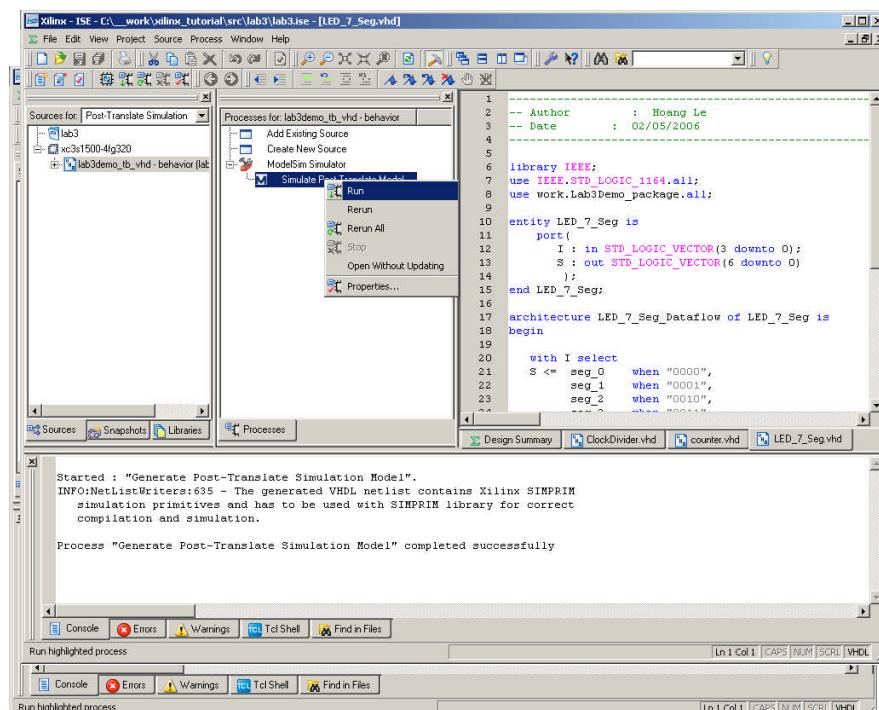
If you are successful with this part you should generate Post Translate Simulation Model.

4.3 Post-Translate Simulation

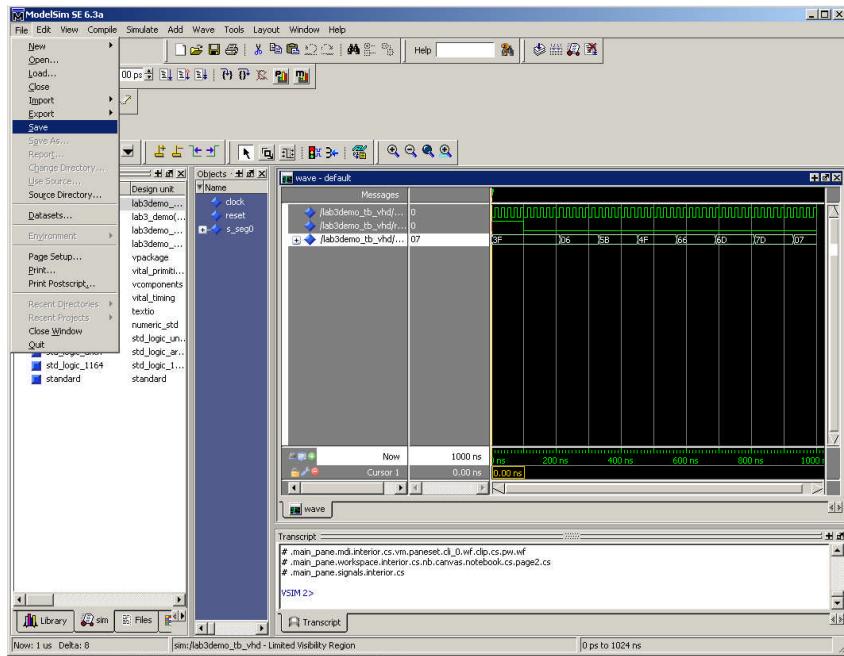
When you right-click on **Generate Post-Translate Simulation Model** then pop-up menu appears. Choose **Run** to start.



