USER MANUAL
OF

CPLD PROJECT BOARD

Manufactured By

Regd. Office & Works:
4A, Shree Sadgurunivas Soc.,16/5/1 Hingne Khurd, Sinhgad Road, Pune- 411051
Ph.-020 24356456 / +91 99211 59583

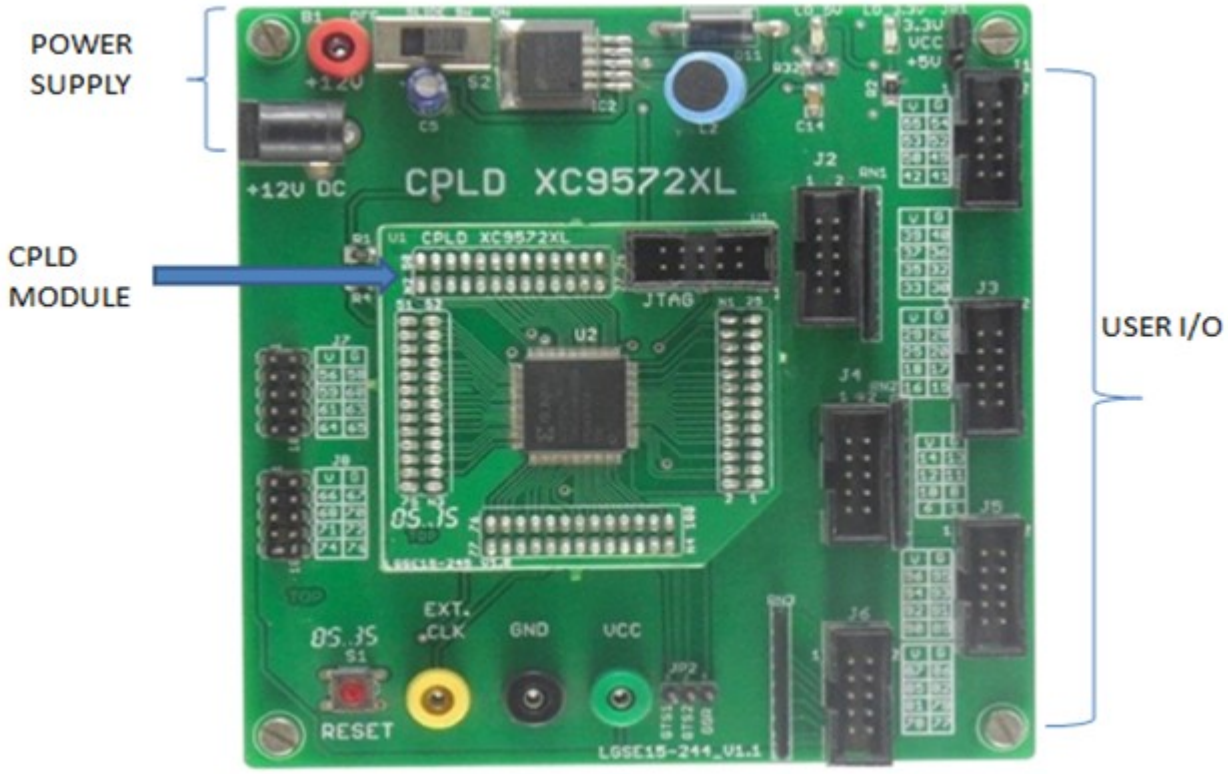
Email: - support@logsun.com
Web: www.logsun.com | www.logsunonline.com

UNIVERSIAL EMBEDDED TRAINER WITH CPLD

INDEX

A. CPLD

A. INTRODUCTION:



Complex programmable logic devices (CPLDs) are integrated circuits (ICs) or chips that application designers configure to implement digital hardware such as mobile phones. CPLDs can handle significantly larger designs than simple programmable logic devices (SPLDs), but provide less logic than field programmable gate arrays (FPGAs).

CPLDs contain several logic blocks, each of which includes eight to 16 macrocells. Because each logic block performs a specific function, all of the macrocells within a logic block are fully connected. Depending upon the application, however, logic blocks may or may not be connected to one another.

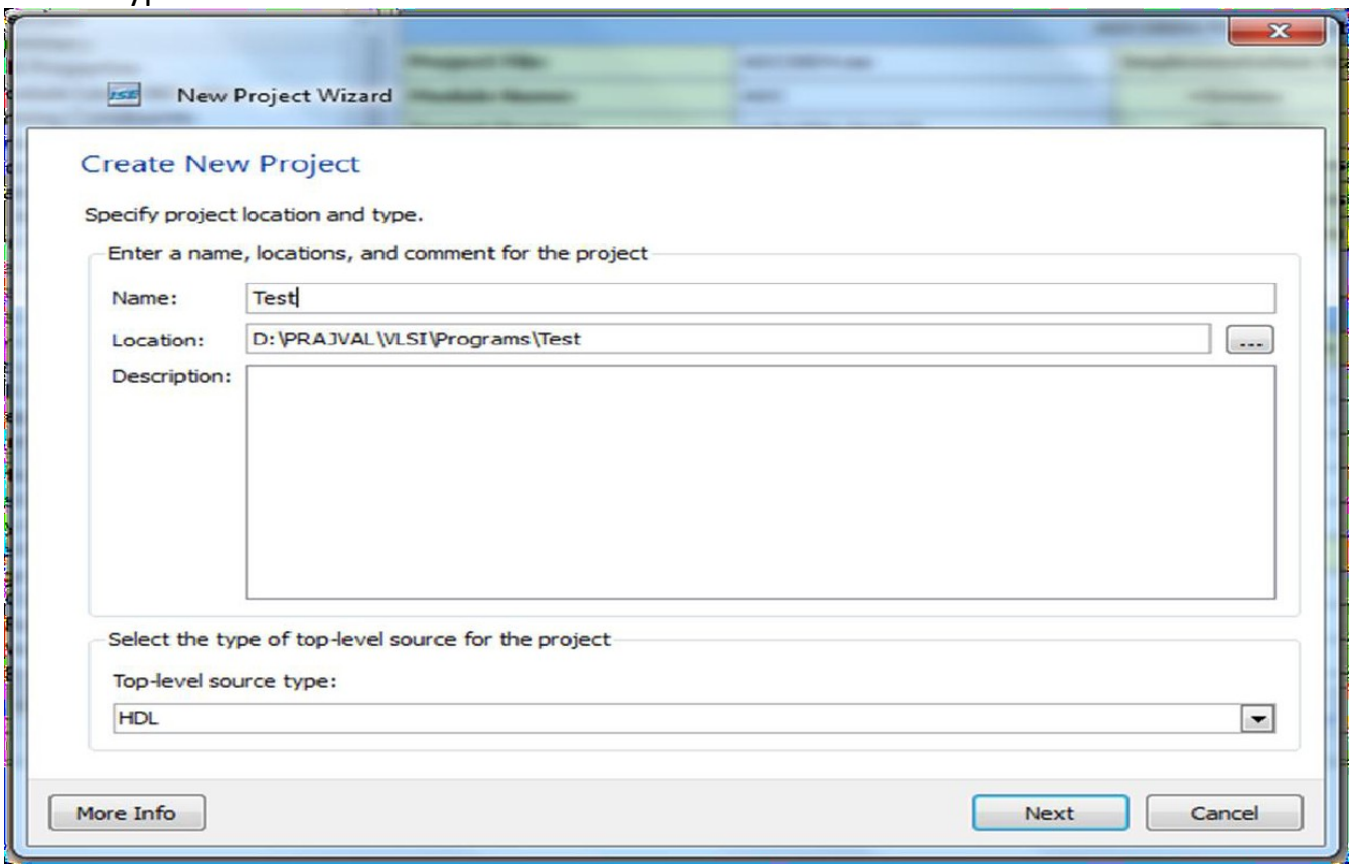
Most complex programmable logic devices contain macrocells with a sum-of-product combinatorial logic function and an optional flip-flop. Depending on the CPLD, the combinatorial logic function supports from four to sixteen product terms with wide fan-in. Complex programmable logic devices also vary in terms of logic

gates and shift registers. For this reason, CPLDs with a large number of logic gates may be used in place of FPGAs. Another CPLD specification denotes the number of product terms that a macrocell can manage. Product terms are the product of digital signals that perform a specific logic function.

