LABORATORY EXERCISE **#1 – FPGA INTRO**

Objective:

This exercise will introduce you to using the TLL5000 FPGA development platform. The Xilinx ISE 12.4 development environment will be used for the FPGA design. Our lab computers have the full licensed version of ISE, but you can download the free web edition version of the ISE onto your personal computer if you want to work on your FPGA design outside of the lab. The TLL Hardware Manager software will be used to power up the TLL5000 board.

Required Materials:

To complete this exercise, you will need the TLL5000 development board, power supply, Xilinx Platform USB Cable, and two (2) USB cables. If the TLL6219 (ARM926) card and/or the prototyping mezzanine cards are installed on the TLL5000 mezzanine positions, just leave them there.

Warnings:

Warning: Always handle the development board in a static-safe manner to prevent ESD damage. You should always touch the metal frame before touching anything on the development board.
Warning: Be careful not to drop or lay probes, wires, or other conductive objects on the FPGA board. Be especially careful of the scope ground lead.
Warning: NEVER program the FPGA unless a valid UCF file is used in the project, and pin assignments are verified. Failure to do so may result in contention and destroy the FPGA, other devices, or both.

Set-up:

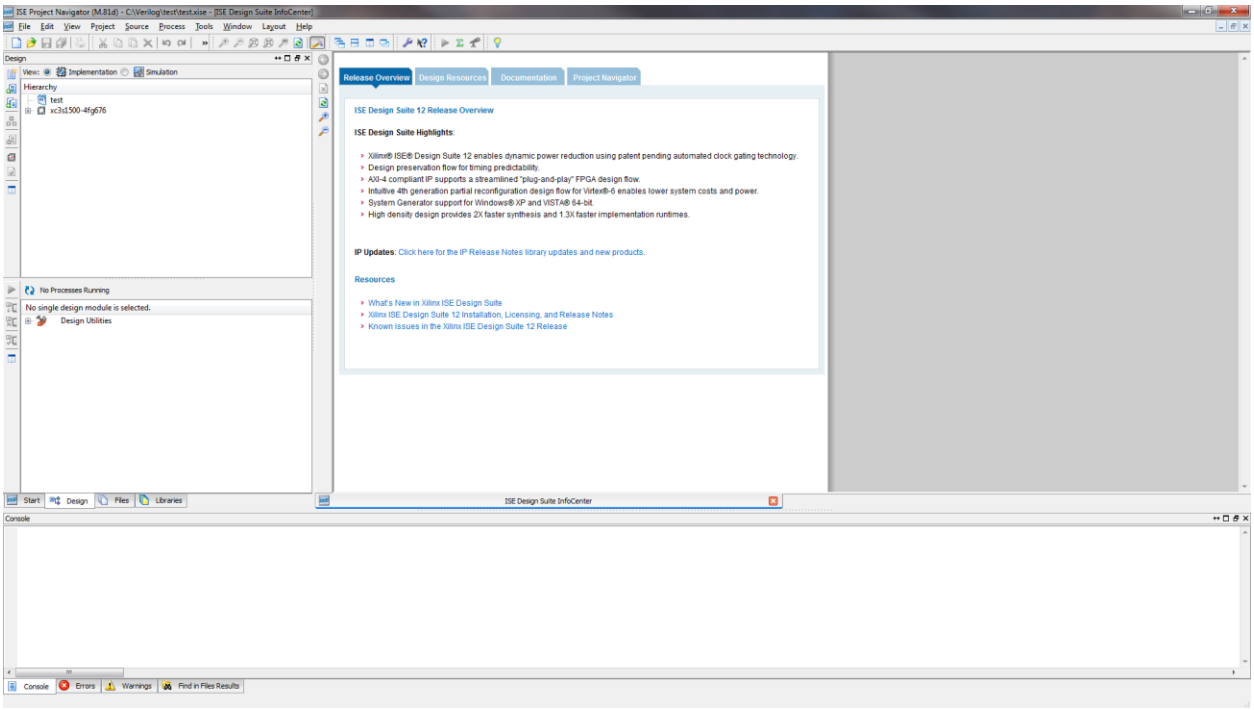
- Be careful to observe safe handling and ESD precautions, place the TLL5000 board on the bench. Refer Chapters 1-2 of the TLL5000 Getting Started manual available on the course web site (Lab Equipment Documentation folder).