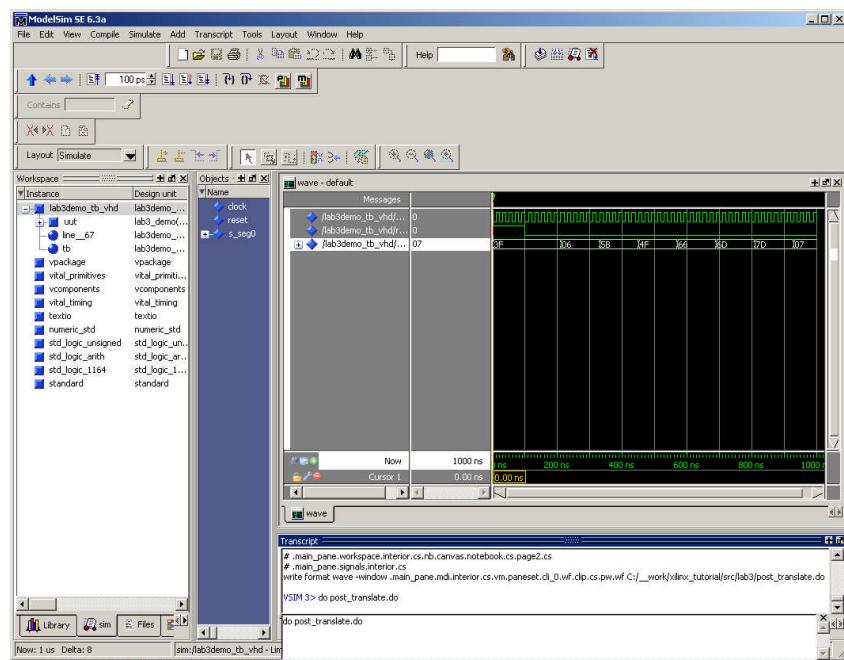
In the menu **Sources for** we choose **Post-Translate Simulation**. The testbench for this design was set as a top level. In the menu **Processes for** choose **ModelSim Simulator**, right-click and choose **Run**.



ModelSim SE will be launched and waveforms for our design should appear. We can save script for further re-simulation by saving .do type files.