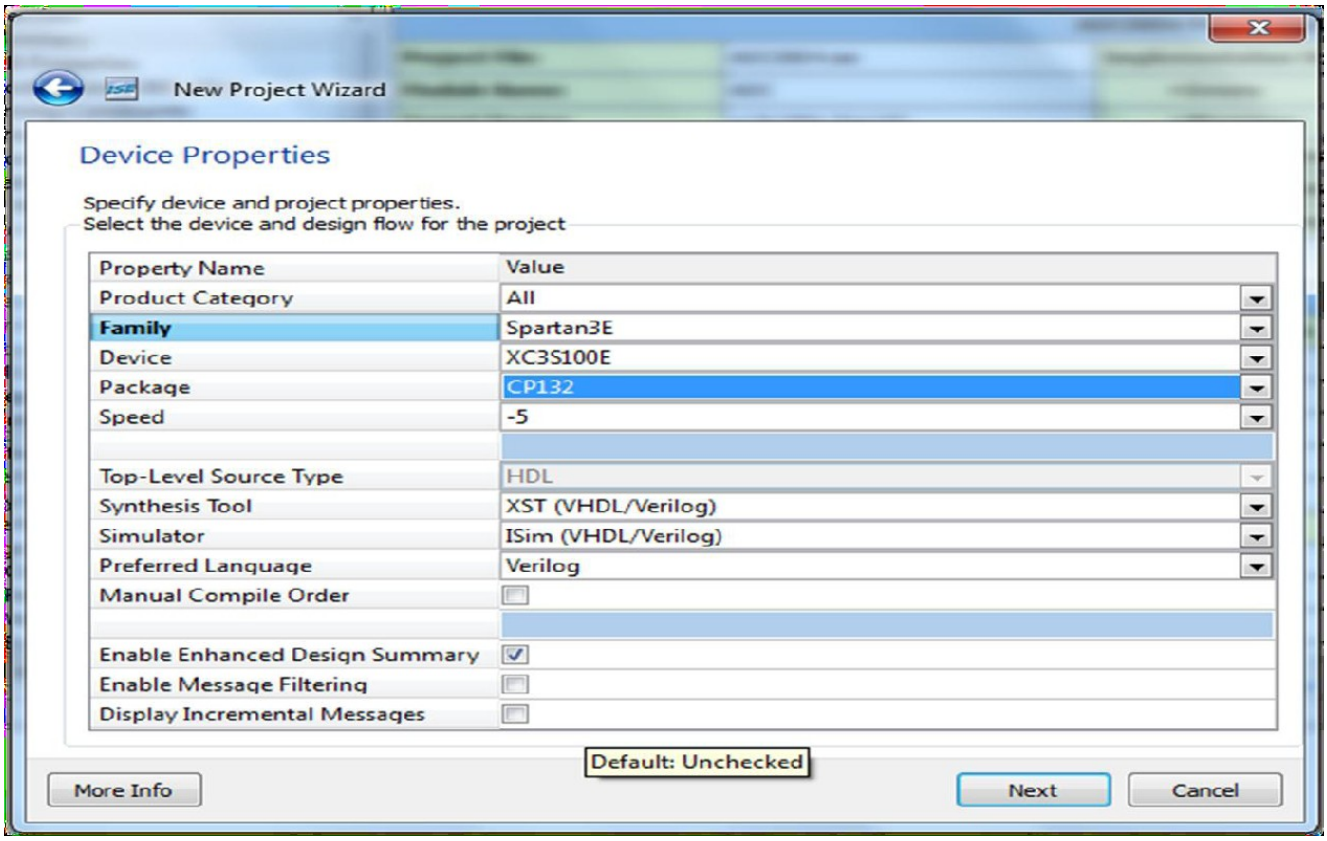
B. HOW TO CREATE AND UPLOAD PROGRAM INTO PROJECT BOARD

Design flow for Xilinx ISE series software
Step 1: open Xilinx ISE design suite 11

Step 2 : Create new project
1. To create a new project, open Project Navigator either from the Desktop shortcut icon or by selecting **Start > Programs > Xilinx ISE Design Suite 11 > ISE > Project Navigator**.
2. In Project Navigator, select the New Project option from the Getting Started menu (or by selecting Select File > New Project).
3. This brings up a Dialog box where you can enter the desired project name and project location. You should choose a meaningful name for easy reference. In this tutorial, we call this project “Test” and save it in a local directory. You can place comments for your project in the Description text box. We use HDL for our top-level source type in this tutorial.



4. The next step is to select the proper Family, Device, and Package for your project. This depends on the chip you are targeting for this project. The appropriate settings for a project suited for the BASYS 2 board are as follows:



Once the appropriate settings have been entered, click next. Above device properties are for CPLD.

For CPLDF, select

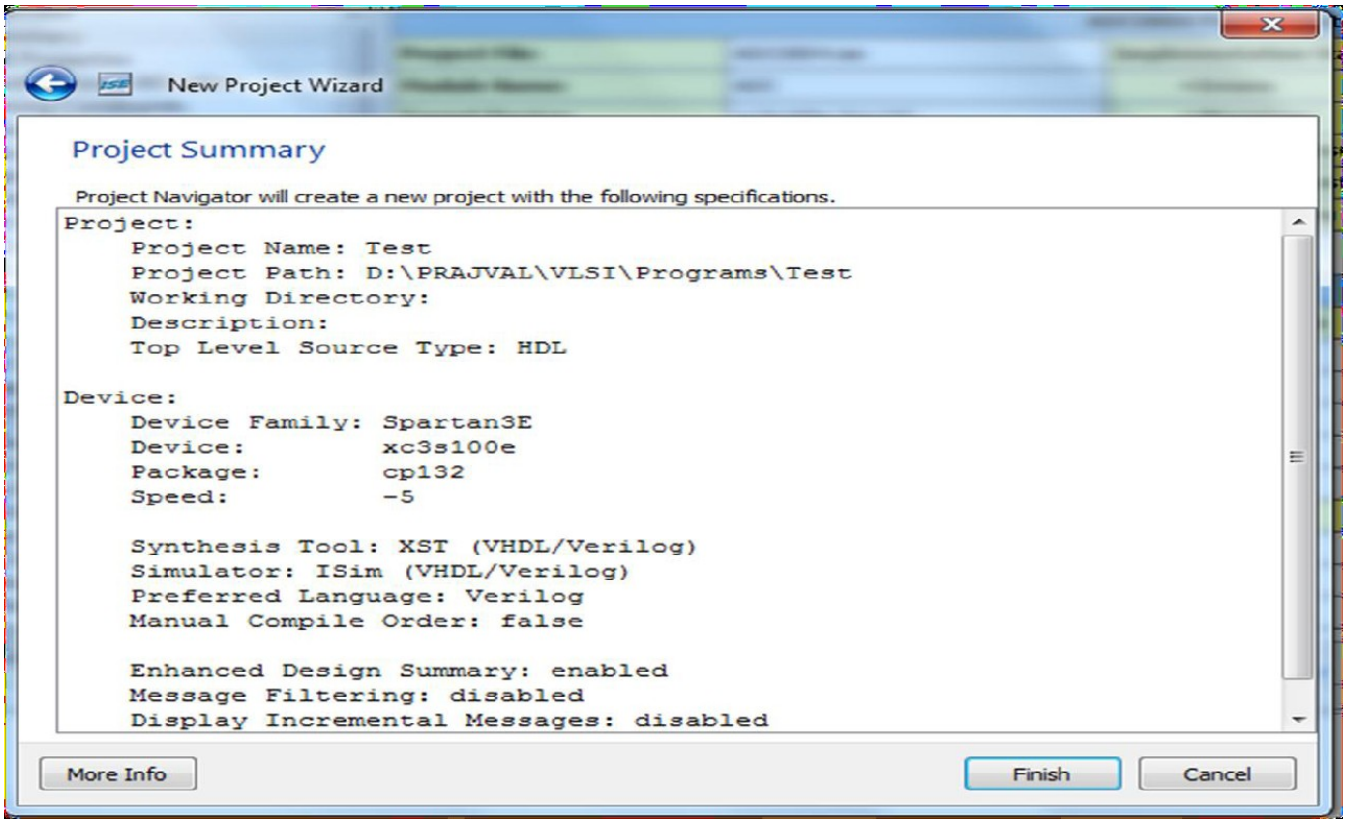
1) **Family : XC9500XL CPLD**
2) **Device : XC9572XL**
3) **Package : TQ100**
4) **Speed : -10**
5) **Preferred Language : VHDL**

5. The next two dialog boxes give you the option of adding new or existing source files to your project. Since we will fulfill these steps later, click next without adding any source files.

1. Before the new project is created, the New Project Wizard gives you a project summary consisting of the selected specifications you have chosen for the project.