- There are some small differences between the ISE 9.2 used in the Getting Started screen captures and the ISE 12.4 we are using. They should be fairly obvious, but if in doubt, just ask the lab instructor. Relevant ISE 12.4 instructions are included in this document, so use the Getting Started manual as a more detailed reference.

Caution: Do NOT place any project source files (Verilog, UCF, etc.) in the root directory of the drive containing your project. I.e. if your project is located on the D: drive, make sure D:\ does not contain any files with the same name as any source file in your project. (Files in subdirectories are acceptable.) If you fail to follow this, the Xilinx synthesis engine may use the files in the root directory instead of the ones you actually included in the project, creating a very bizarre situation for you to debug.

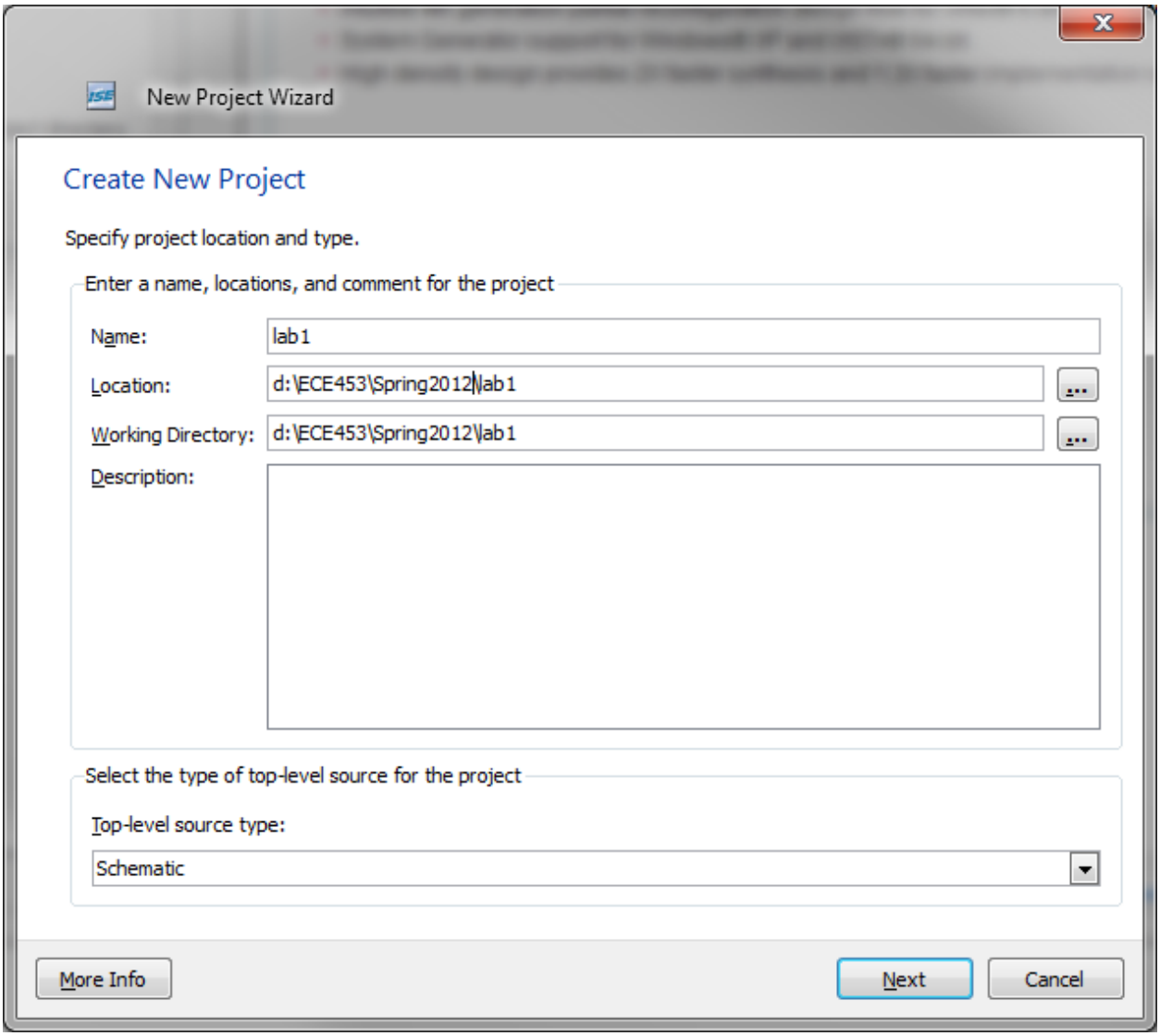
- Connect the USB cables for the TLL5000 and the Xilinx Platform USB Cable to the front panel USB connectors on your computer.

