Xilinx ISE WebPACK Tutorial for VHDL Design on Spartan-3 FPGA

The purpose of this tutorial is to guide you through the steps of creating a project using Xilinx ISE WebPACK. It serves as a very simple way to get you start using the tool. Details are not mentioned. You should read *ISE 10.1 Quick Start Tutorial*, which can be downloaded from Xilinx website, for complete instructions and information. The latest version of the Xilinx FPGA design solutions software is called ISE Design Suite 10.1. ISE WebPACK is one of the Design Suite softwares and the latest update is service pack 2. It is free and you can download a copy from the Xilinx website. You need to register an account at the Xilinx website in order to download the software.

Unlike Code Composer Studio, ISE does not require a hardware to run the software. You can design and program your project and build up the programming file that can be loaded to FPGA without having the hardware at hand. That means you can work on your project with this software anywhere. You can compile your codes and simulate the circuits in ISE. Once the programming file is generated, all you have to do is to bring the whole project or programming file to the lab and loaded to the hardware to see if it actually works. The FPGA chips supported by ISE WebPACK and the full version are listed in Figure 1.

		ISE WebPACK	*ISE Foundation*
Platforms		Microsoft® Windows® XP Professional (32-bit) Microsoft Vista Business (32-bit) Red Hat Enterprise Linux 4 WS (32-bit) Red hat Enterprise Linux Enterprise 5 (32-bit) SUSE Linux Enterprise 10 (32 bit)	Microsoft Windows XP Professional (32/64-bit) Microsoft Vista Business (32/64-bit) Red Hat Enterprise Linux 4 WS (32/64-bit) Red Hat Enterprise Linux Desktop 5 (32/64-bit) SUSE Linux Enterprise 10 (32 and 64 bit))
Devices (FPGAs)	Virtex Series	**Virtex:** XCV50 - XCV600 **Virtex-E:** XCV50E - XCV600E **Virtex-II:** XC2V40 - XC2V500 **Virtex-II Pro:** XC2VP2 - XC2VP30 **Virtex-4:** **LX:** XC4VLX15, XC4VLX25 **SX:** XC4VSX25 **FX:** XC4VFX12 **Virtex-5:** **LX:** XC5VLX30, XC5VLX50 **LXT:** XC5VLX30T, XC5VLX50T **FXT:** XC5VFX30T **Virtex Q:** XQV100- XQV600 **Virtex QR:** XQVR300, XQVR600 **Virtex-E Q:** XQV600E	**Virtex:** All **Virtex-E:** All **Virtex-II/Pro:** All **Virtex-4:** **LX:** All **SX:** All **FX:** All **Virtex-5:** **LX:** All **LXT:** All **FXT:** All **Virtex Q/QR:** All **Virtex-E Q:** All
	Spartan Series	**Spartan-II/IIE:** All **Spartan-3:** XC3S50 - XC3S1500 **Spartan-3A:** All **Spartan-3AN:** All **Spartan-3A DSP:** XC3SD1800A **Spartan-3E:** All **Spartan-3L:** XC3S1000L, XC3S1500L **XA (Xilinx Automotive) Spartan-3:** All	**Spartan-II/IIE:** All **Spartan-3:** All **Spartan-3A:** All **Spartan-3AN:** All **Spartan-3A DSP:** All **Spartan-3E:** All **Spartan-3L:** All **XA (Xilinx Automotive) Spartan-3:** All
Devices (CPLDs)	CoolRunner™ XPLA3 CoolRunner-II CoolRunner-IIA XC9500 Series	All	

Figure 1: ISE features (from Xilinx ISE Product Brief).

Start Xilinx ISE 10.1 by clicking the icon on the desktop. The following window will show up (Figure 2).

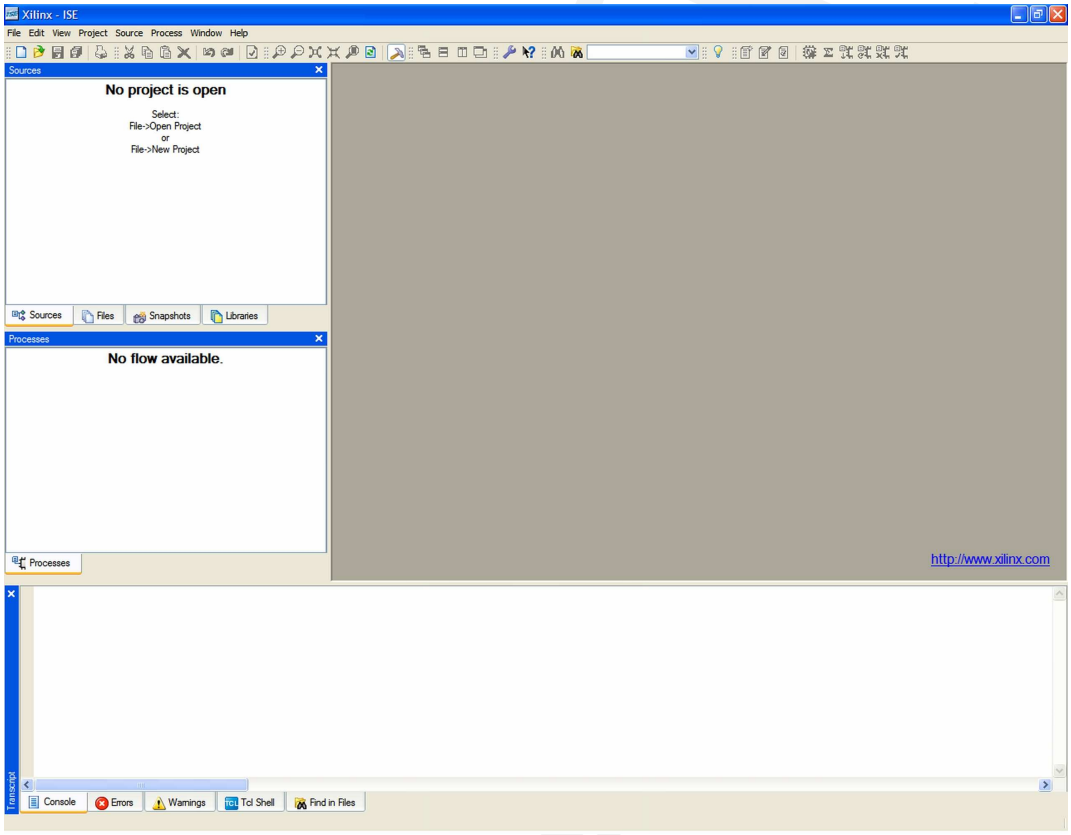

Figure 2: ISE WebPACK starting window.

From the File menu, select New Project. The New Project Wizard window will show up as shown in Figure 3.