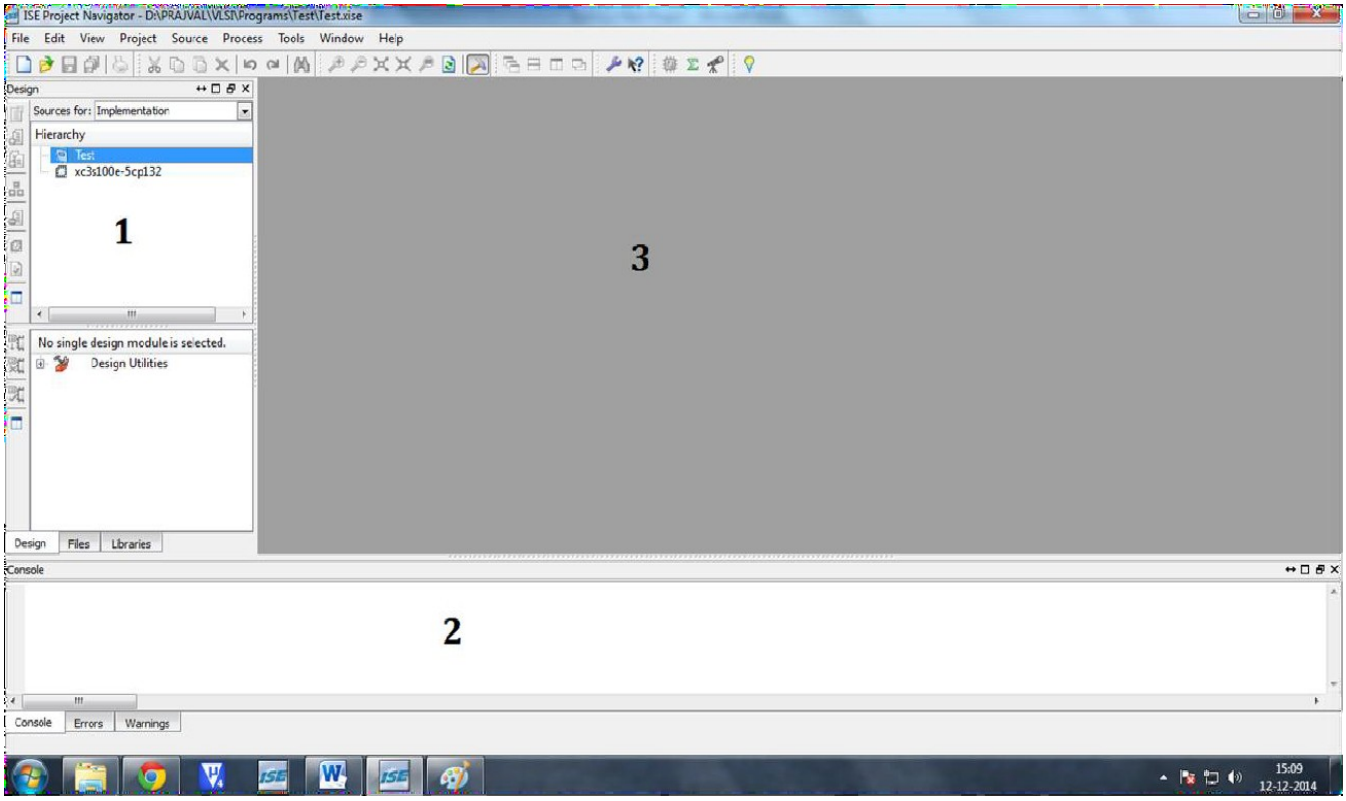
Make sure all settings are correct before clicking Finish to end the New Project Wizard.

2.



b) Project Navigator Overview

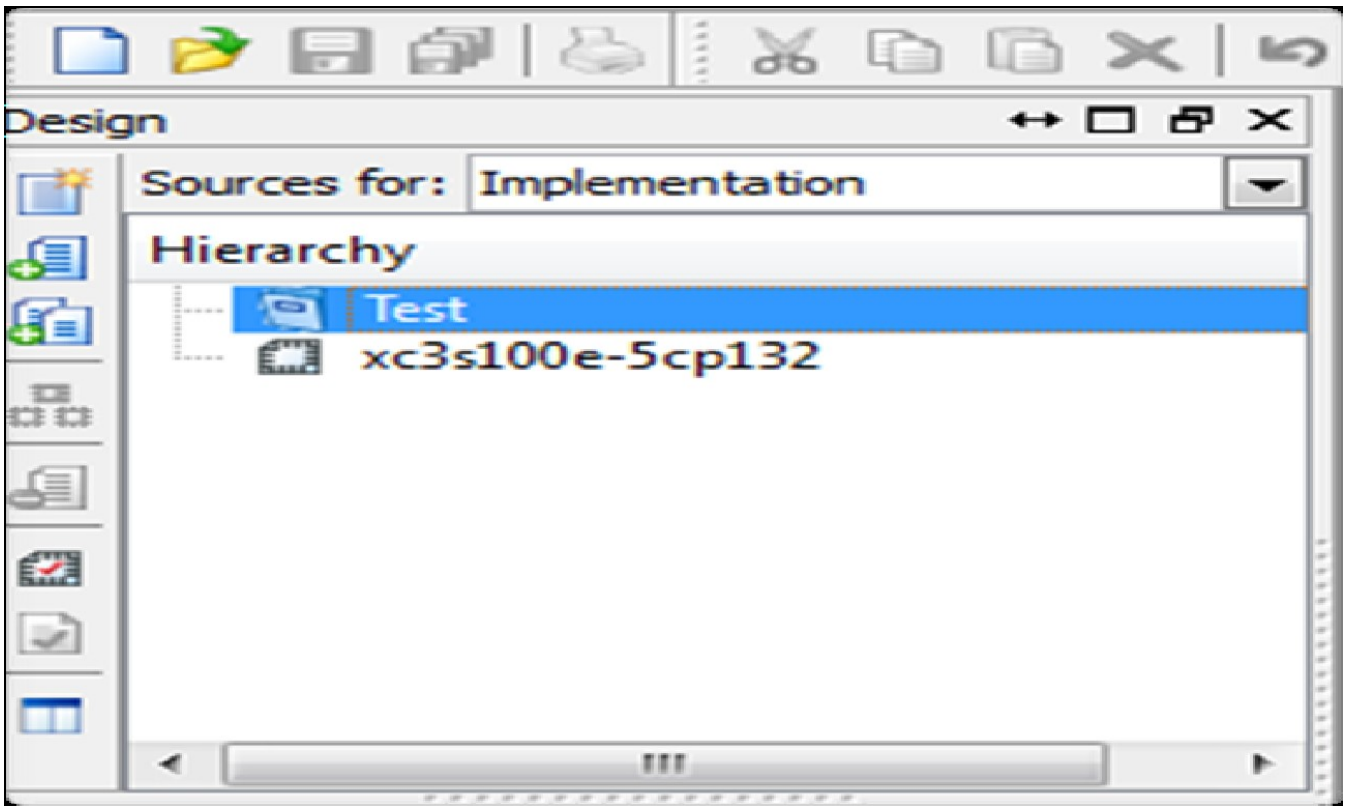
Once the new project has been created, ISE opens the project in Project Navigator. Click the Design tab to show the Design panel and click the Console tab to show the Console panel.



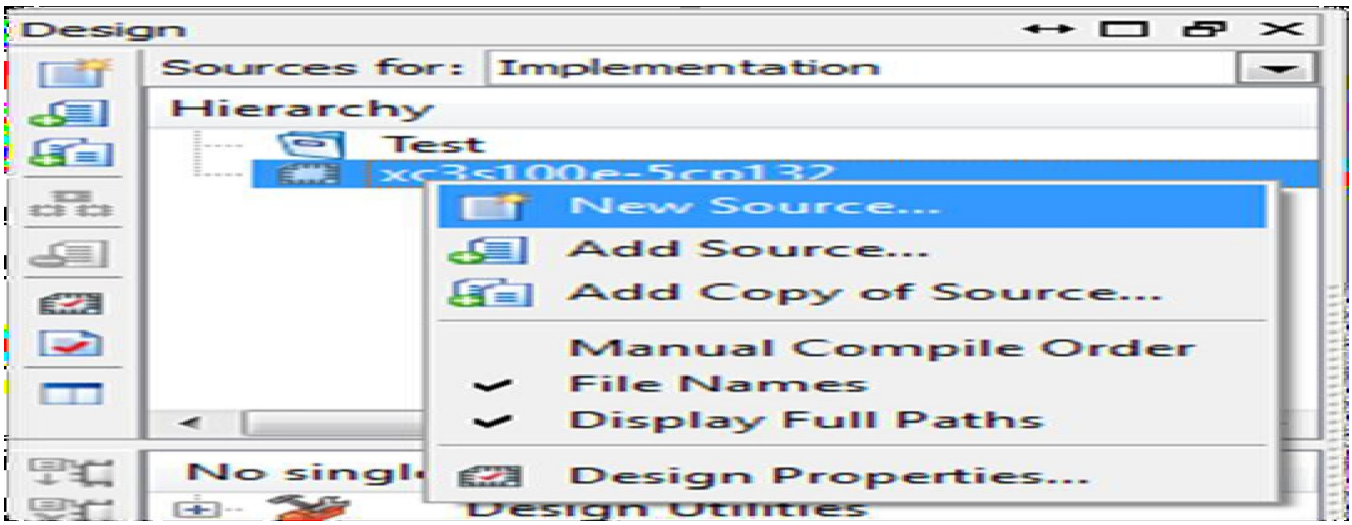
The Design panel (1) contains two windows: a Sources window that displays all source files associated with the current design and a Process window that displays all available processes that can be run on a selected source file.
The Console panel (2) displays status messages including error and warning messages.
The HDL editor window (3) displays source code from files selected in the Design panel.