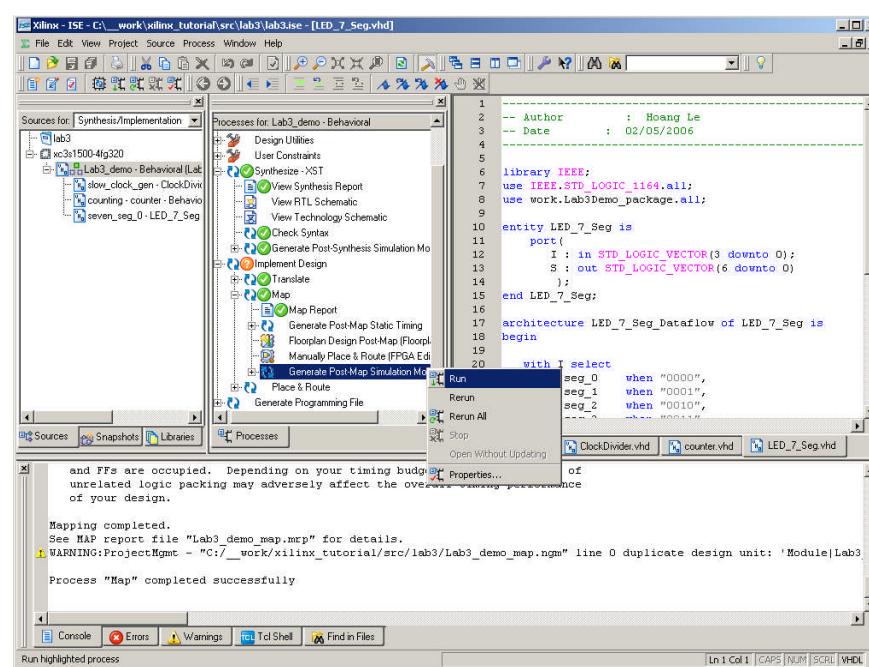
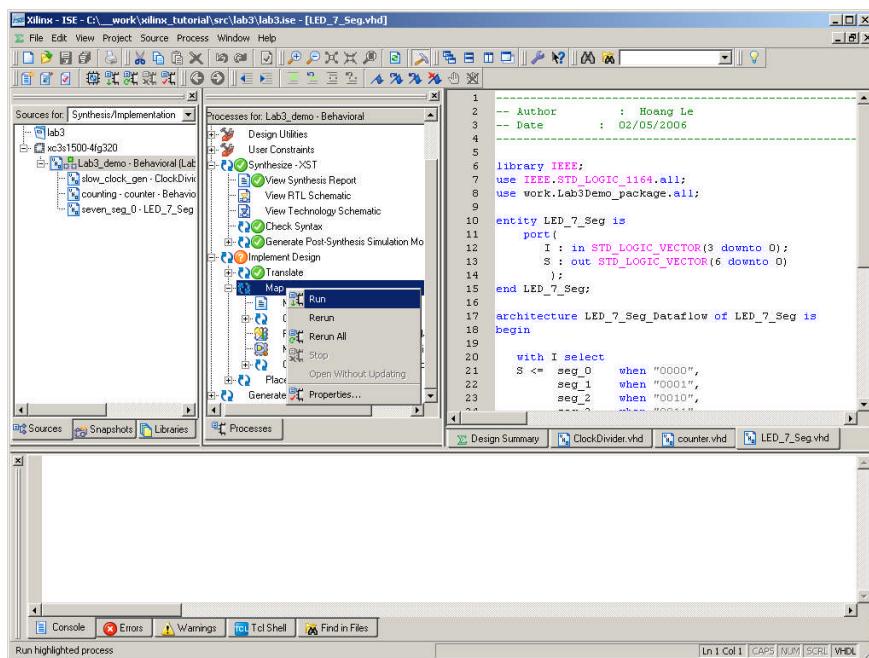


There is in the bottom part ModelSim terminal. We can type there commands such like **do name_of_script.do** and our waveform will be updated. It is possible to use terminal in every step of implementation.



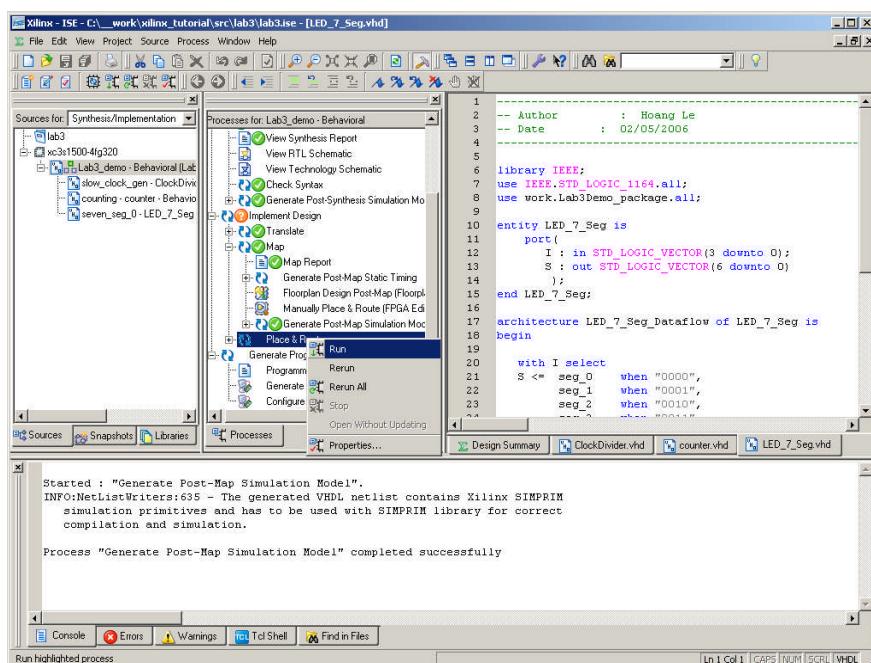
4.4 Map

Go to the menu **Sources for** and change this option to **Synthesis/Implementation**. From the **Implement Design** menu choose **Map**. Do right-click and pop-up menu should appear. Choose **Run** to start mapping process.