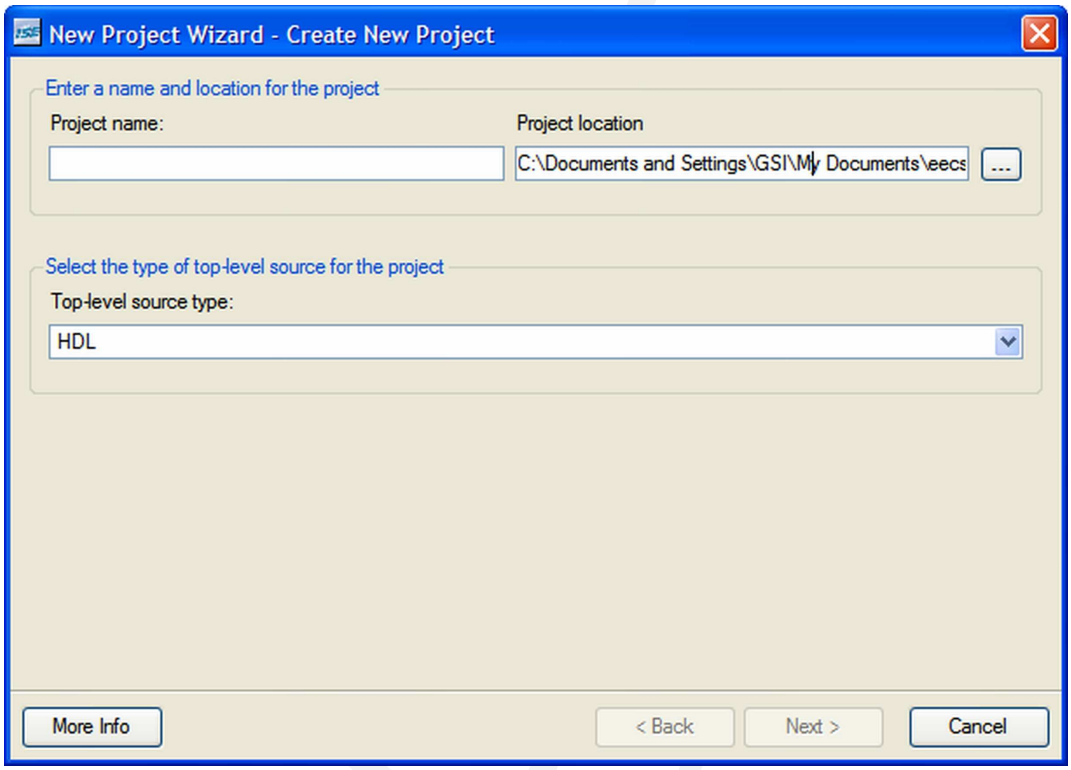

Figure 3: New Project Wizard window.

In the Create New Project window, choose a Project Name and click Next.

Then we need to specify the Device Properties. Follow the settings as shown below in Figure 4. The FPGA chip we use on the starter board is the Xilinx Spartan-3 FPGA XC3S1000. This chip has 1000K system gates with FT256 package (173 User I/O). The speed is set to be −4. When you are done with the setting, click Next.

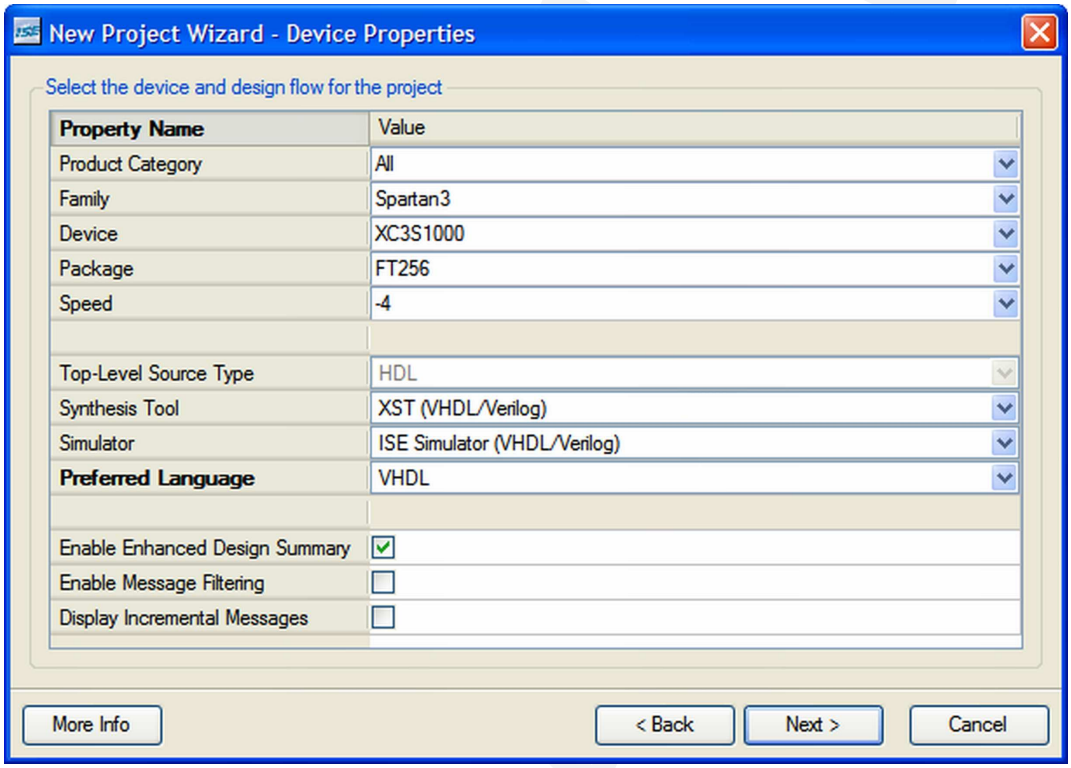

Figure 4: Set device properties.

Next, we need to create a new source for the project. In the Create New Source window as shown in Figure 5, click on New Source.

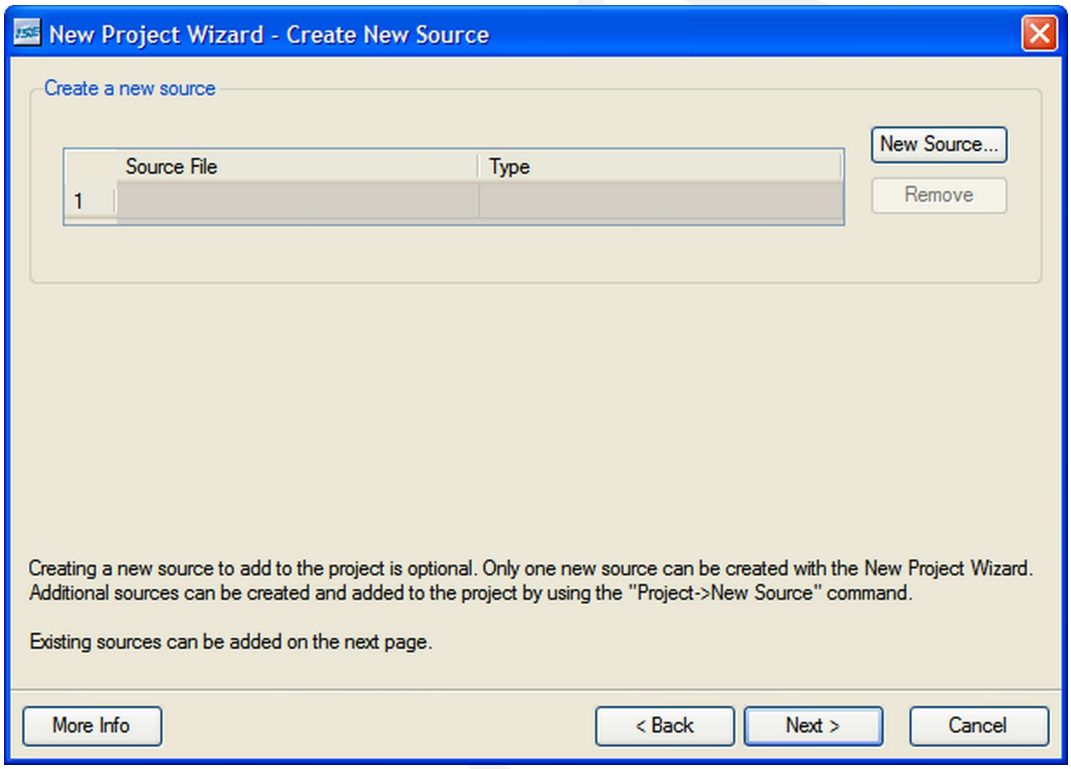

Figure 5: Create new source window.