1. Run the ISE application by double clicking on the shortcut icon for the ISE tool or selecting it from your Programs menu.

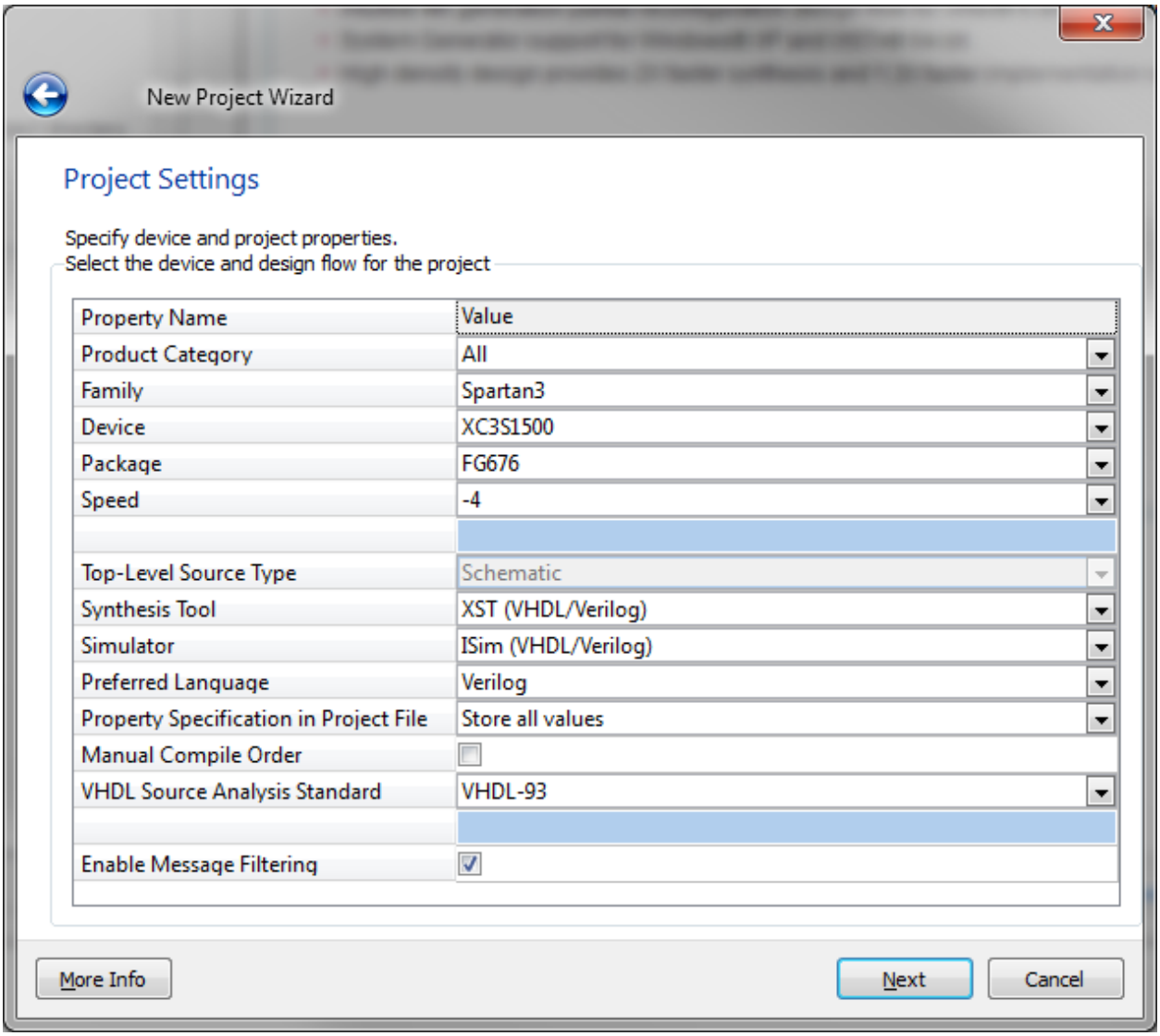


2. Create a new project under File→New Project

3. When the new project option is selected the following window is seen. In this window type a project name "lab1" and specify a location. Select "Schematic" as the Top-Level Source Type. And click "Next".