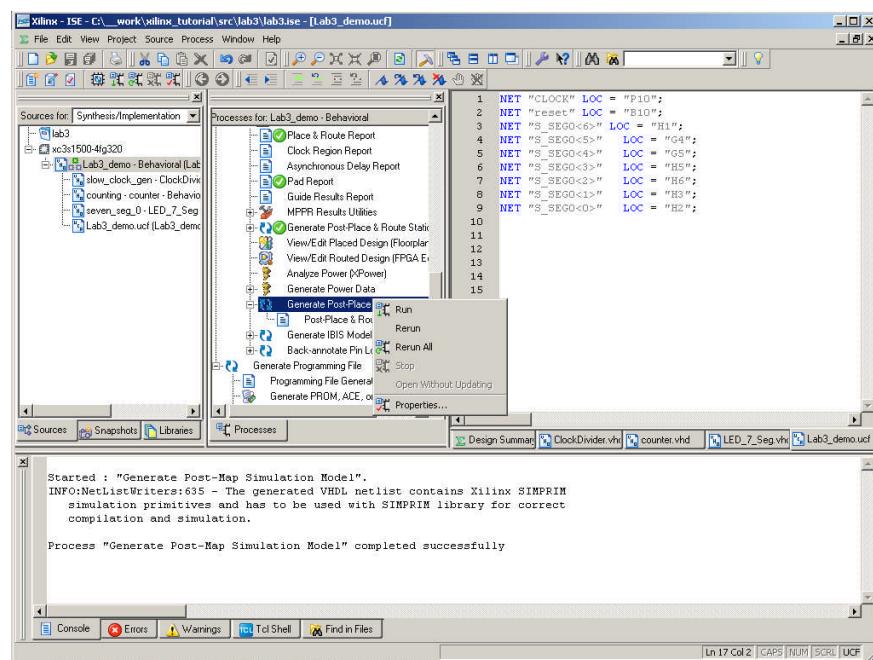


4.5 Place and Route

Go to menu **Sources for** and change this option to **Synthesis/Implementation**. From the **Implement Design** menu choose **Place and Route**. Do right-click and pop-up menu should appear. Choose **Run** to start place and route process.

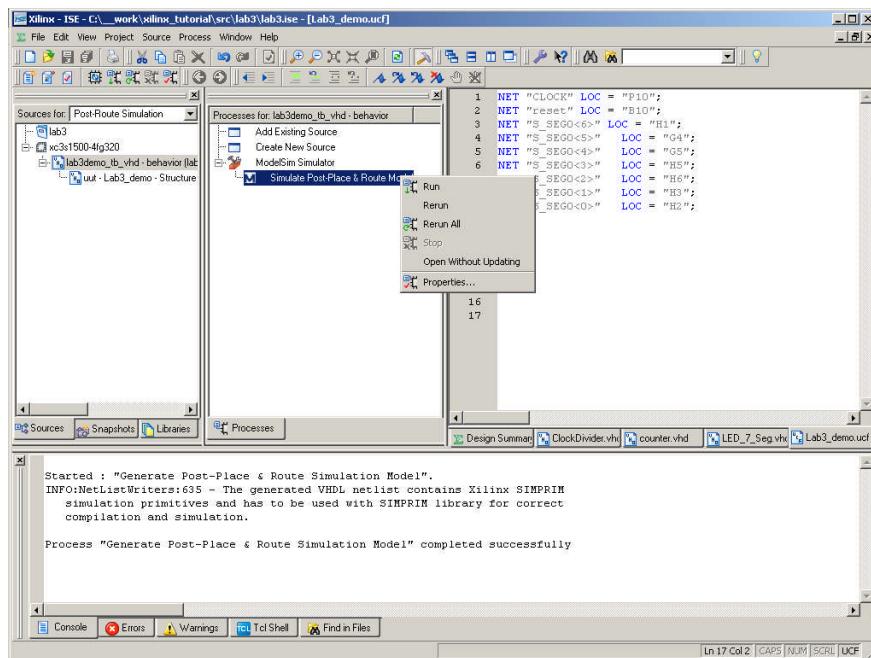


When we right-click on Generate Post Place and Route Simulation Model then pop-up menu appears and you should choose