Then the Select Source Type window will show up as in Figure 6. Choose the VHDL Module from the list and fill out the File Name. Check "Add to project" and click Next.

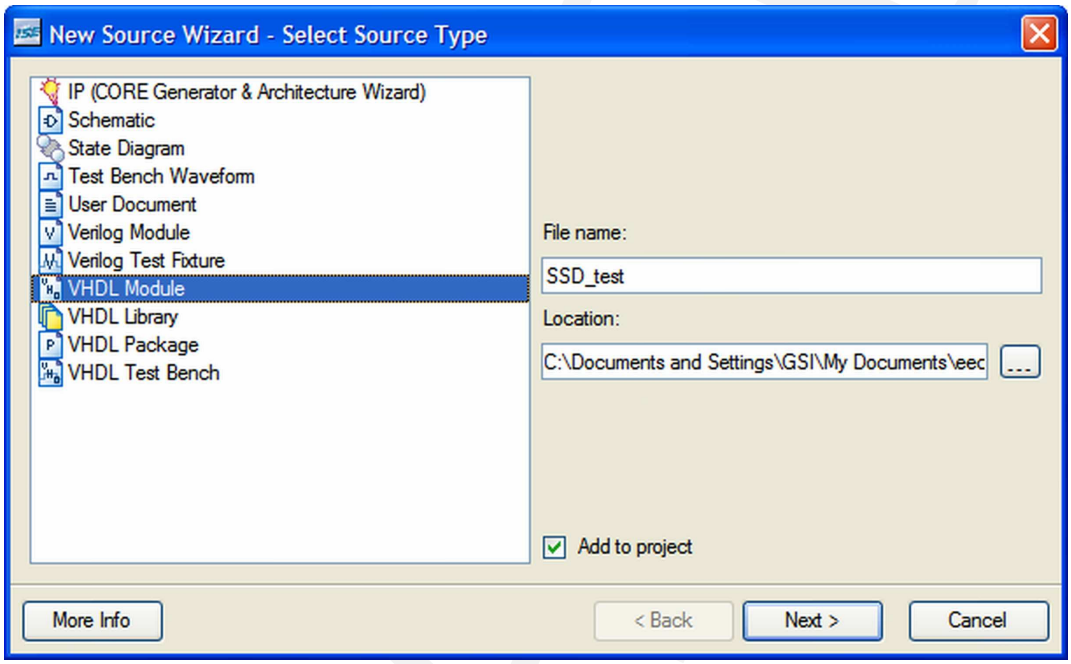

Figure 6: Select source type.

Then you need to define the I/O signals in the VHDL module you just created in the Define Module window as shown in Figure 7. Fill out the Port Name, Direction (in, out, or inout), and specify the number of bits (Bus, MSB, LSB) and click Next.

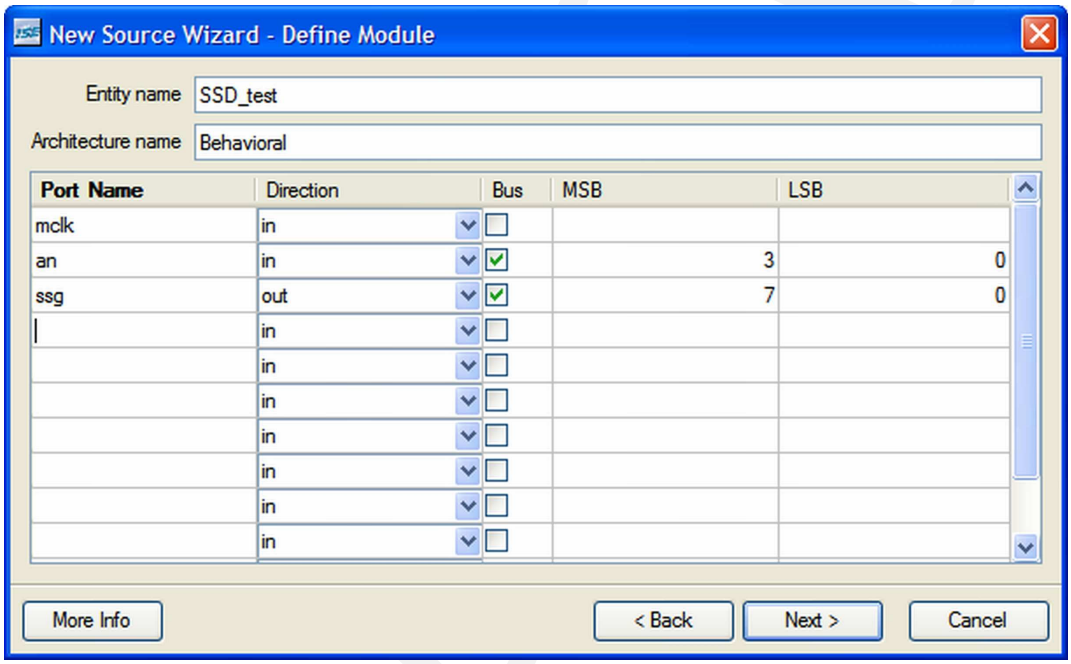

Figure 7: Define module window.

When you are done, check out the Summary as shown in Figure 8 and click Finish. You should see the source file you just created in the file list as in Figure 9. Click Next.

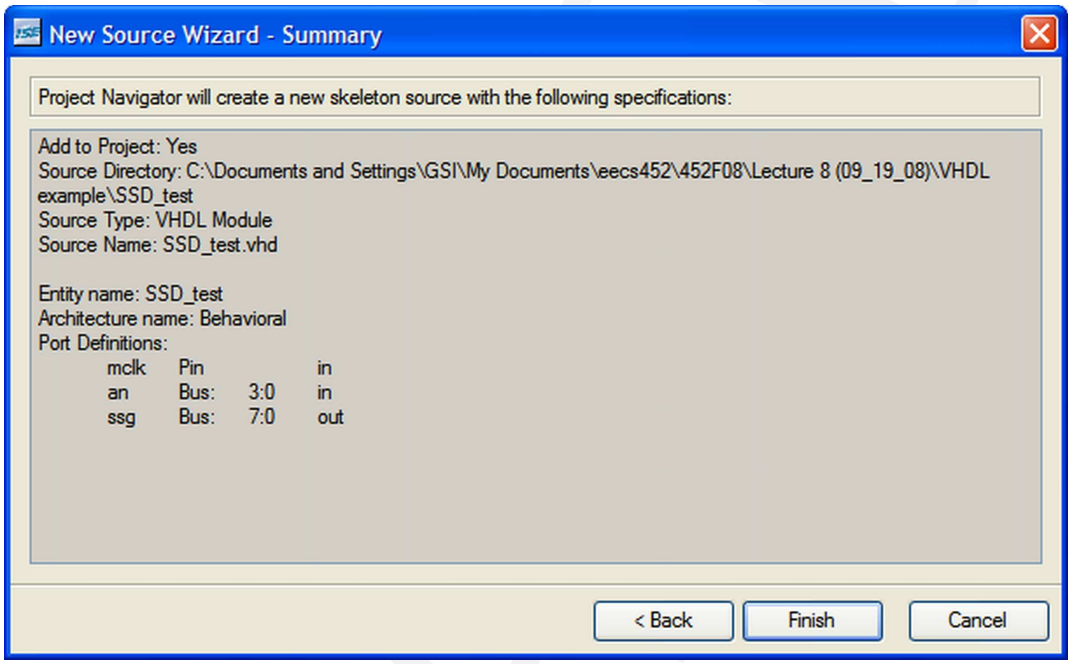

Figure 8: New source file summary.