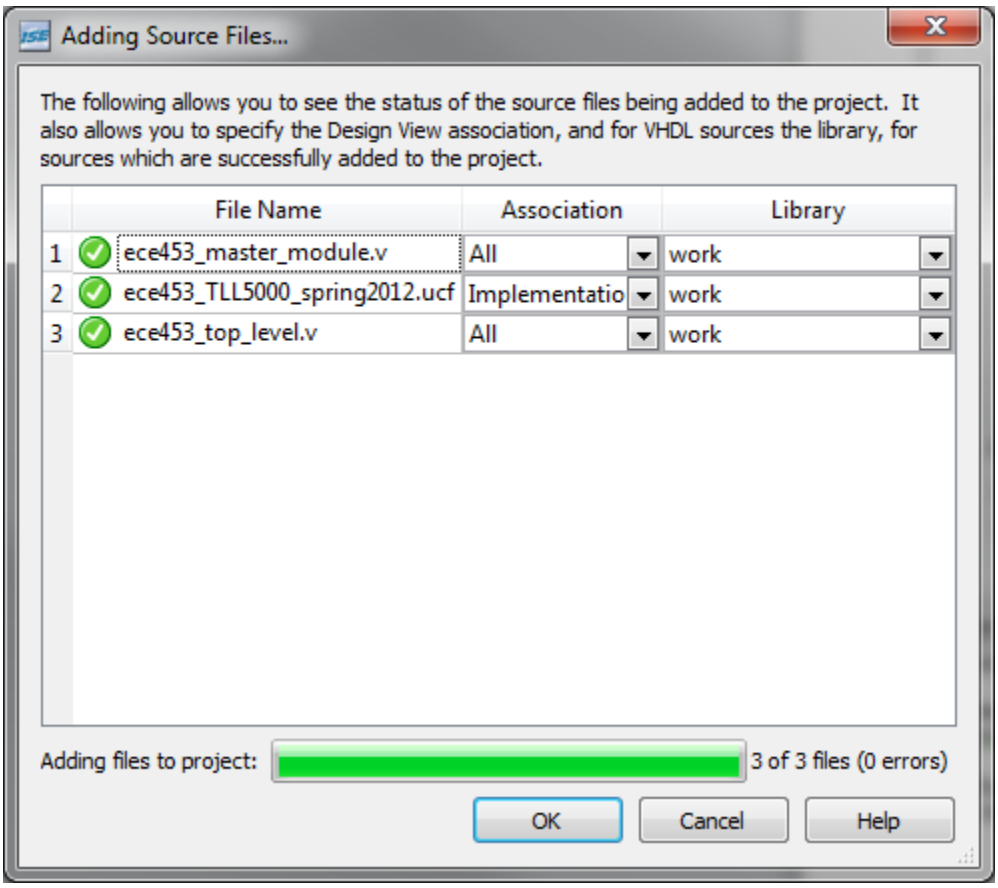


4. After these inputs the tool will go to the next window. Here select the target device and the tools that will be used in developing the design. Then click "Next".


New Project Wizard
Project Settings
Specify device and project properties.
Select the device and design flow for the project
Property Name
Value
Product Category
All
Family
Spartan3
Device
XC3S1500
Package
FG676
Speed
-4
Top-Level Source Type
Schematic
Synthesis Tool
XST (VHDL/Verilog)
Simulator
ISim (VHDL/Verilog)
Preferred Language
Verilog
Property Specification in Project File
Store all values
Manual Compile Order
VHDL Source Analysis Standard
VHDL-93
Enable Message Filtering
More Info
Next
Cancel

5. Add all 3 default ECE453 Course documents to the project by selecting

   Project→Add Source



CHECKPOINT: When you are done, ask the instructor to check your work before moving on.