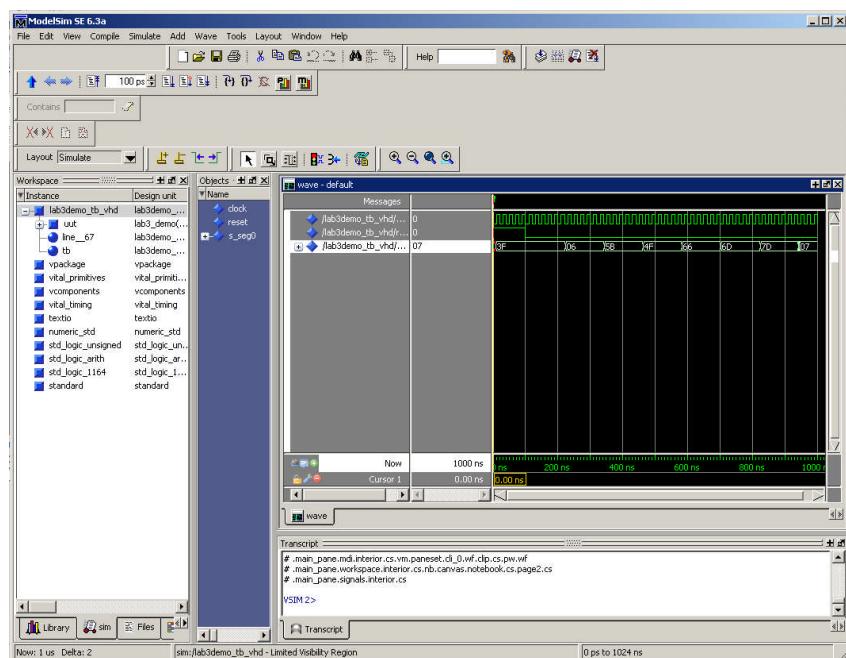


4.6 Post Place and Route Simulation

In the menu **Sources for** we choose **Post-Route Simulation**. The testbench for this design was set as a top level. In the menu **Processes for** choose **ModelSim Simulator**, right-click and choose **Run**.