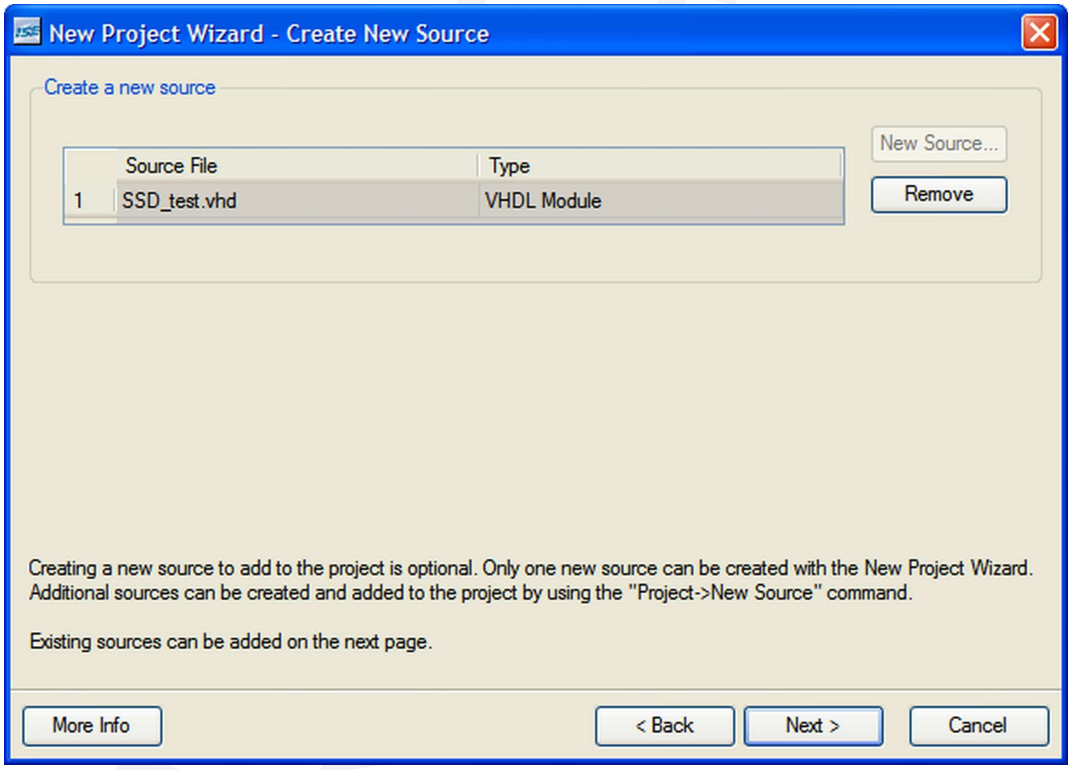

Figure 9: New source file list.

In the following window, you can add existing source files to the project as seen in Figure 10. Once you are done, click Next.

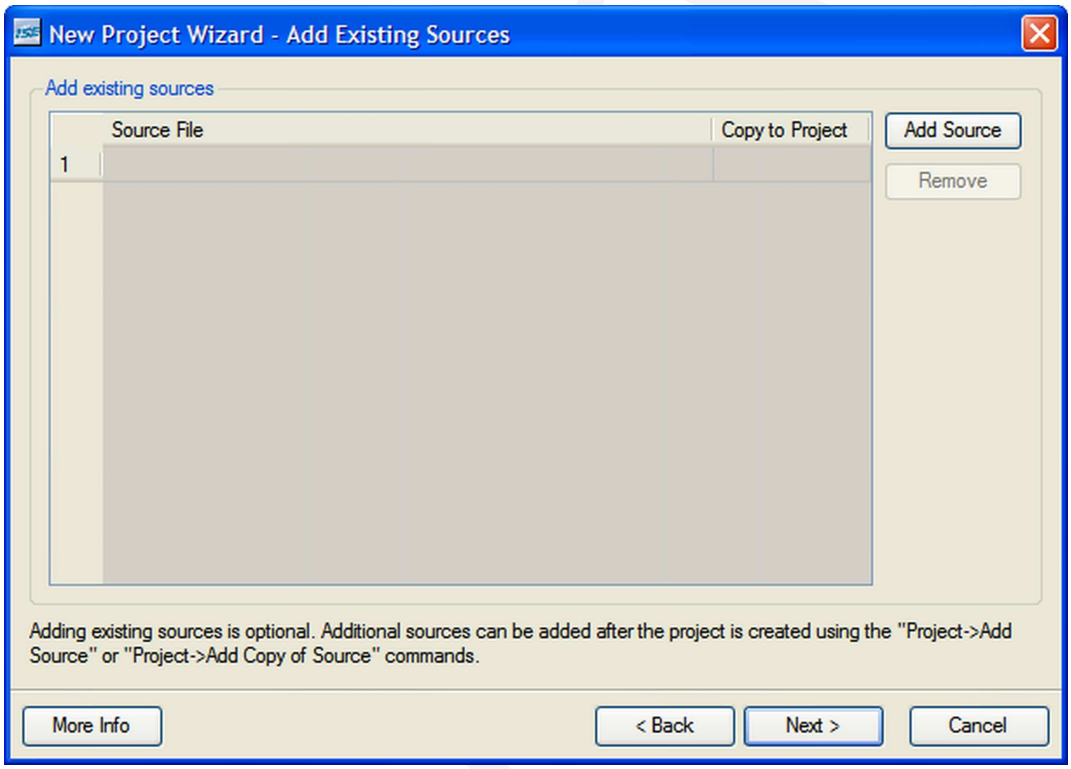

Figure 10: Add existing sources.

Once all the files are created/added, check the Project Summary as shown in Figure 11 and click Finish. When all the above steps are done, you will see the main ISE window as shown in Figure 12.