Using Verilog with the Xilinx FPGA:

Our designs in the course will be done using Verilog. You will create a new project as before, but instead of adding a schematic file, you will start your project with several files downloaded from the course web page.

ece453_top_level.v – This module will always be used as the top-level module in every Verilog project for the course. **It is NOT to be altered in any way.** A good idea is to set it to read-only so it cannot be changed inadvertently. It defines all of the external connections to and from the FPGA, and instantiates a single module *ece453_master_module*. The other very important thing this module does is that it ensures that the FPGA pins connected to the transceivers on the prototyping adapter board, and the prototyping adapter board transceivers, are never driven so that they would contend with each other.

ece453_master_module.v – This module is the second level module where you can add your Verilog. In general, you will instantiate modules here that are defined in other files. For this exercise, you will need to instantiate just a single module. The default assignment code in *ece453_master_module* is used to put various ports in a safe state. If you are not using those ports, the code should be left as is. If you are using a port in your design, you should comment out the default assignments.

ece453_TLL5000_spring2012.ucf – This file defines all of the pin assignments for the TLL5000, and is to be included in every ISE project. **It is NOT to be altered in any way.**

6. After creating the project lab1 create a file lab1.v and add it to your project.

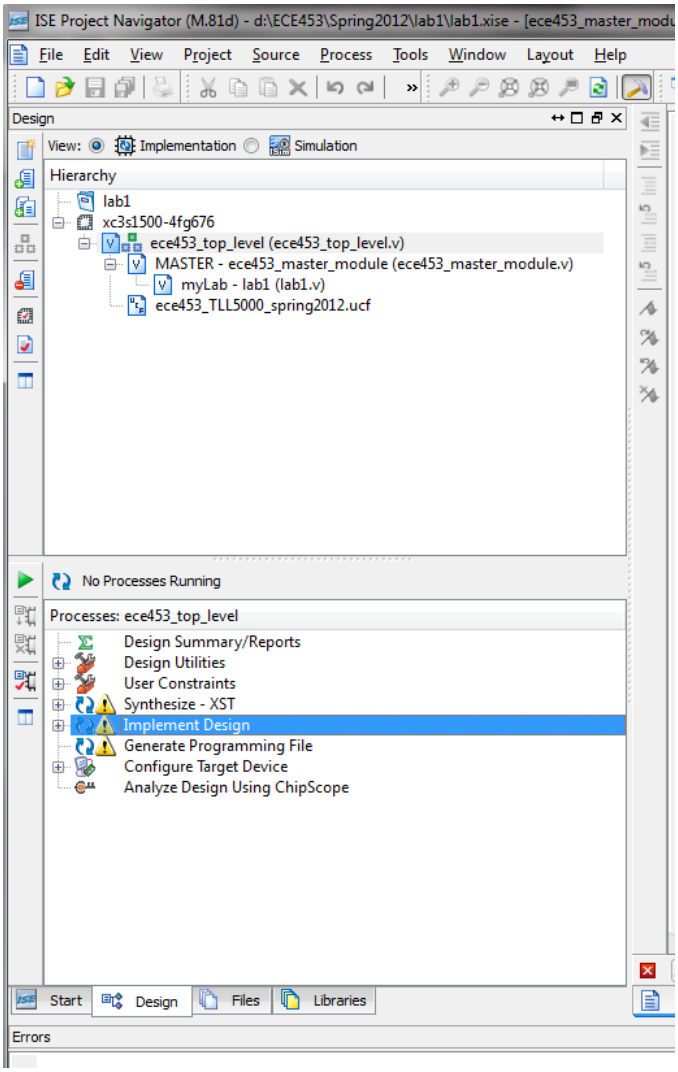
   Define a module with the following ports,

   - LED, output, vector 7:0
   - SW, input, vector 7:0
   - CLK, input (clock signal)
   - RST, input (active-high reset signal)

