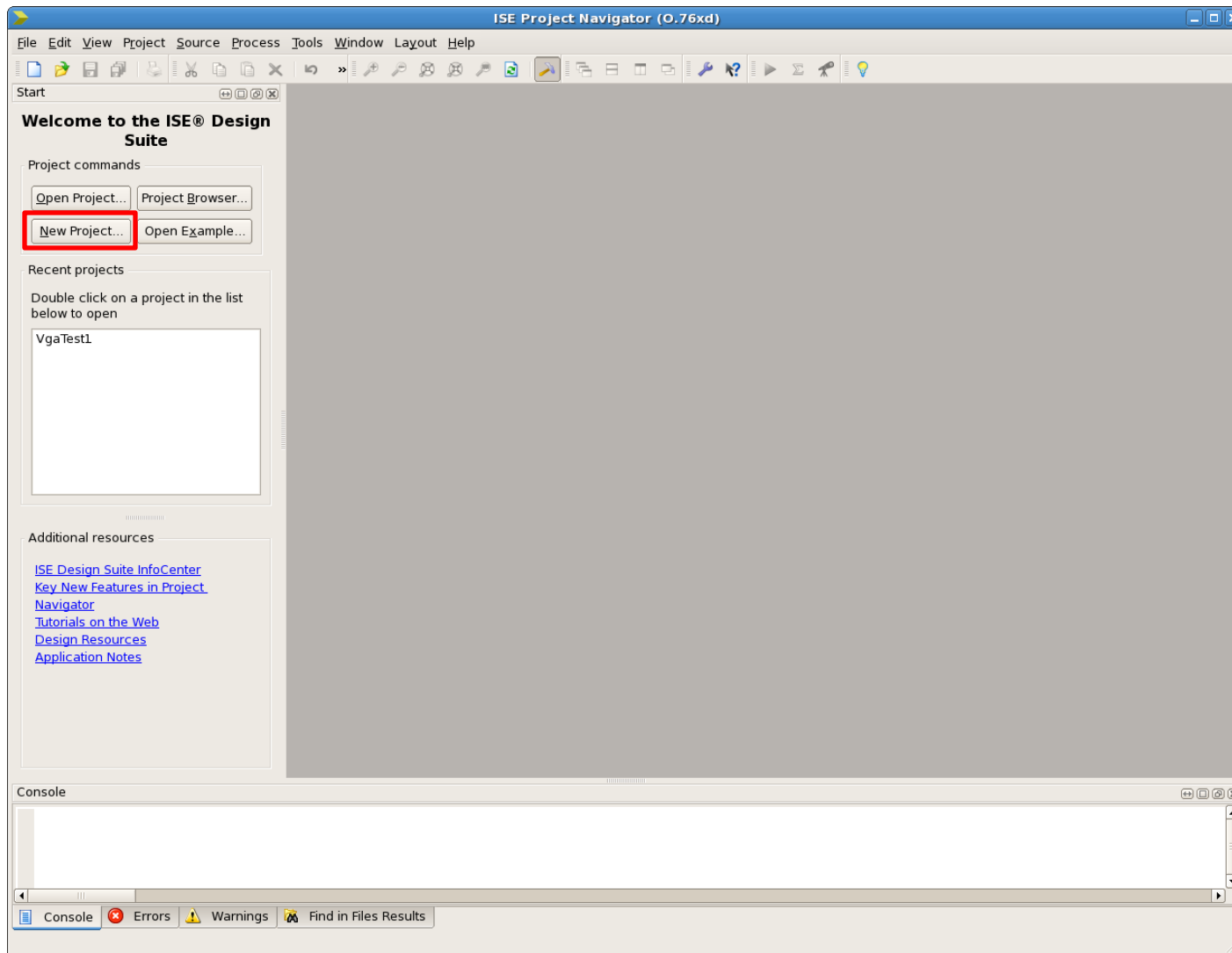


COE 758 Xilinx ISE 13.4 Tutorial 1

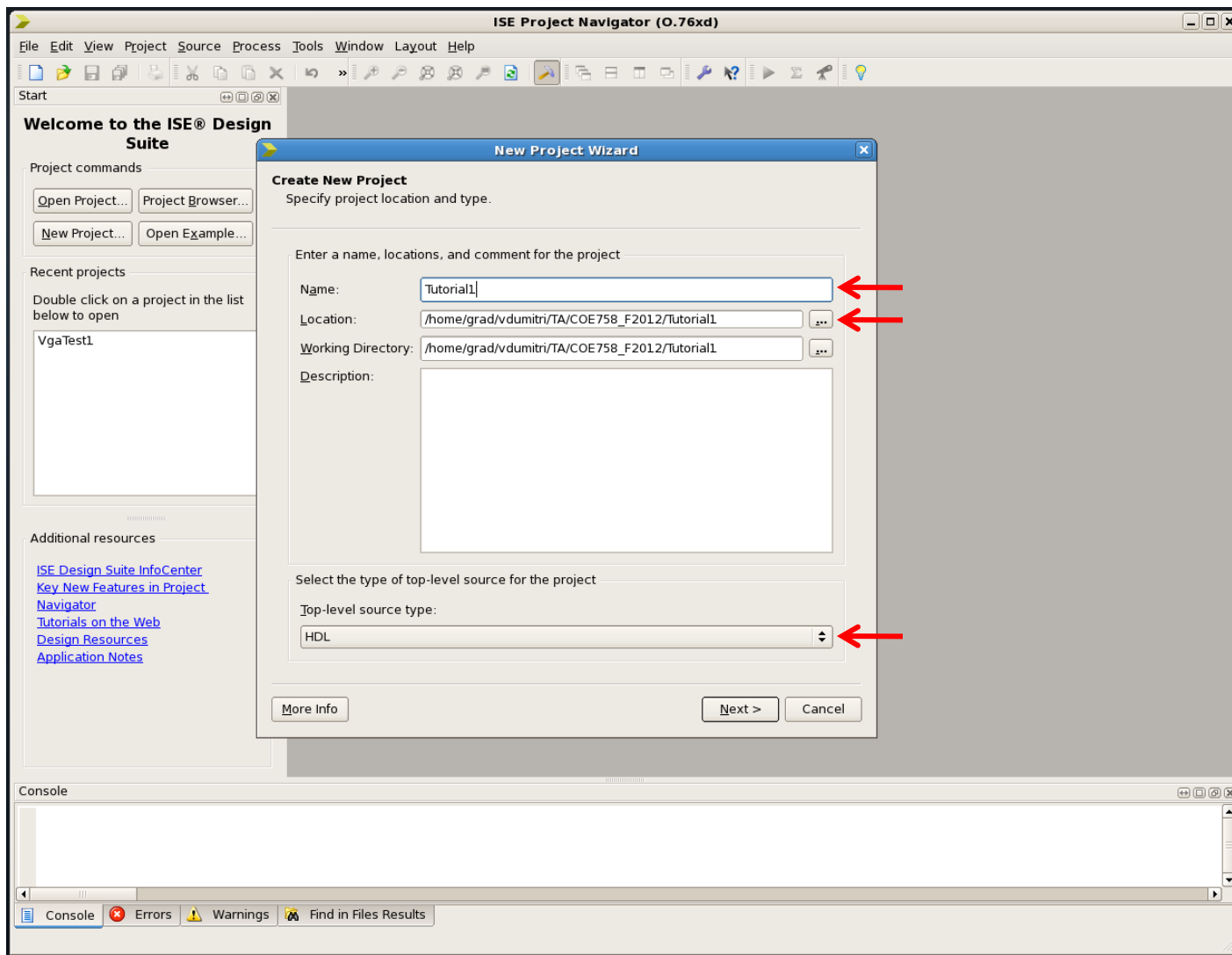
Creating a Basic Project

Tutorial 1 Overview

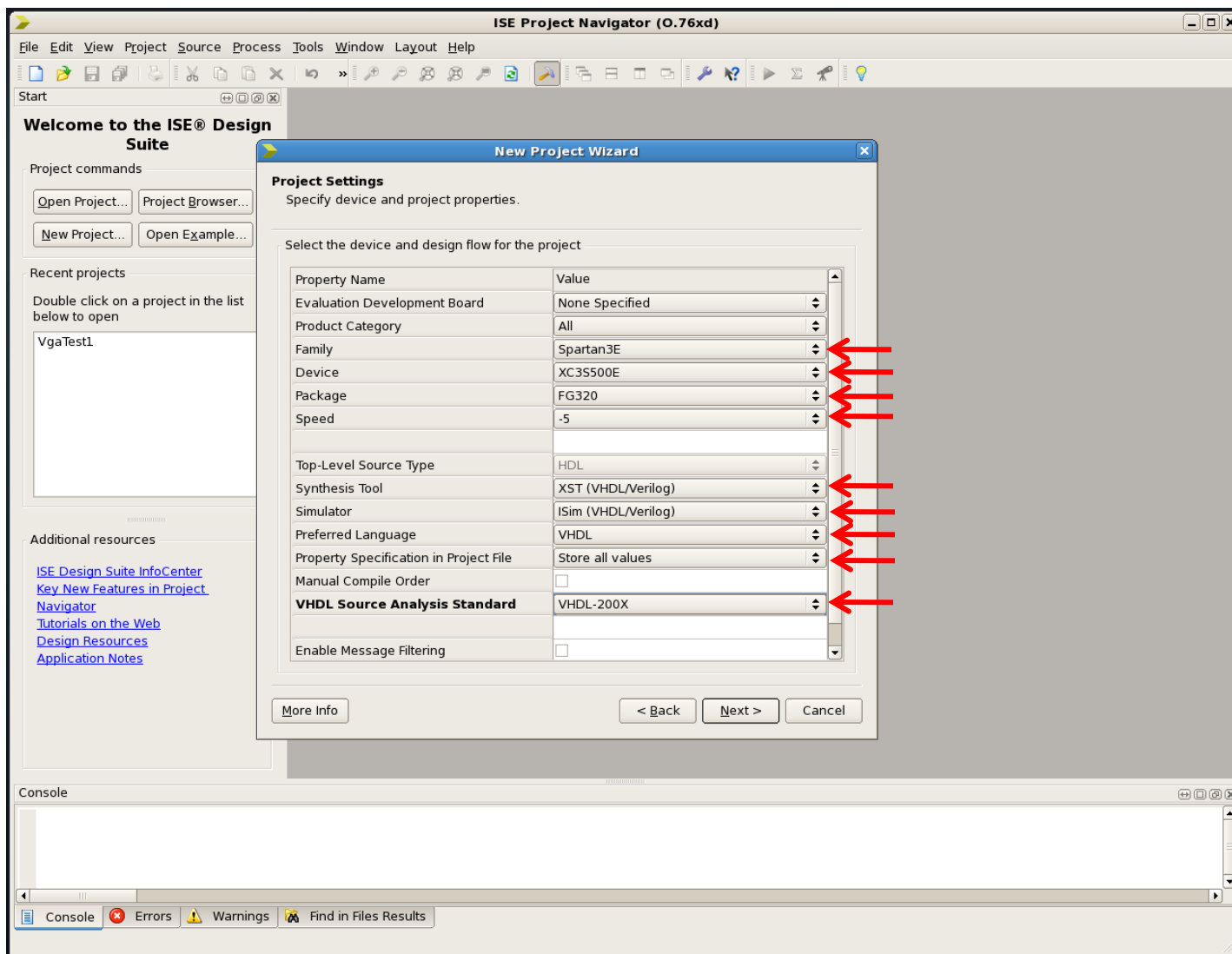
- This tutorial will cover the basics of creating an ISE project, and implementing the design using a Spartan FPGA. The topics covered are:
 - Project creation.
 - HDL design entry.
 - Constraint creation.
 - Generation of design bit-stream.
 - FPGA configuration.



To begin the tutorial, launch the ISE 13.4 Application. Close the Message of the Day window (by selecting OK). What you are left with is a window similar to the above image. If you have worked on a previous project, the last project will open automatically. Begin by creating a new project. To do this, either click on the menu item **File ➔ New Project**, or click on the button outlined in the image above.



Select an appropriate name for the project, and specify a location where the project working directory will be stored. Finally, specify the top-level source type as being HDL, and click **Next**.