c) Adding New Source Files

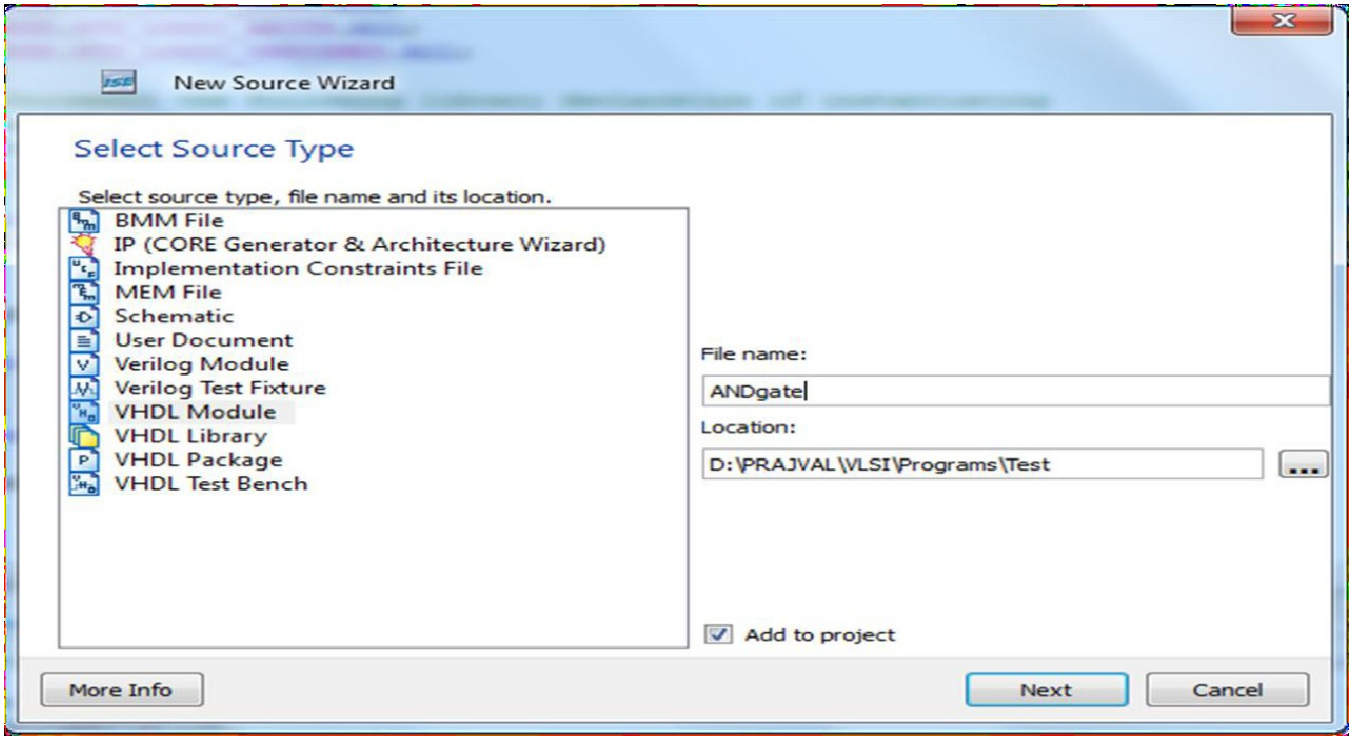
1. Once the new project is created, two sources are listed under sources in the Design panel: the Project file name and the Device targeted for design.



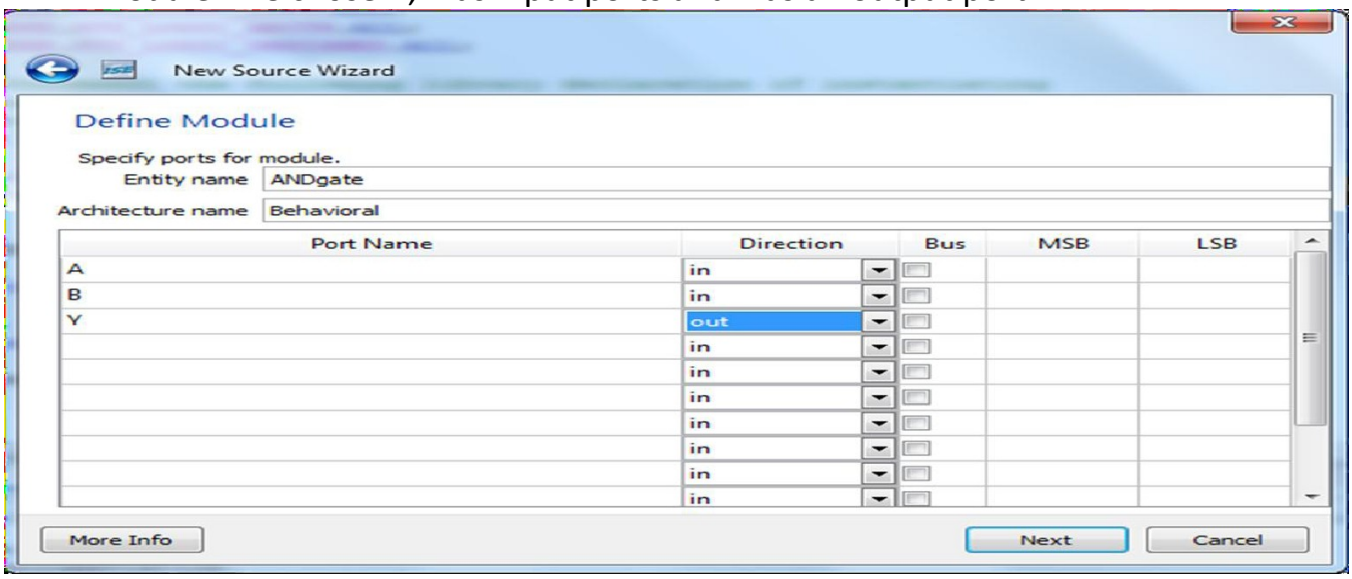
2. You can add a new or existing source file to the project. To do this, right-click on the target device and select one of the three options for adding source files.



3. In this tutorial, we will create a new source file, so select New Source from the list. This starts the New Source Wizard, which prompts you for the Source type and file name. Select VHDL module and give it a meaningful name (we name it ANDgate).



4. When you click next you have the option of defining top-level ports for the new VHDL module. We chose A, B as input ports and Y as an output port.



Click next and then finish completing the VHDL source file creation.

d) HDL Editor Window

Once you have created the new VHDL file, the HDL Editor Window displays the ANDgate.vhd source code.