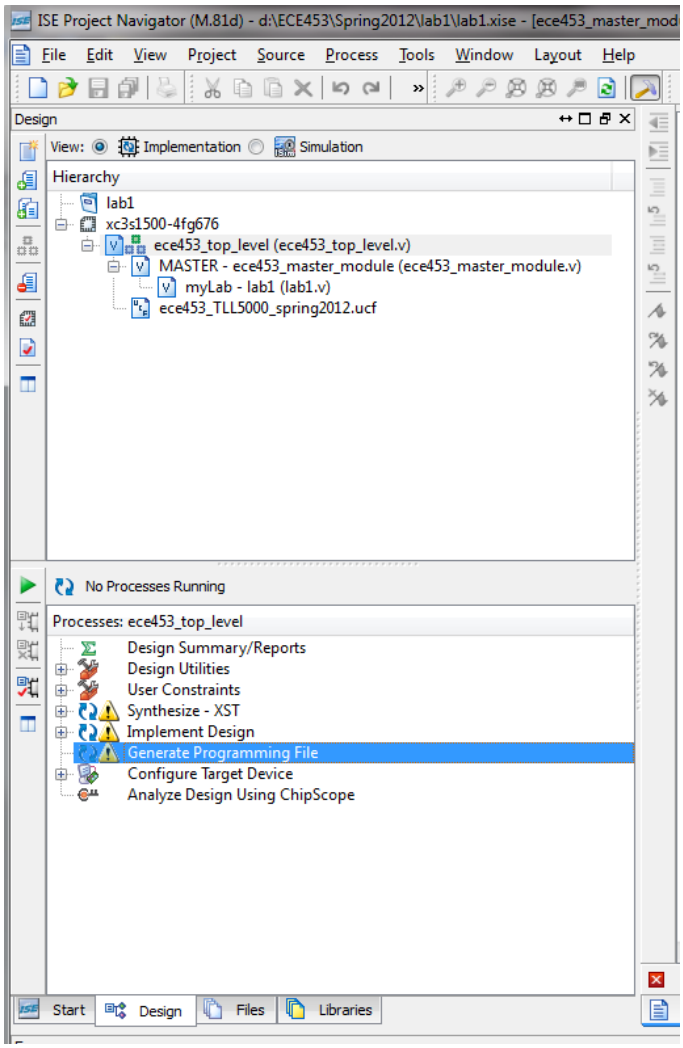
   In ***lab1.v***, add a single always block sensitive to posedge CLK and posedge RST. Make RST an asynchronous reset. Add code so that LED[7] displays the state of SW[7]. If SW[7] is a 1, LED[6:0] should show the state of SW[6:0]. If SW[7] is a 0, LED[6:0] should show the complement of SW[6:0].

   In ***ece453_master_module.v***, instantiate your module and connect it to the ***ece453_master_module*** ports SYS_CLK (24MHz clock), SYS_RST_N (active-low reset, so it needs to be complemented), DIP_SW (the DIP switches) and LED. Connect your ports by name, not by position. (This should be standard coding practice throughout the course.

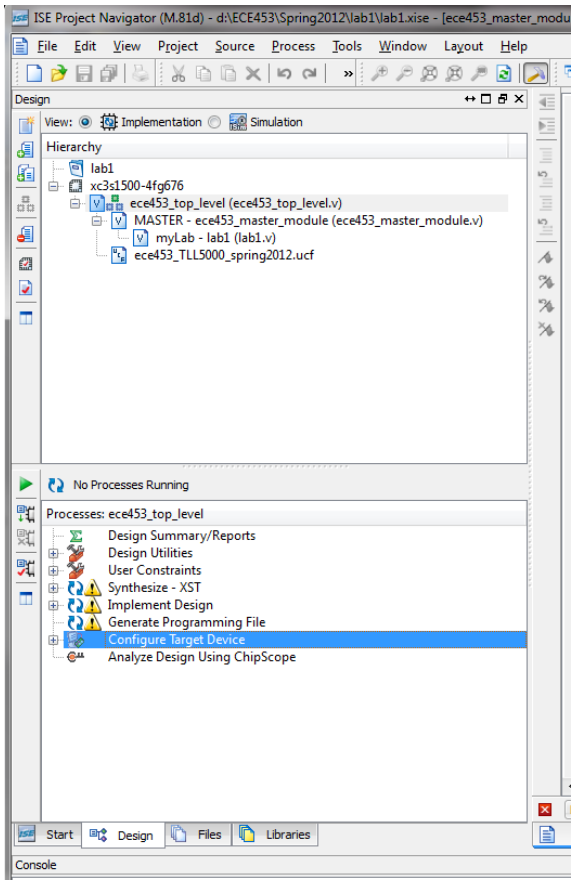
7. In the Processes pane, run **Implement Design**. Correct any errors