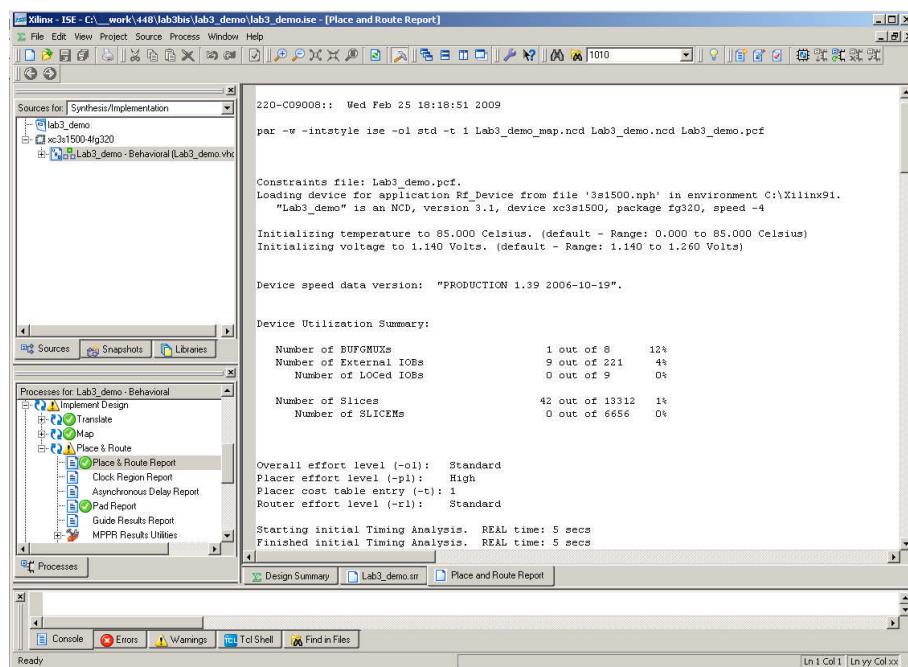
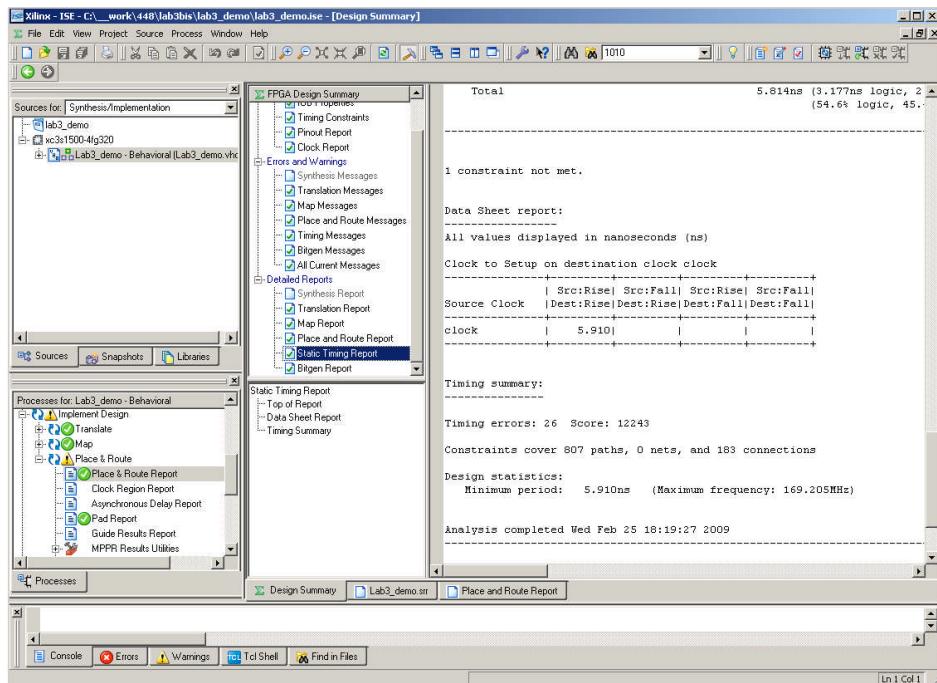


ModelSim SE will be launched and waveforms for our design should appear.