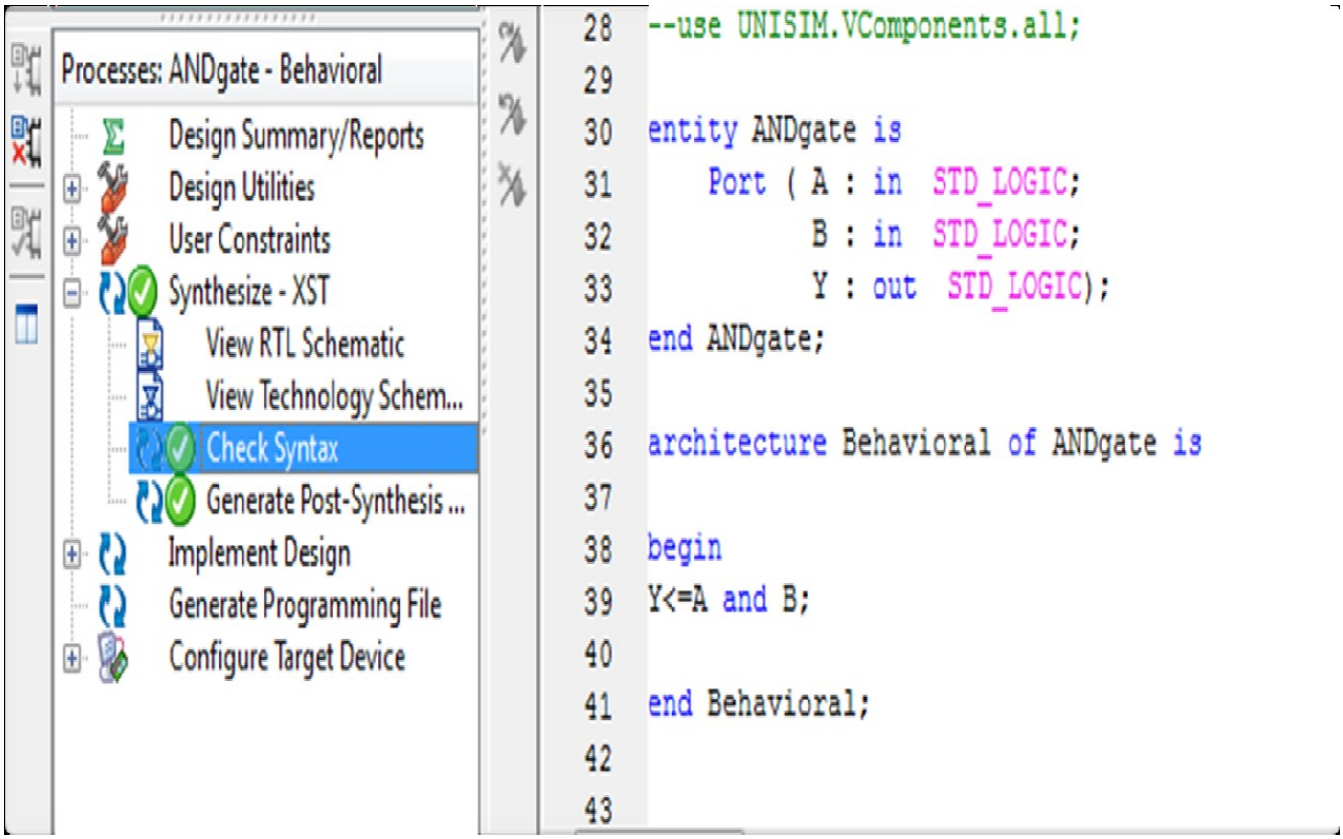
```
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  ---- Uncomment the following library declaration if using
26  ---- any Xilinx primitives in this code.
27  --library UNISIM;
28  --use UNISIM.VComponents.all;
29
30  entity ANDgate is
31      Port ( A : in  STD_LOGIC;
32             B : in  STD_LOGIC;
33             Y : out  STD_LOGIC);
34  end ANDgate;
35
36  architecture Behavioral of ANDgate is
37
38  begin
39
40
41  end Behavioral;
```

ISE automatically generate lines of code in the file to get you started with circuit development. This generated code includes:

- library definitions
- An entity statement
- An architecture statement with begin and end statements included
- A comment block template for documentation.

The actual behavioral or structural description of the given circuit is to be placed between the “begin” and “end Behavioral” statements in the file. Between these statements, you can define any VHDL circuit you wish. In this tutorial we use a simple combinational logic example, and then show how it can be used as a structural component in another VHDL module. We start with the basic logic equation: Y<= (A·B).

The following figure shows the implementation & synthesis:

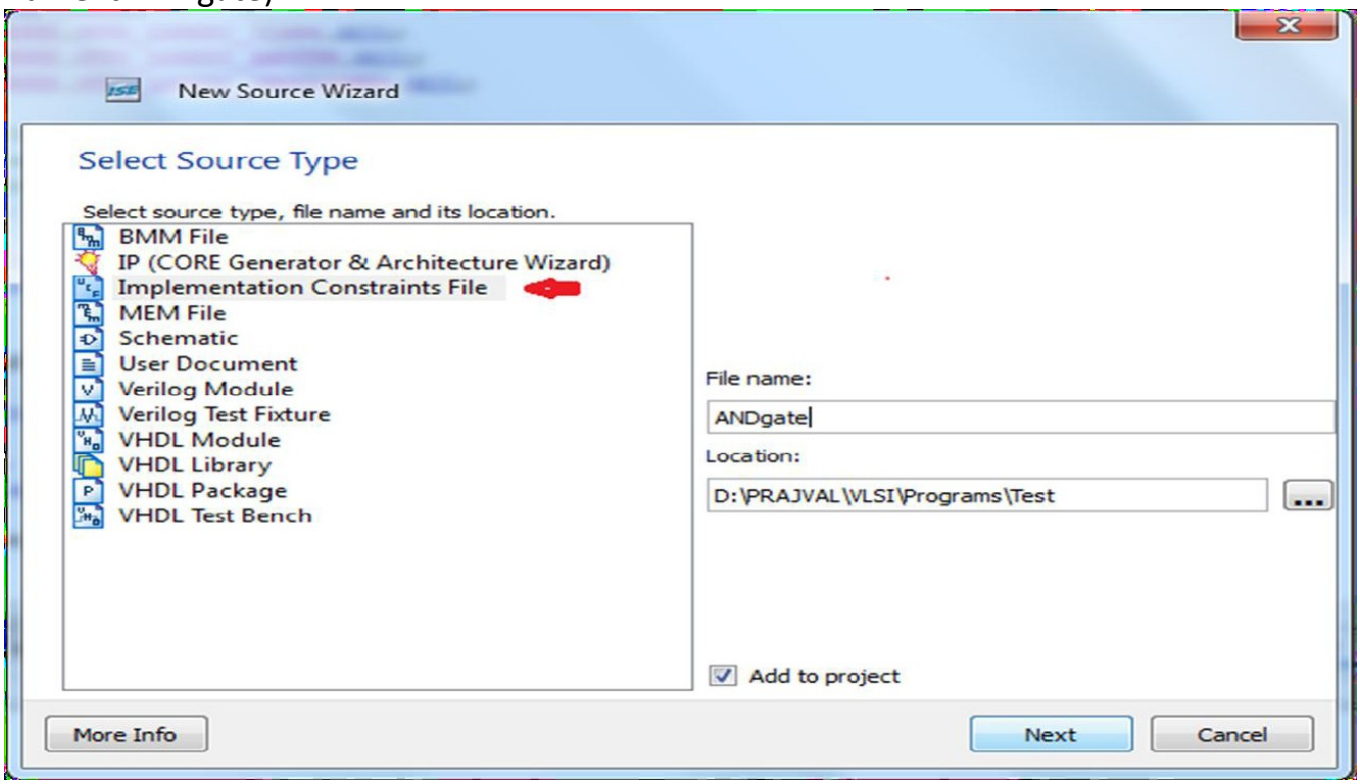


If there are any syntax errors in the source file, ISE can help you find them. For example, the parentheses around the A and B inputs are crucial to this implementation; without them, ISE would return an error during synthesis.