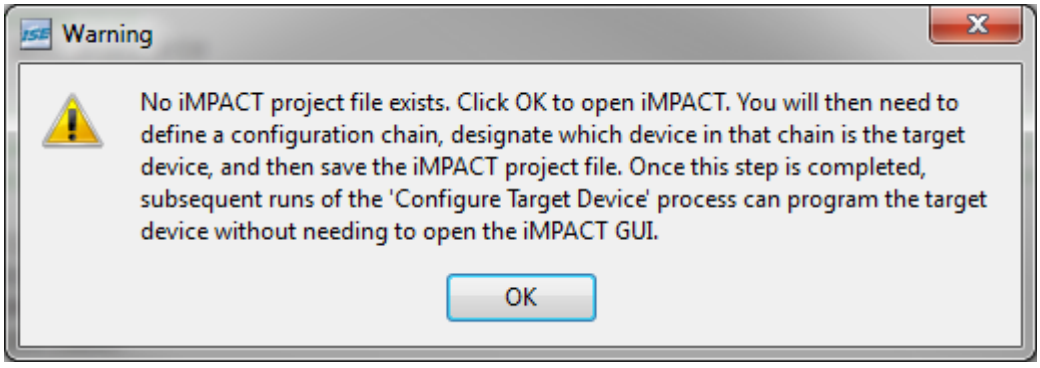
8. Generate a programming file



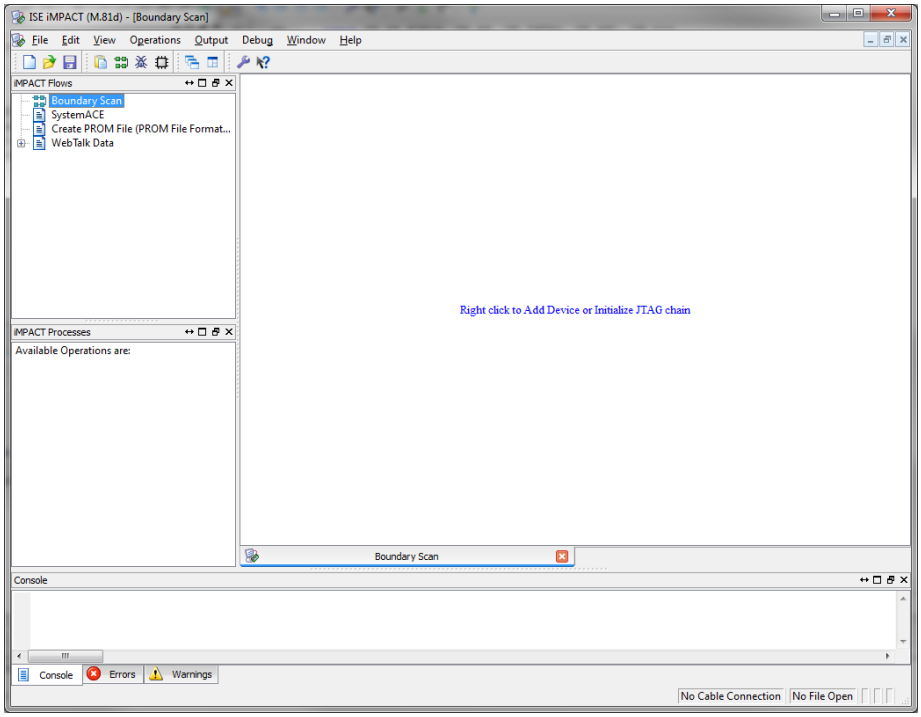
9. Download and verify your circuit. Double Click ‘Configure Target Device’.