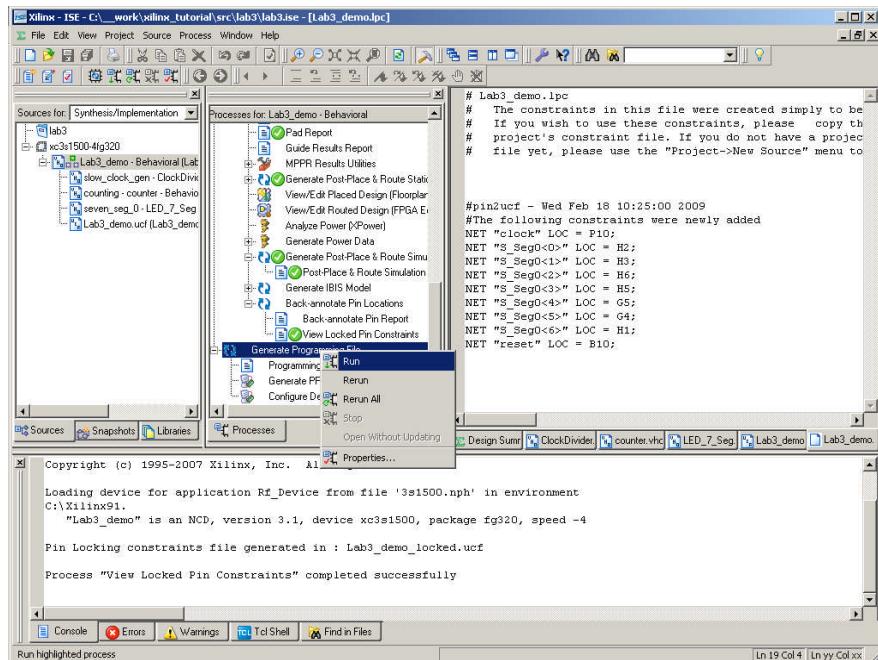
4.7 Implementation Reports

In the Place & Route Report you can find final information about the amount of resources utilized and the minimum clock period/maximum clock frequency of the design.

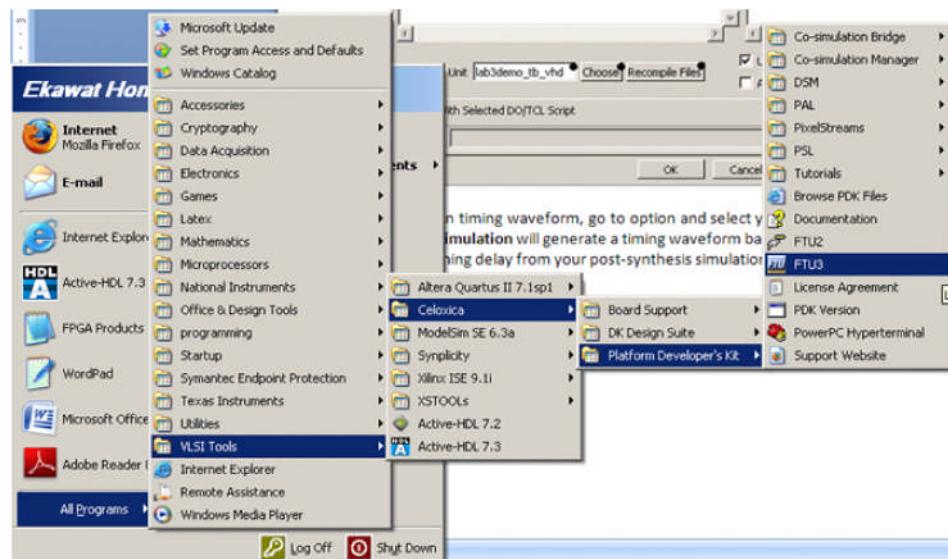


4.8 Bit Stream Generation

Go to menu **Sources for** and change this option to **Synthesis/Implementation**. Choose **Generate Programming File**, do right-click and pop-up menu should appear. Choose **Run** to start bit generation process.

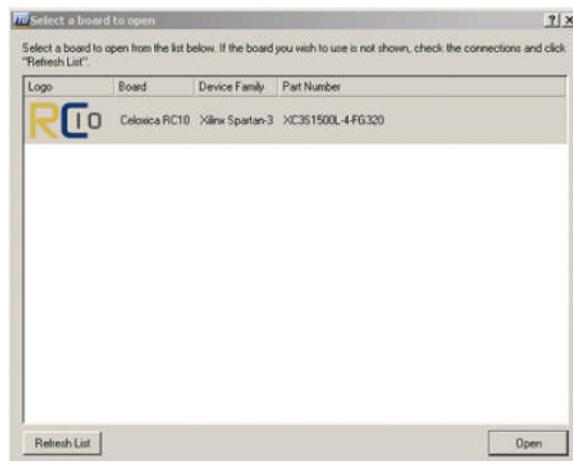


5. Uploading Bitstream to FPGA Board



Before uploading Bit file, make sure that you change your **constant values** in all your files to proper value, and re-synthesize/implement all the files. In particular, in our example, please change the value of the constant slow_clock_period in the Lab3Demo_package.vhd.

Select FTU3 program as shown in the picture above. When the program is opened, a device will be shown if it is connected and recognized. Select your FPGA and click **open**. Then, **Clear FPGA** and select the bit file located in **\implement\ver\rev1** of your workspace. Upload (**Configure**) the code and test your design whether it works correctly on the FPGA board.



Good luck! Have fun debugging =)