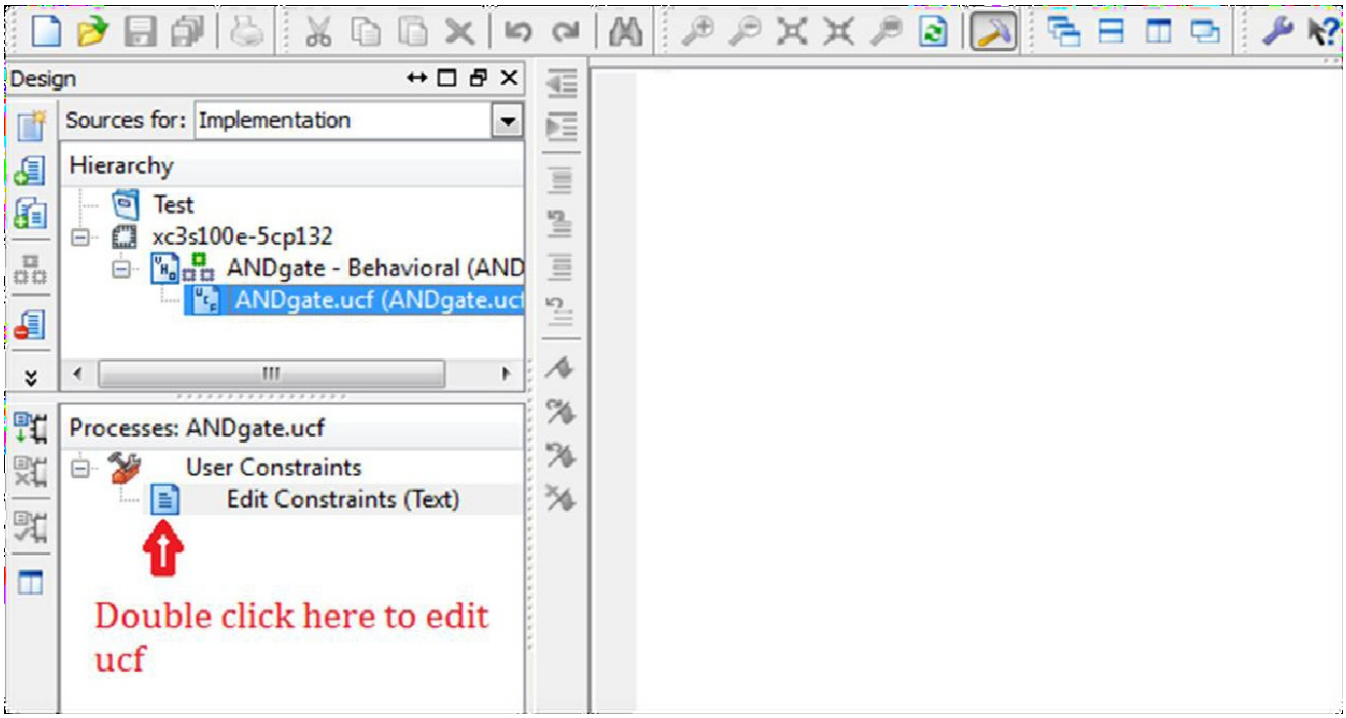
e) UCF File Creation

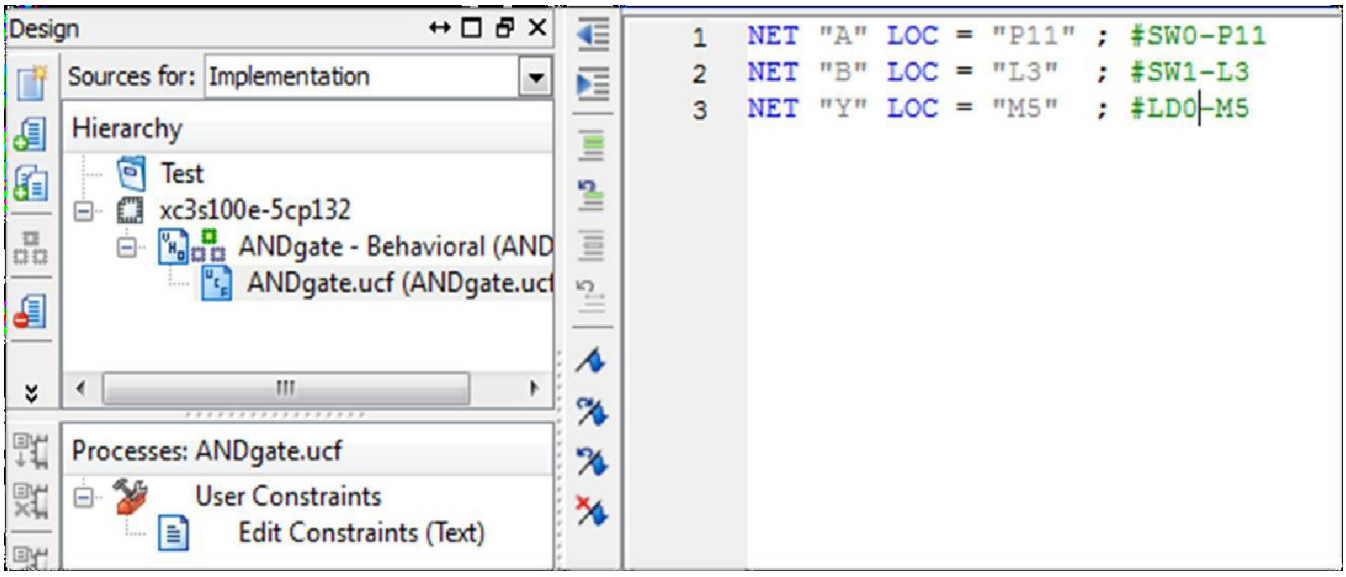
The Xilinx tools use a User Constraints File (.ucf file) to define user constraints like physical pin to circuit net mappings. This is sometimes referred to as an Implementation Constraints File. The .ucf file can be modified inside ISE using a text editor.

To add a .ucf file to your design, go to the Sources window and right-click the source file that requires user constraints. Select the Add New Source option in the drop-down menu. The New Source Wizard prompts you for the Source type and file name. Select Implementation Constraints File and give it a meaningful name (we'll name it ANDgate).



To edit the .ucf file, select it in the sources window, expand the User Constraints option in the Processes window below, and double-click the Edit Constraints (Text) option. A blank text editor appears.



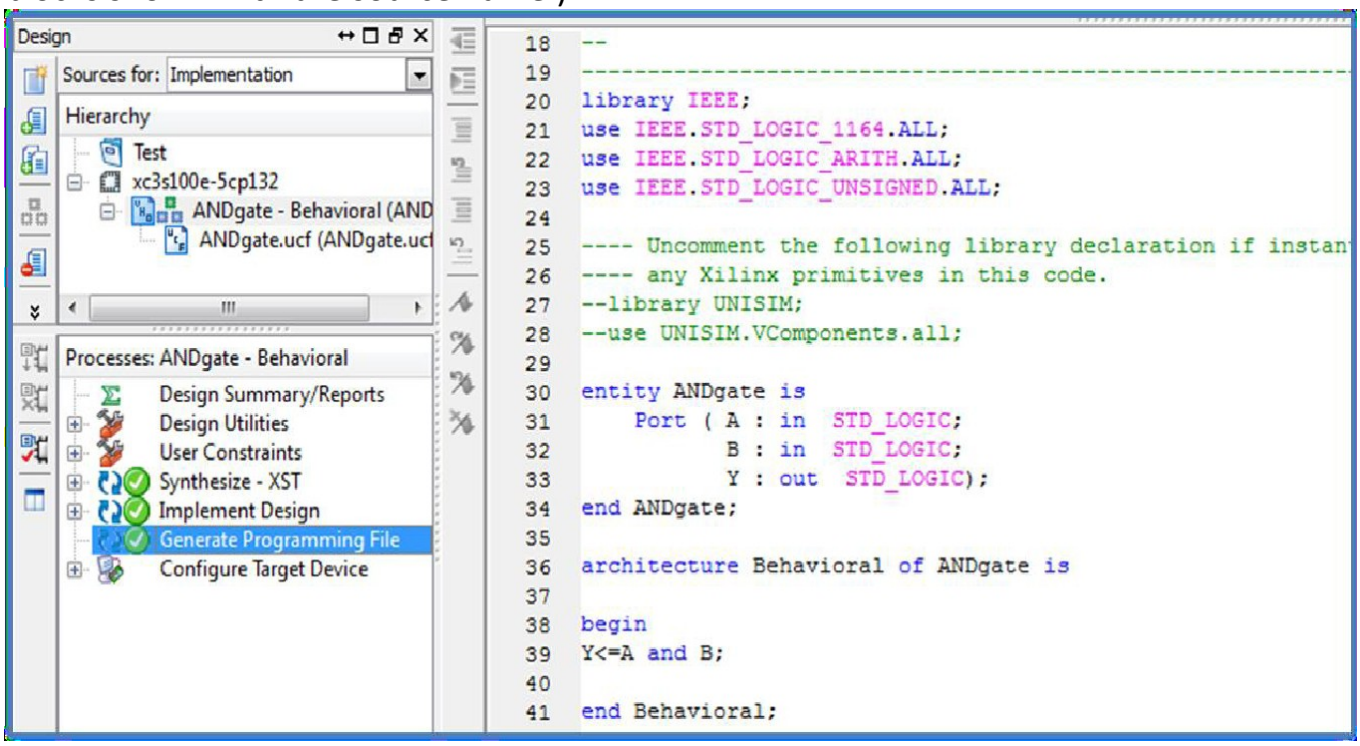


To associate a physical pin with a given net name, type: NET “net name” LOC =”XXX”; on a line in the .ucf file. In the statement, “net name” (quotes included) is the name of the net to attach to pin number XXX (quotes included).