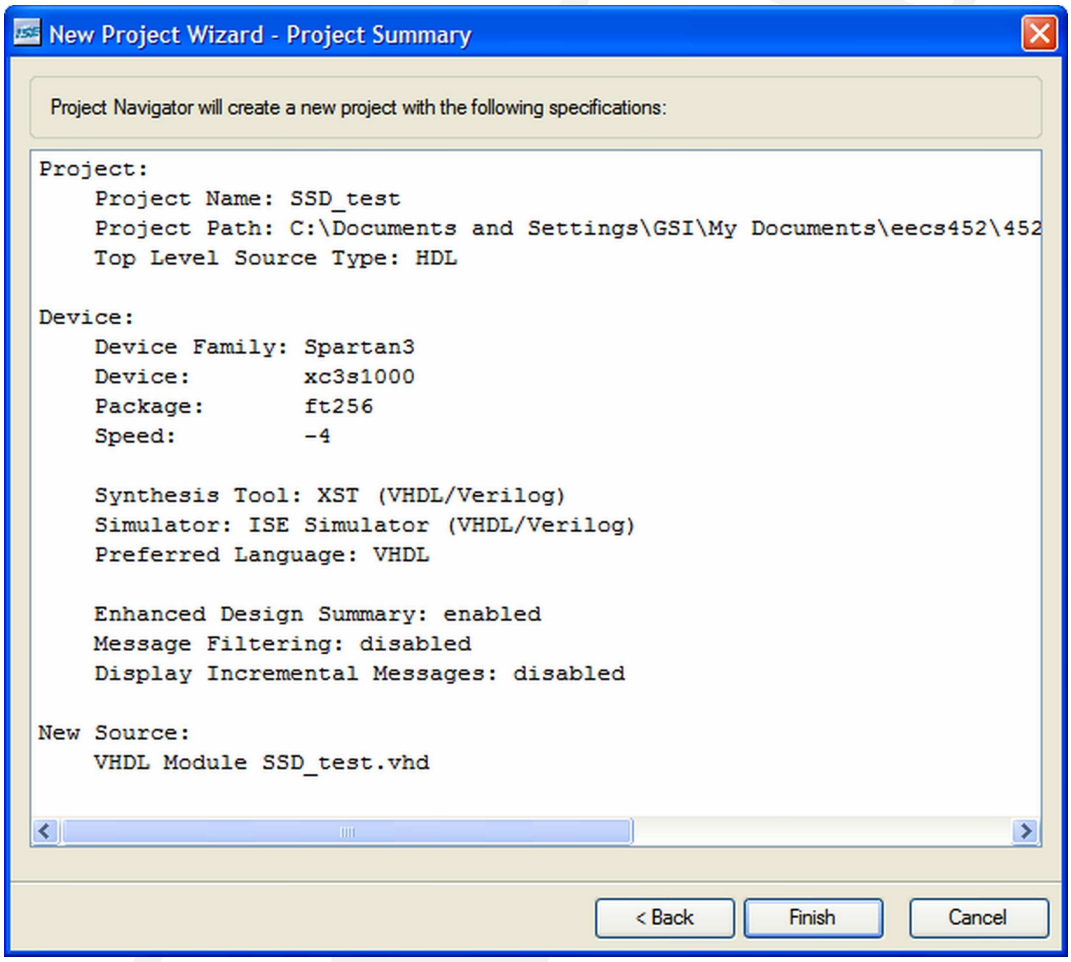

Figure 11: Project Summary.

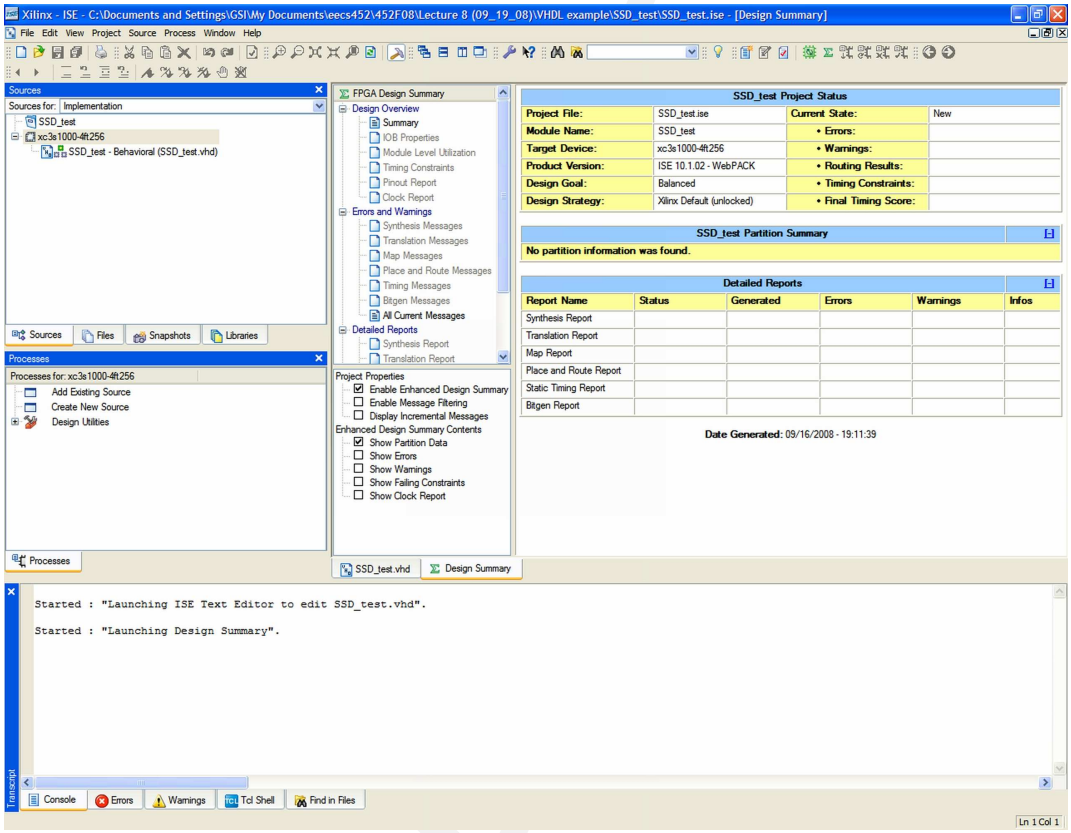

Figure 12: The ISE working environment.

Now we finished creating a new project. Next we can edit the source file using the ISE build-in editor.

The details of the design process and VHDL programming will be mentioned in lecture and lab. Basically, you need to edit all the source files and define the constraints file for the project. The code generated by ISE from the template when you create a new source code in the earlier step is shown in the editor window as in Figure 13.

```
1  ----------------------------------------------------------------------------------
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    19:11:35 09/16/2008
6  -- Design Name:
7  -- Module Name:    SSD_test - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 ----------------------------------------------------------------------------------
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity SSD_test is
31     Port ( mclk : in  STD_LOGIC;
32            an : in  STD_LOGIC_VECTOR (3 downto 0);
33            ssg : out  STD_LOGIC_VECTOR (7 downto 0));
34 end SSD_test;
35
36 architecture Behavioral of SSD_test is
37
38 begin
39
40
41 end Behavioral;
42
43
```

Figure 13: A VHDL example code.

When finish editing the source files, choose the main VHDL source file in the Sources panel, and then you can click on Synthesize, Implement Design, and Generate Programming File, and Configure Target Device in the Processes panel as seen in Figure 14. Once you finish all the processes without any error, iMPACT will open automatically.

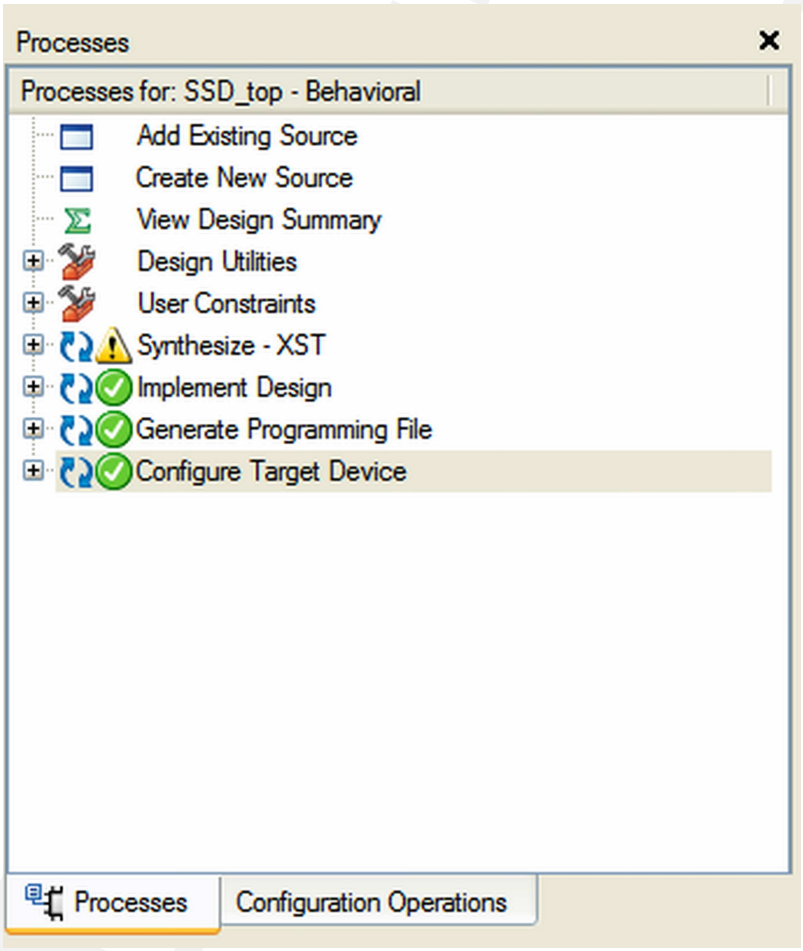

Figure 14: Processes window.