10. Click ‘OK’ when the following warning is displayed.

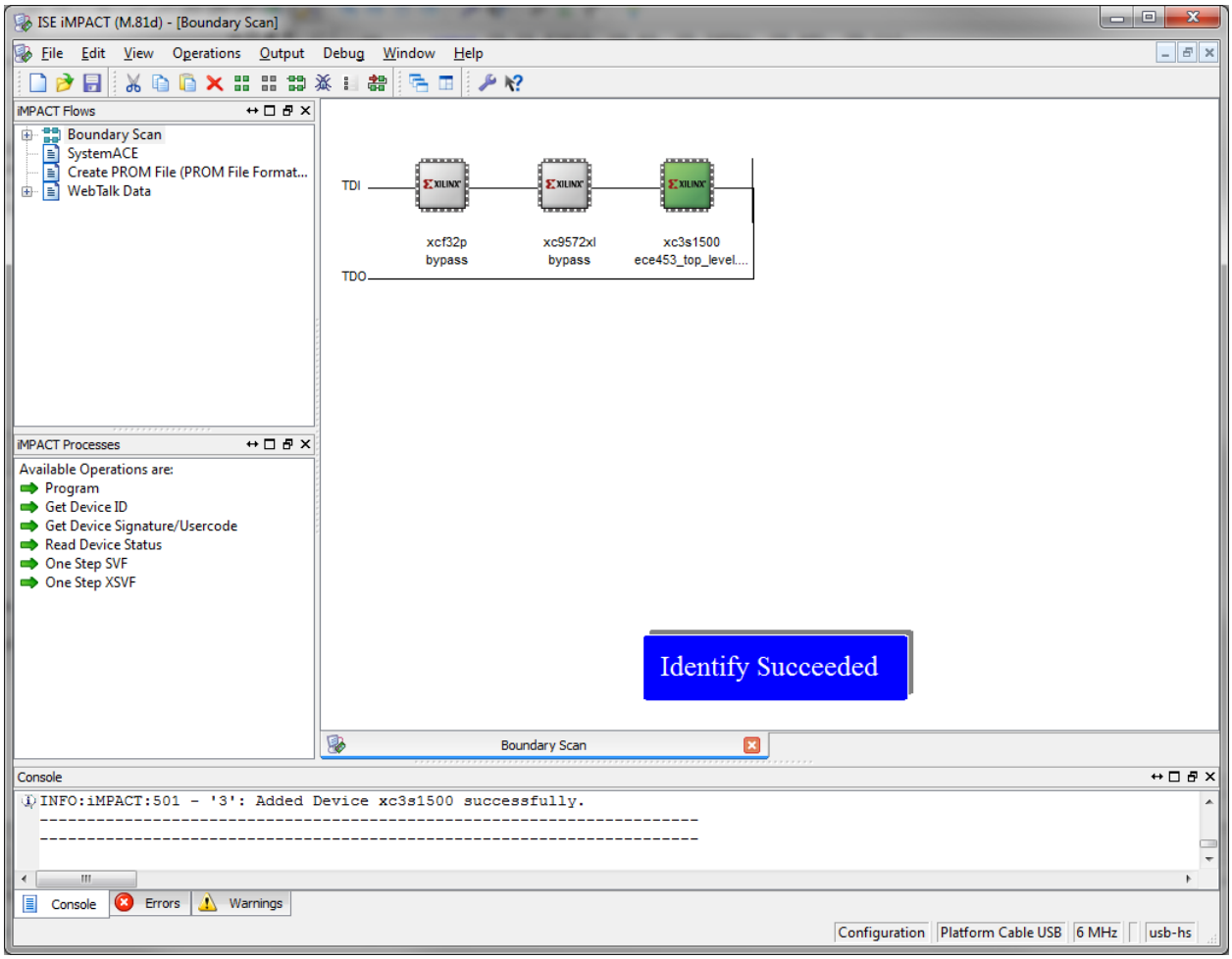


11. Double click 'Boundry Scan'. In the resulting window, Right Click where is asks you to and choose 'Initialize Chain'.



12. There are three devices in our JTAG scan chain. You will be prompted to load a file for each device, but you will want to select 'Bypass' for the first two. For the 3rd device (the FPGA), you will choose the .bit file that you generated above. Click on 'Apply', then 'OK'.

Your screen should look like this:



13. Right Click on the xc3s1500 and select 'Program'.
14. Verify that the switches and LEDs respond as expected. If there are other errors, those are your fault and need to be fixed!

Once the project is completed and working, notify the instructor.