For our example project, the two inputs are assigned to switches 0 through 1 and the output is assigned to LD0 on the BASYS2 board. The finished .ucf file is as follows:

f) Programming File Generation

Now we are ready to create a programming file (.bit) for the BASYS2 FPGA.
Go to the Sources window and select the top-level module (indicated by the three blocks shown with the source name.)



Now go to the Processes window where there are three particular processes in a row:

1. Synthesize – XST
2. Implement Design
3. Generate Programming file

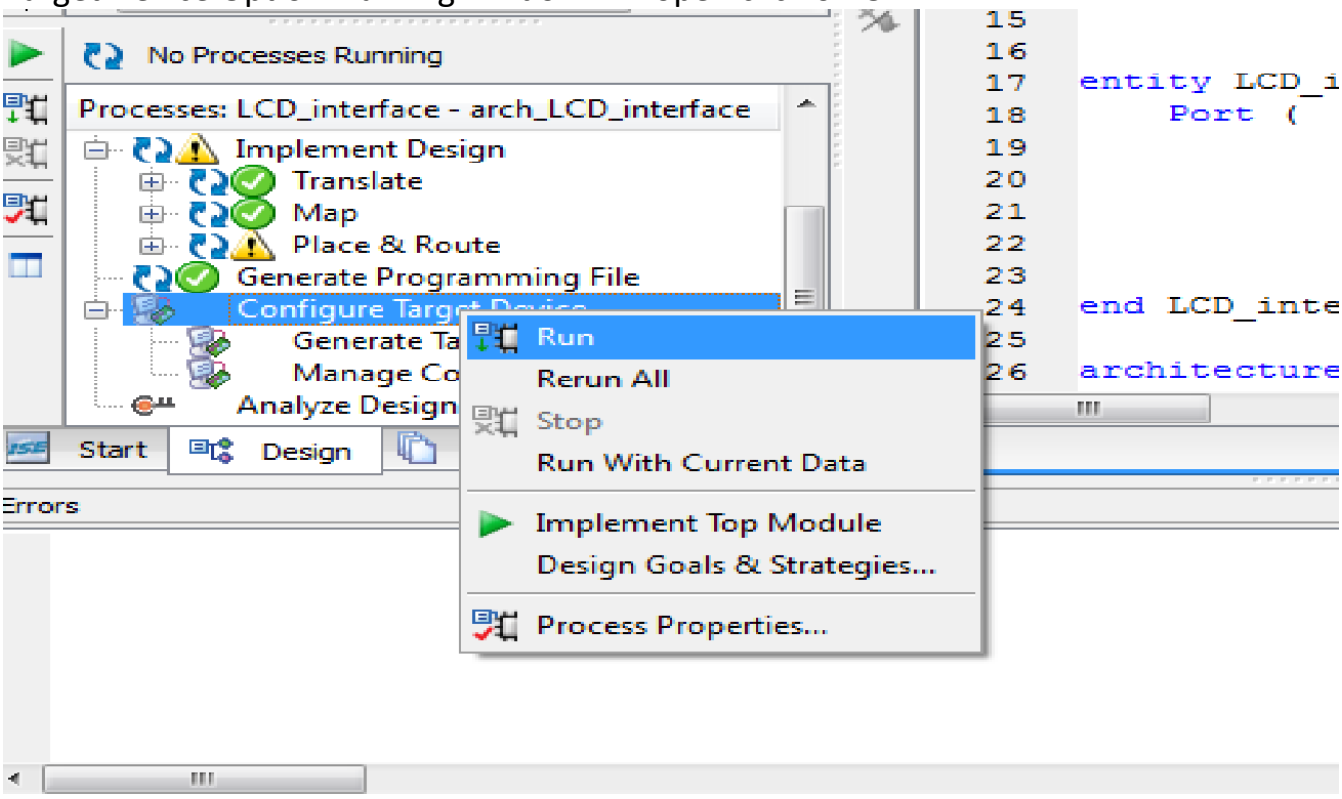
Run the synthesis process by both double-clicking on Synthesize or left clicking and selecting the run option. This process analyzes the circuit you have created, checking for valid connections, syntax, and structure, to verify that the circuit is valid and synthesizable.

If the Synthesize process does not return any errors, you can move on and run the Implement Design process. This process uses various algorithms to map out the

digital circuit and then creates place and route information so that it can be placed on the physical FPGA.

g) Programming Board

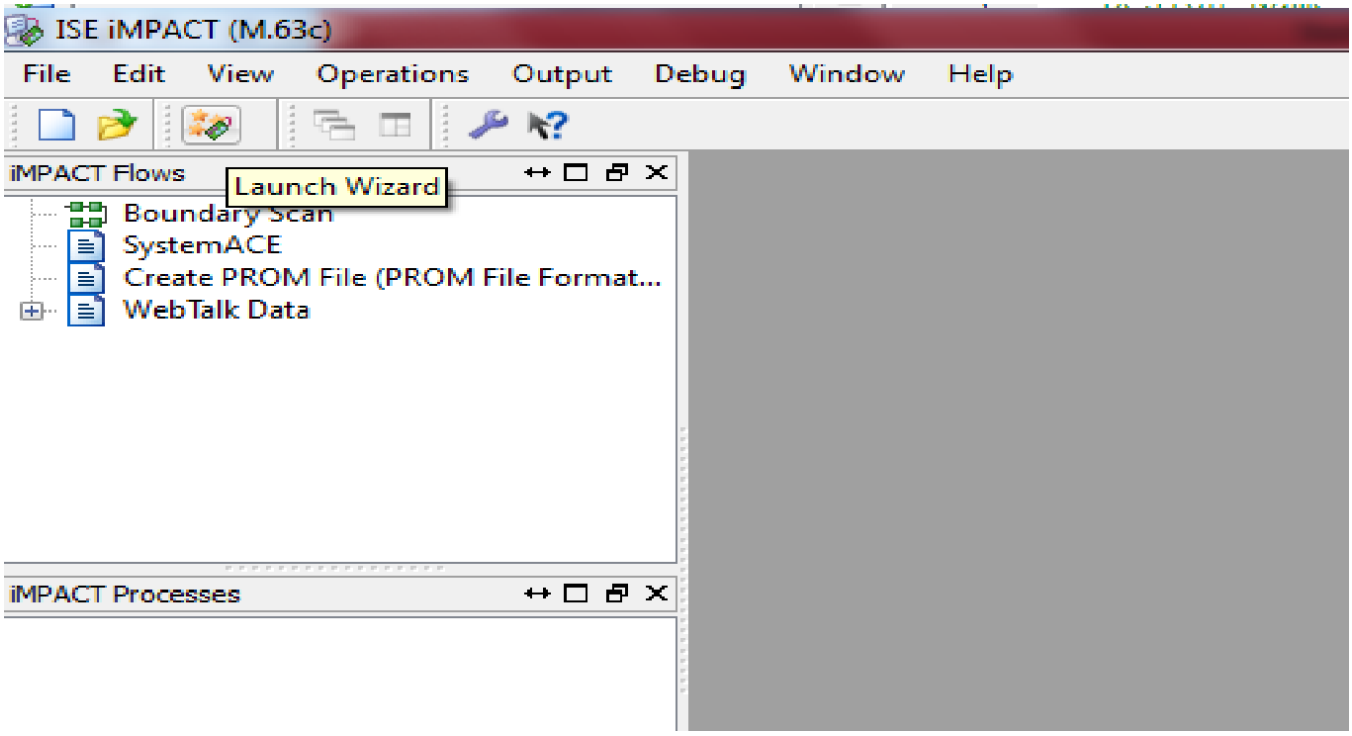
1. After generating programming file double click or Right click and Run on Configure Target Device Option warning window will open click on OK.