iMPACT is the tool used to load your design configuration from the ISE to the hardware. The starting window is shown in Figure 15. The easiest way to load the

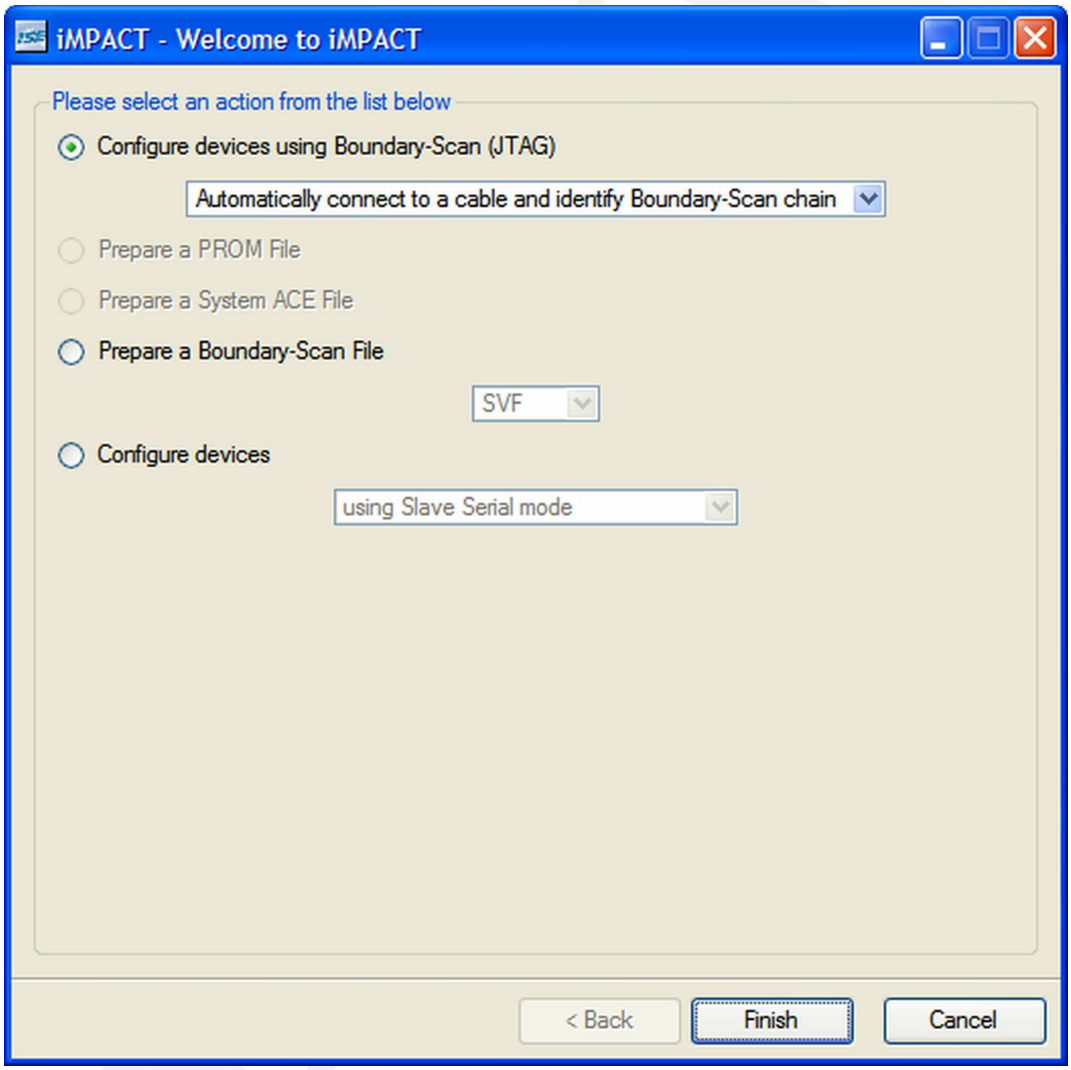

Figure 15: iMPACT starting window.

design into the chip is using the JTAG interface. Leave the default setting in the starting window and click Finish.

Next, the "Assign New Configuration File" window will show up as in Figure 16. Choose the .bit file of your design and click Open.

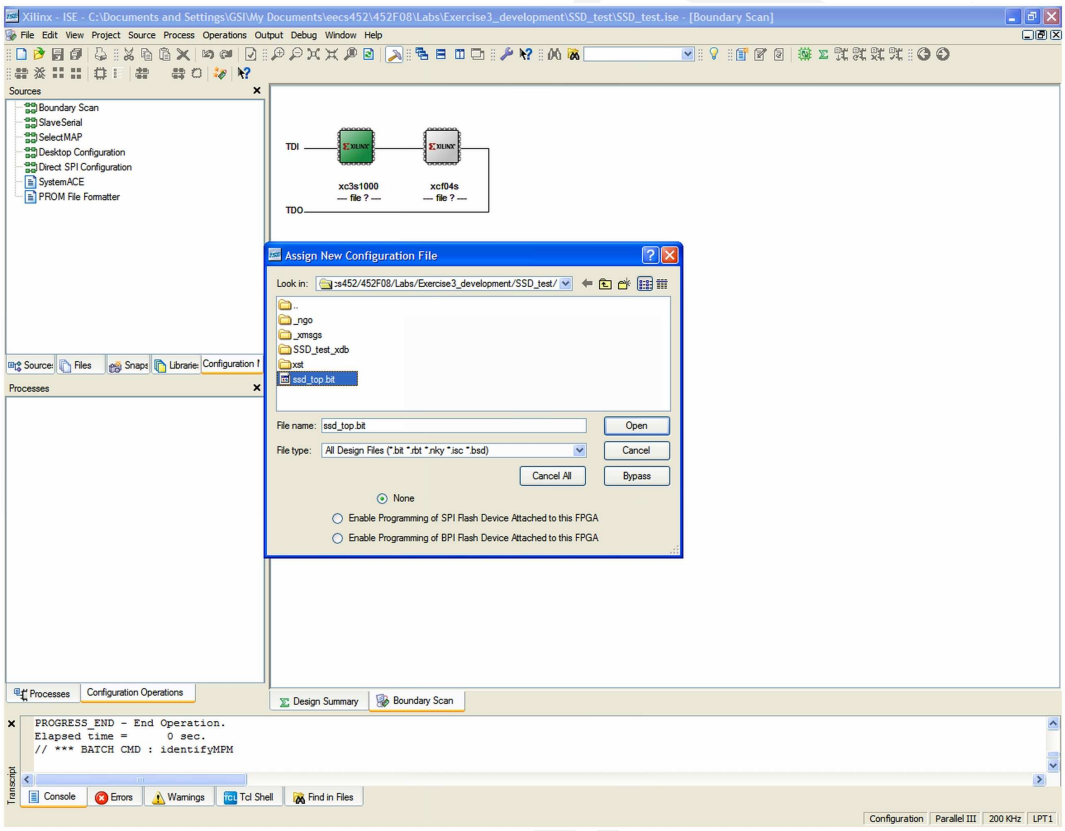

Figure 16: Assign new configuration file to FPGA

Then you will be asked to assign the new configuration file for the EPROM as shown in Figure 17. For now, since we are not using the EPROM, simply click Bypass.