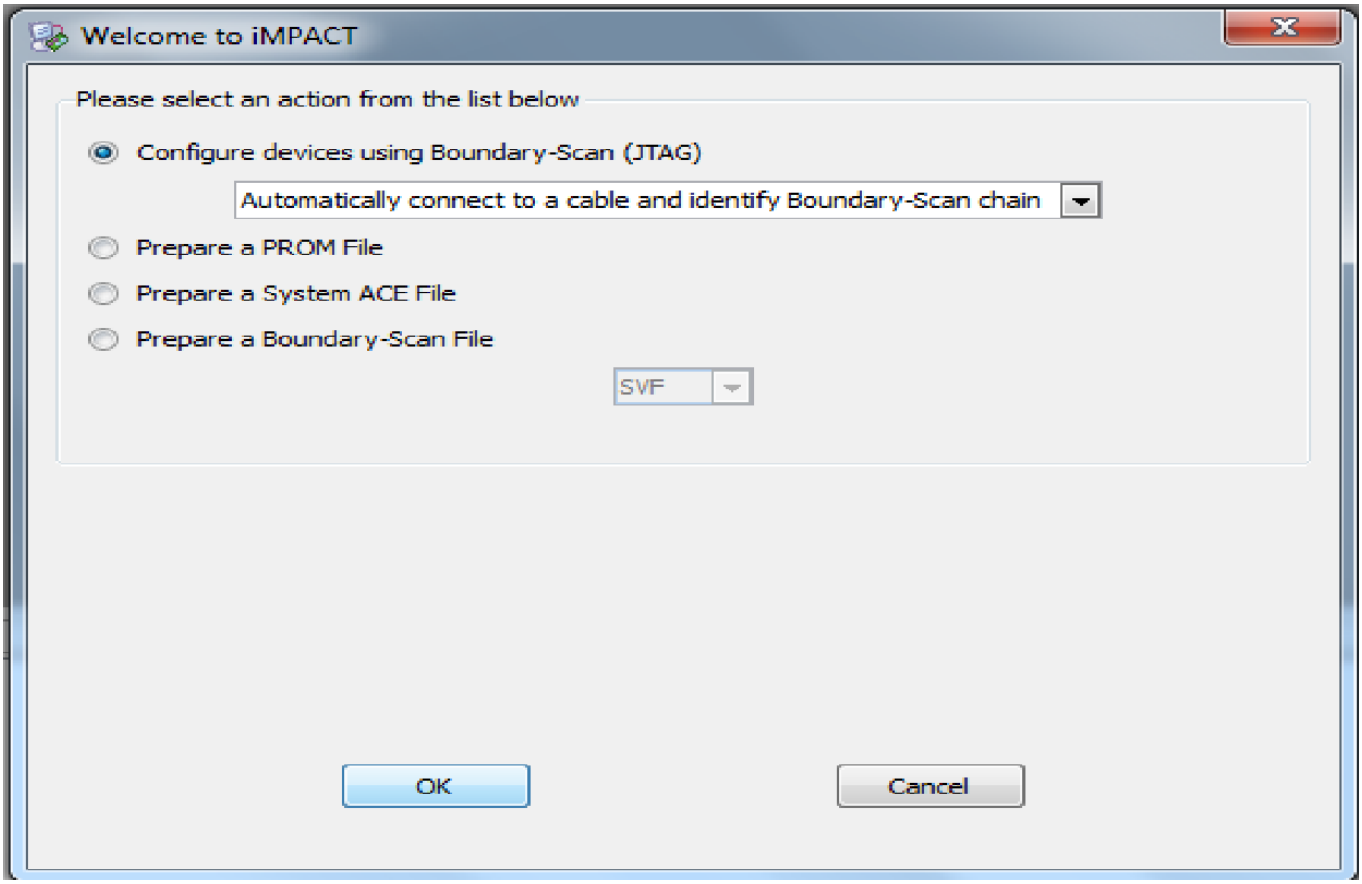
2. After this step ISE iMPAC (M.63c) window will open.

3. Now make connection between your PC and FPGA board via Xilinx Platform Cable USB by connecting 10 pin FRC cable to JTAG Connector of FPGA board.

4. Click on Launch Wizard option.



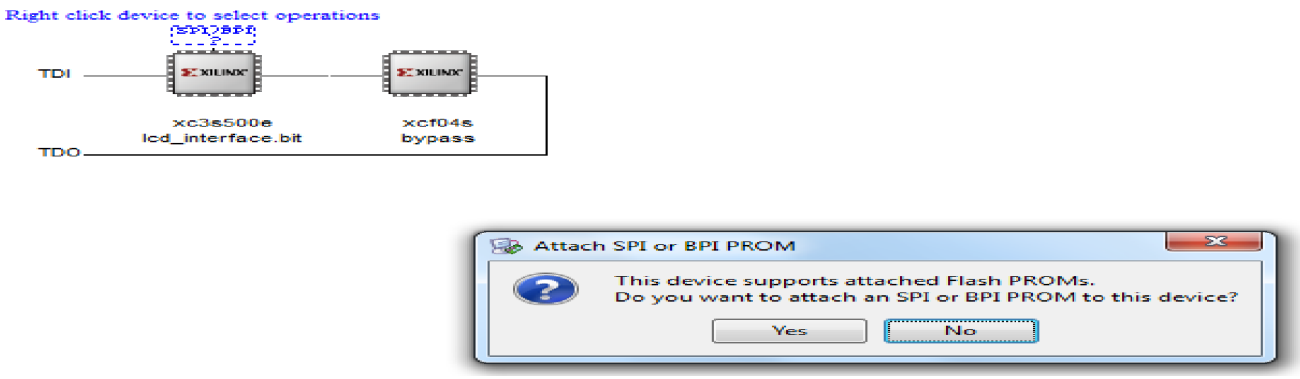
5. After click welcome window will open click OK.



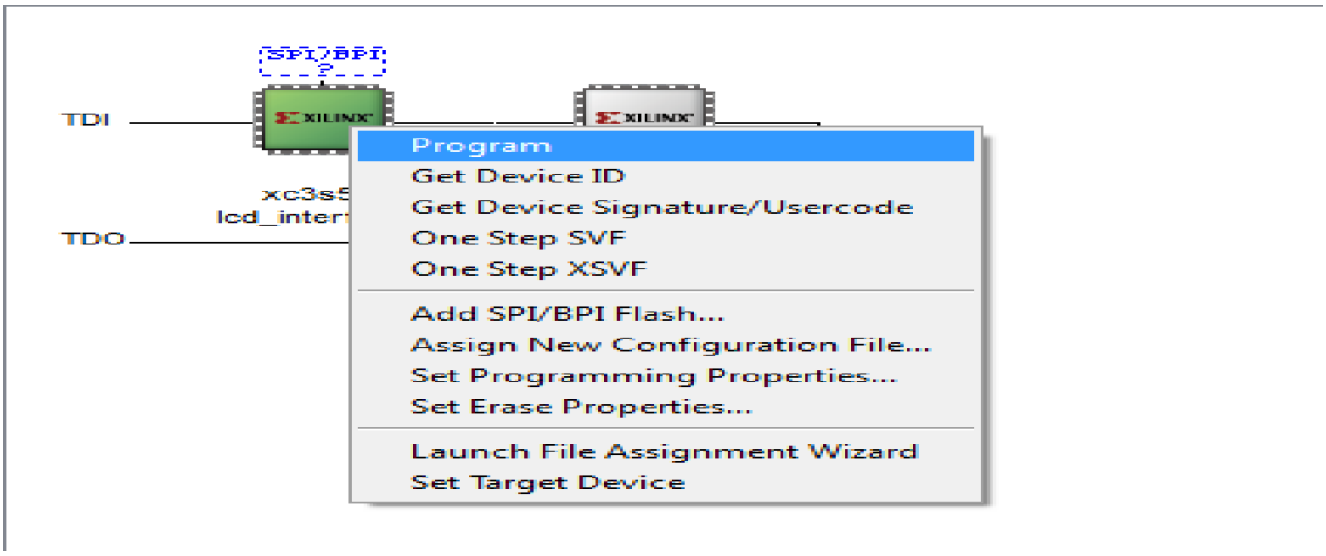
6. If everything ok 'Identify Succeeded' will display. At same time two "Assign New Configuration File" window will open one by one in this window click on Cancel after this "window Device Programming Properties" window will display click on cancel.

7. Now double click on device and Browse for .bit file which you have to program.

8. After opening your .bit file one warning window will open click on **NO.**



9. Now Right click on device and select program option after clicking new window will open click OK. If everything ok 'Program Succeeded' will display**.**



C. PIN CONFIGURATION:

J1:

VCC	GND
55	54
53	52
50	49
42	41

J2:

VCC	GND
39	40
37	36
35	32
33	30

J3:

VCC	GND
29	28
25	20
18	17

16	**15**

J4:

VCC	GND
14	**13**
12	**11**
10	**8**
6	**1**

J5:

VCC	GND
96	**95**
94	**93**
92	**91**
90	**89**

J6:

VCC	GND
87	**86**
85	**82**
81	**79**
78	**77**

J7:

VCC	GND
56	**58**
59	**60**
61	**63**
64	**65**

J8:

VCC	GND
66	**67**
68	**70**
71	**72**
74	**76**