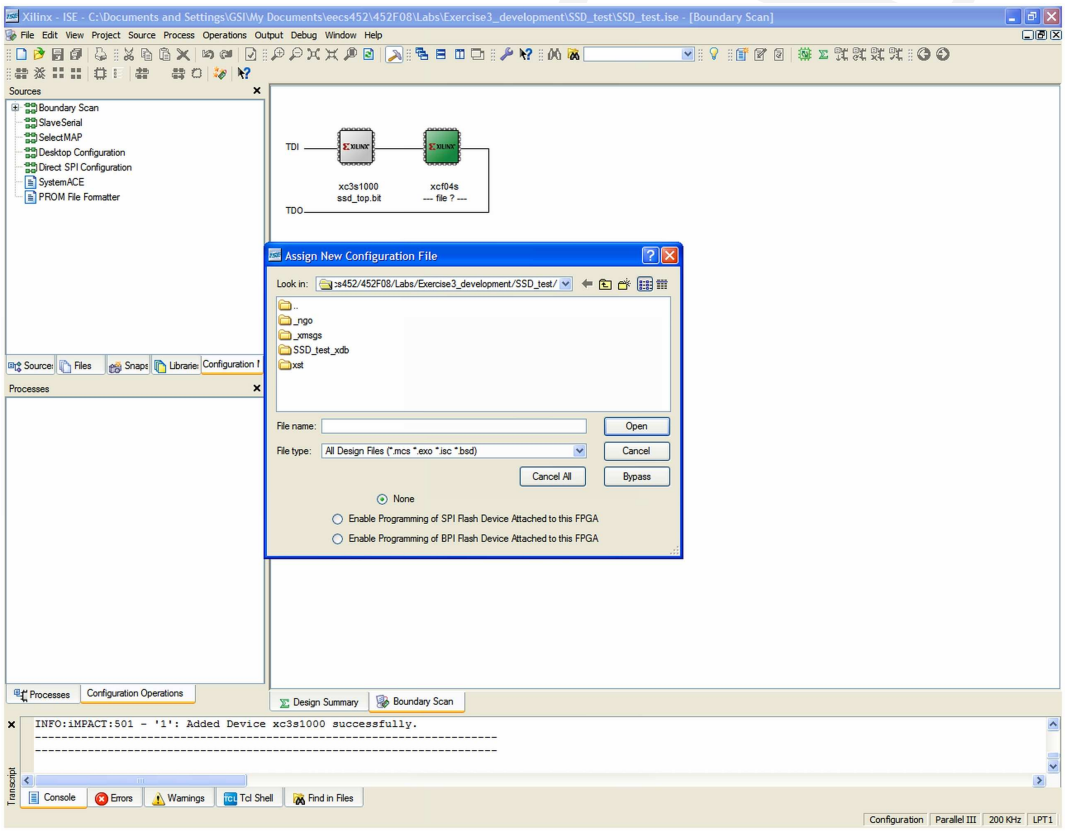

Figure 17: Assign new configuration file to EPROM

Once you finish assigning the configuration file, you will see the following window as shown in Figure 18. Simply click Okay and continue without changing the setting.

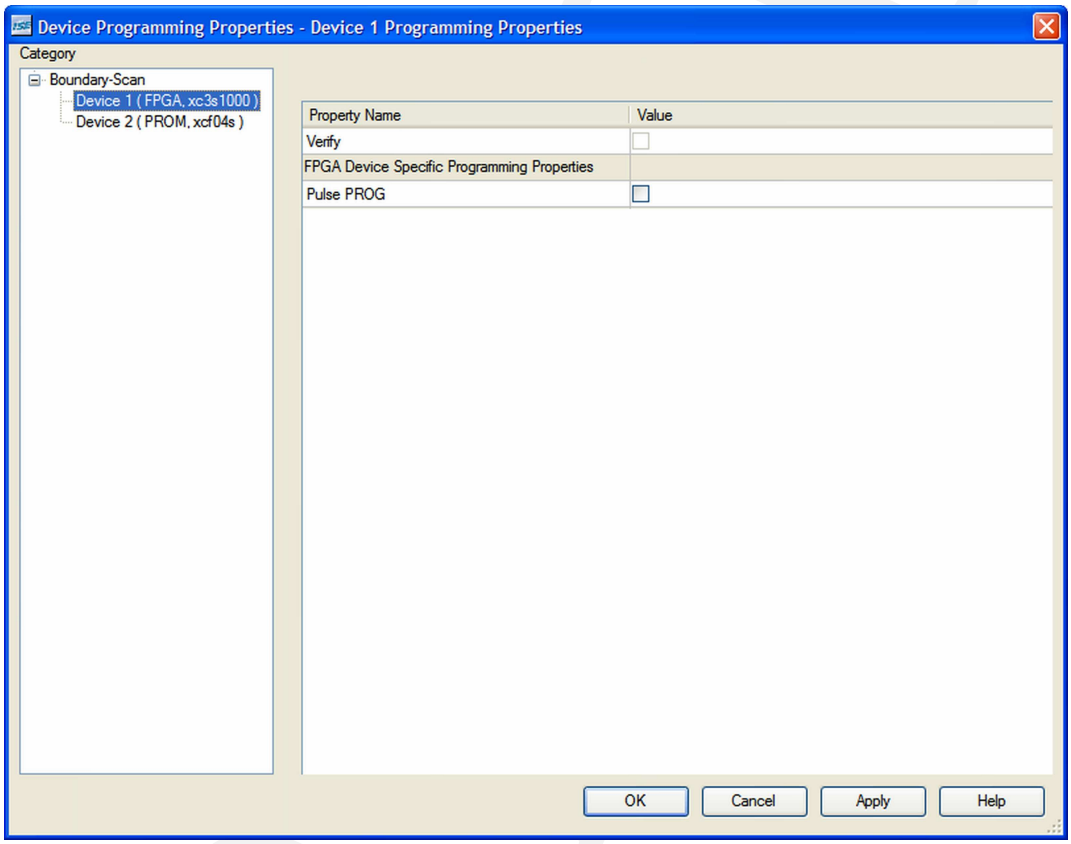

Figure 18: iMPACT window.

Right-click on the chip (XC3S1000) and choose Program as shown in Figure 19.

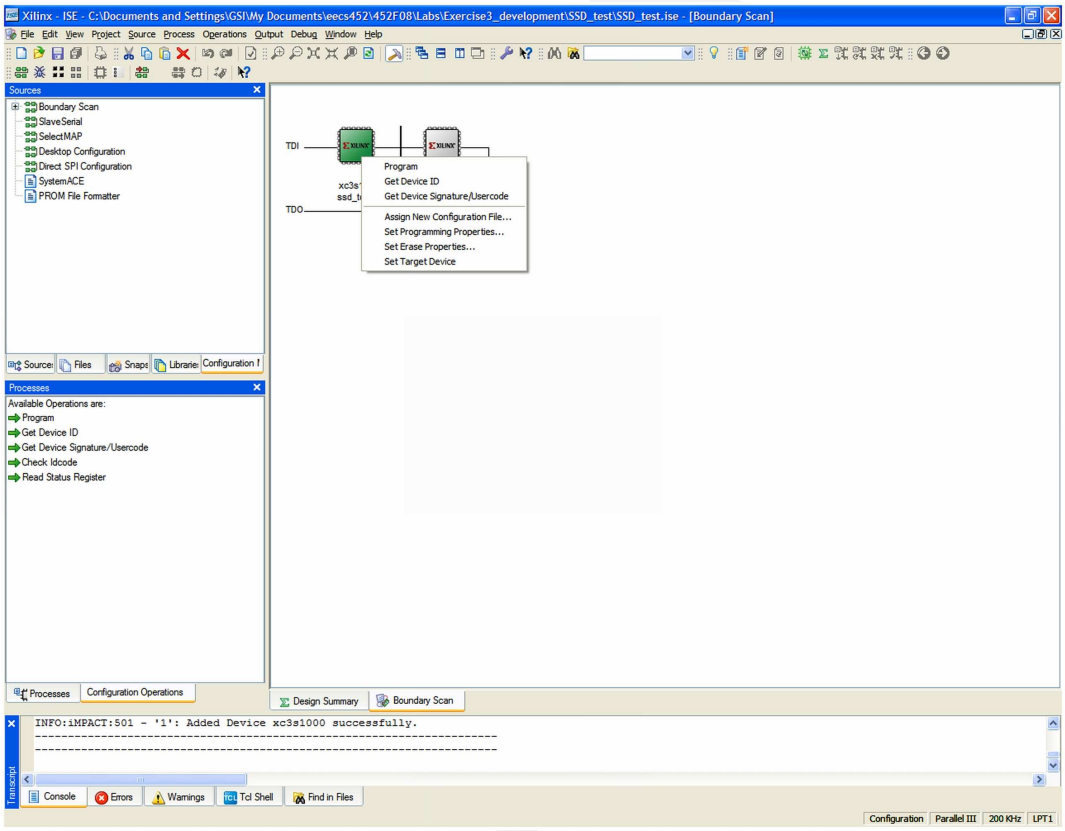

Figure 19: Program the FPGA.

Next you will see the progress of loading the configuration to the FPGA as shown in Figure 20.

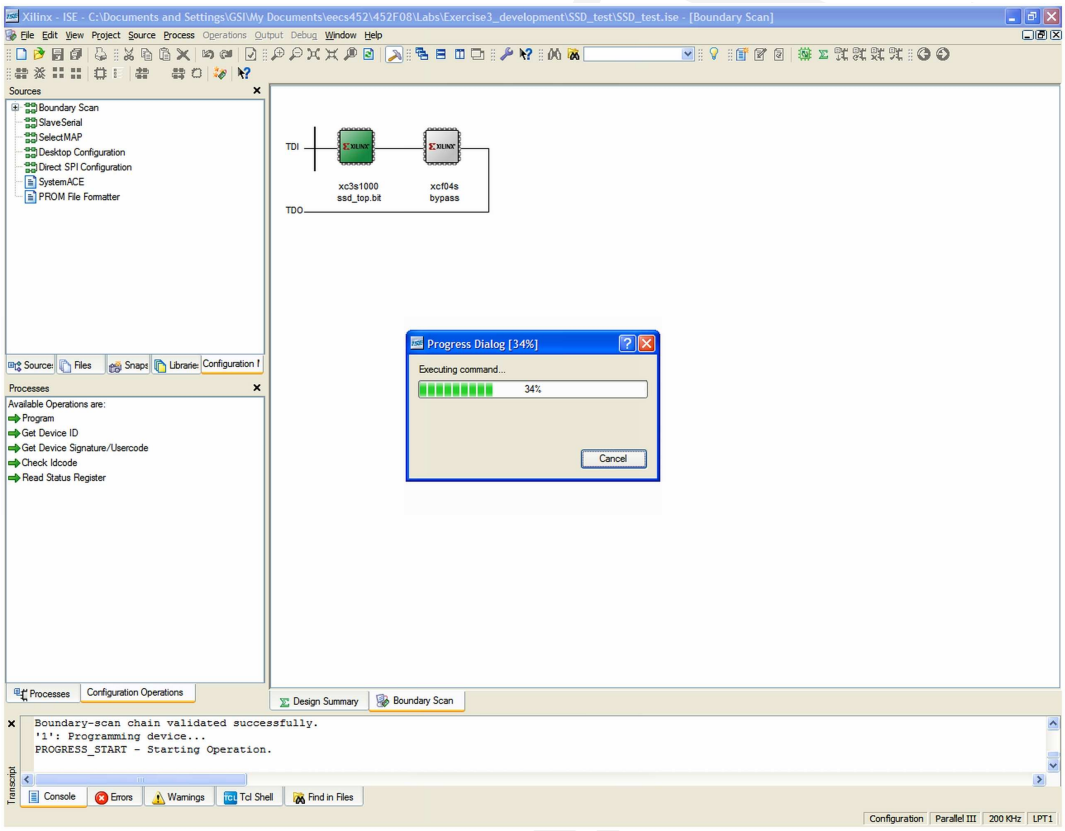

Figure 20: Programming in progress.

When the configuration is done and successful, you will see "Program Successful" in the iMPACT window as shown in Figure 21, and the FPGA board should start running.

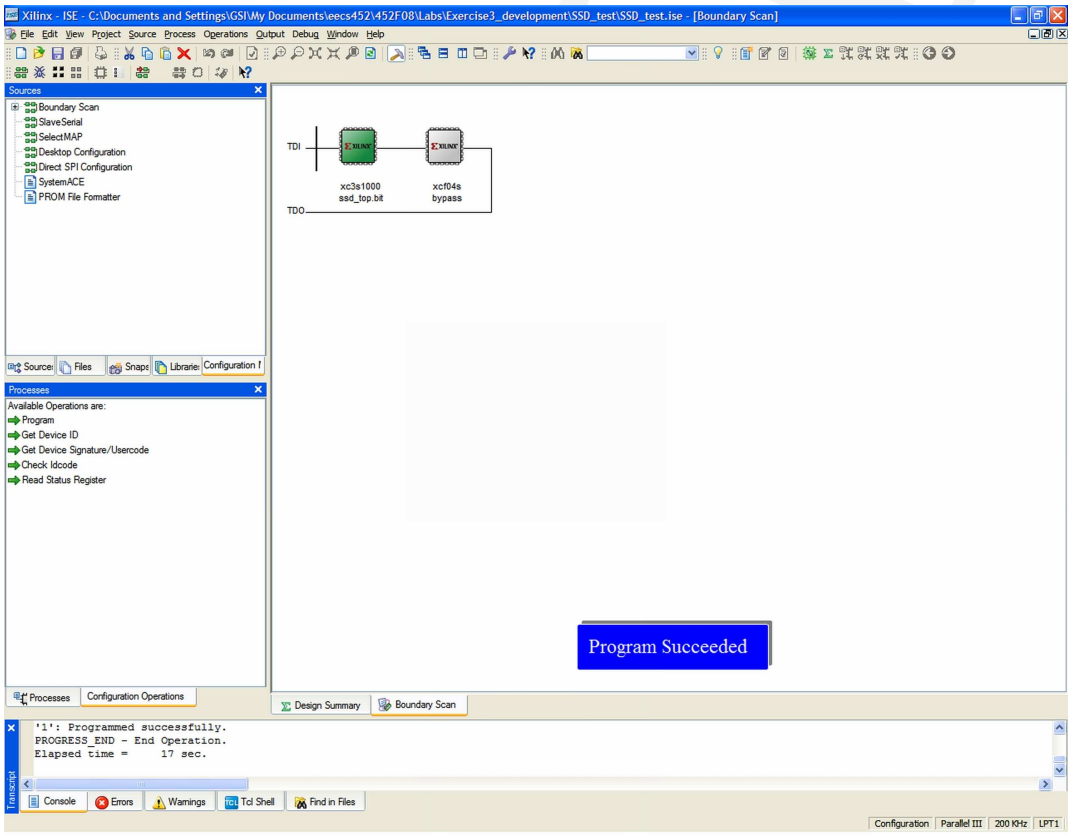

Figure 21: Programming succeeded.