Project Configuration:

Family: **Spartan 3E**

Device: **XC3S500E**

Package: **FG320**

Speed: **-5**

Synthesis tool: **XST**

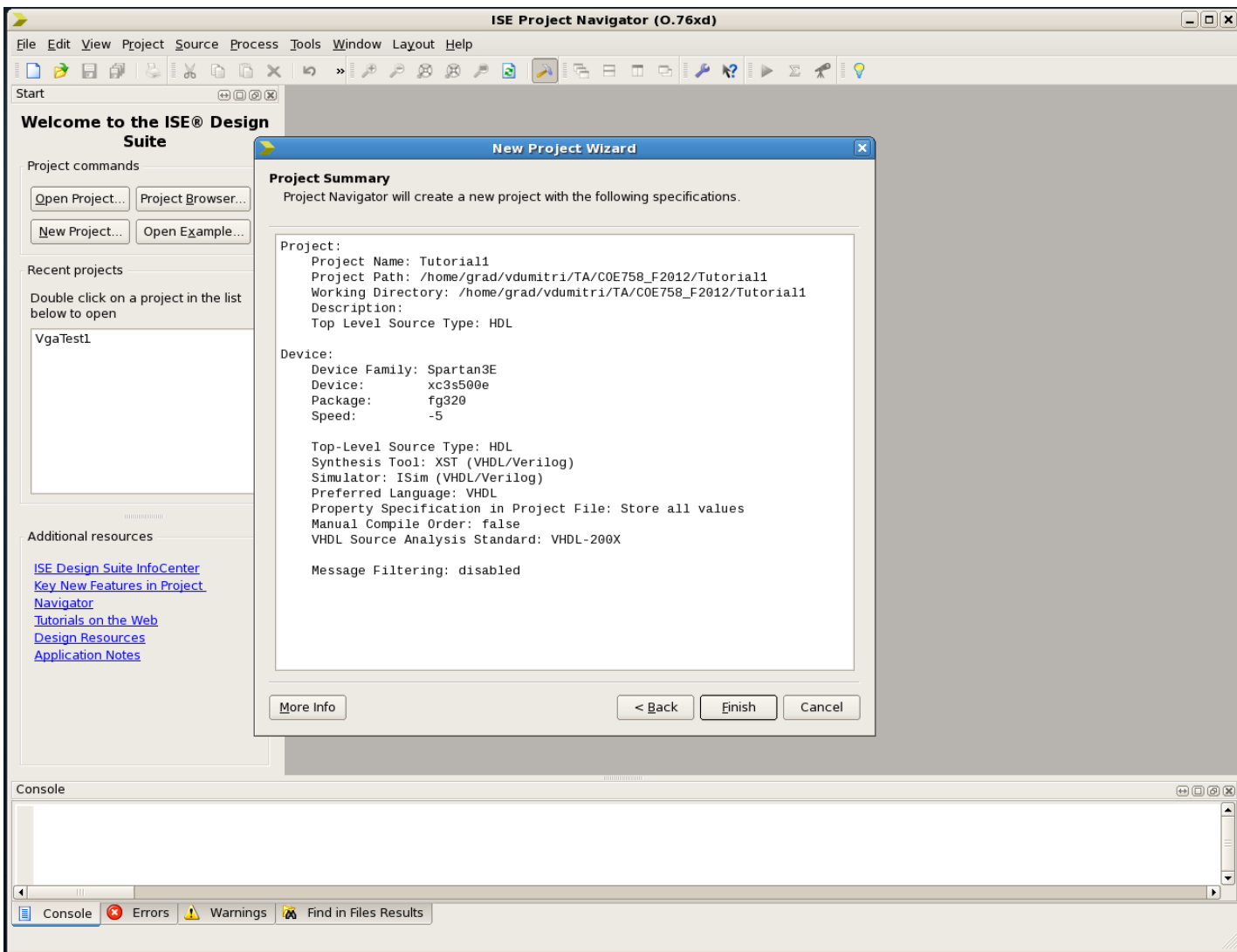
Simulator: **ISim**

Preferred Language: **VHDL**

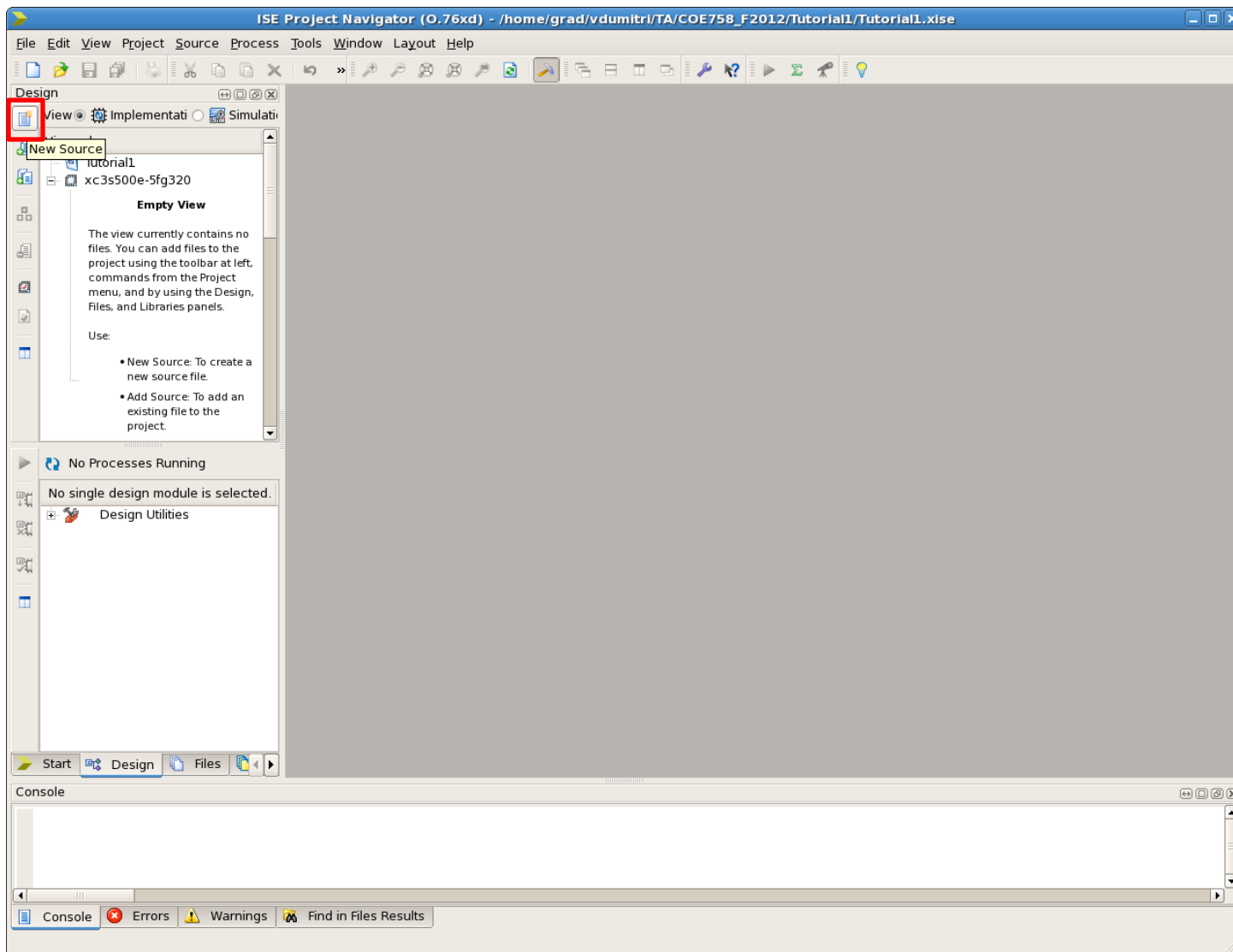
VHDL Source Analysis

Standard: **VHDL-200X**

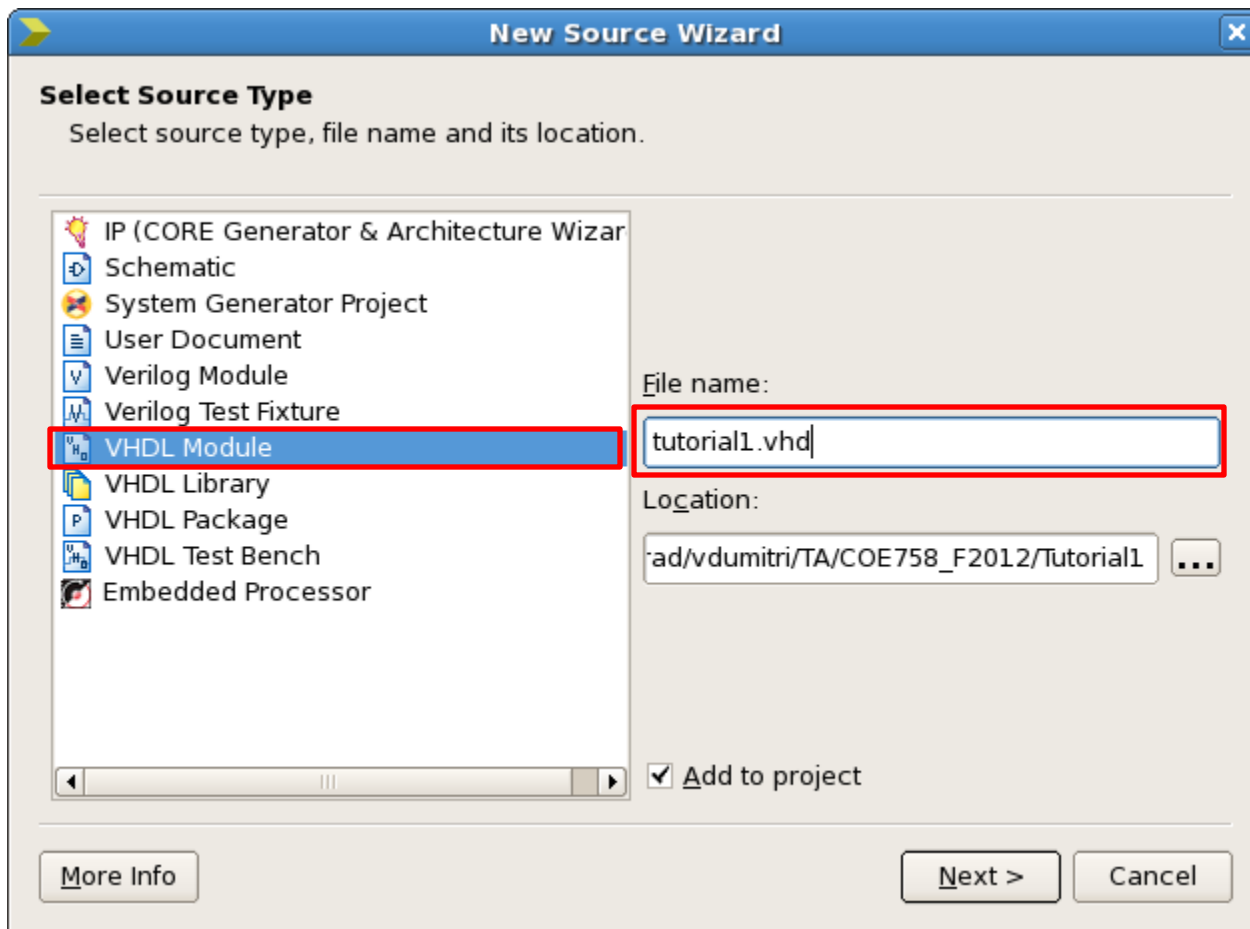
Once a project name and project location have been specified, additional project properties must be selected. Most importantly, the programmable device being used must be selected. For this tutorial, select the device as described above, next to the image. Likewise, certain tools and settings being used with the project must be selected (the synthesizer, simulator and HDL settings). Once again, select the settings as listed above. Finally, click **Next**.



As a final step in the project creation process, the CAD flow presents you with a summary of your Project, including the location, device information, and selected options. Click **Finish** to actually generate the new project.



It is now time to add an actual design to your project, by adding a new source to the project. Do this either by clicking the button outlined above, or by selecting the menu option **Project → New Source ...** .



A new window will appear which allows you to specify the type of source file you wish to create and add to the project, as well as its name. Select **VHDL Module** as the source type, and enter a name for the file. Click **Next** to continue.

New Source Wizard

Define Module
Specify ports for module.

Entity name:

Architecture name:

Port Name	Direction	Bus	MSB	LSB
clk	in	<input type="checkbox"/>		
led	out	<input checked="" type="checkbox"/>	7	0
switches	in	<input checked="" type="checkbox"/>	3	0
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

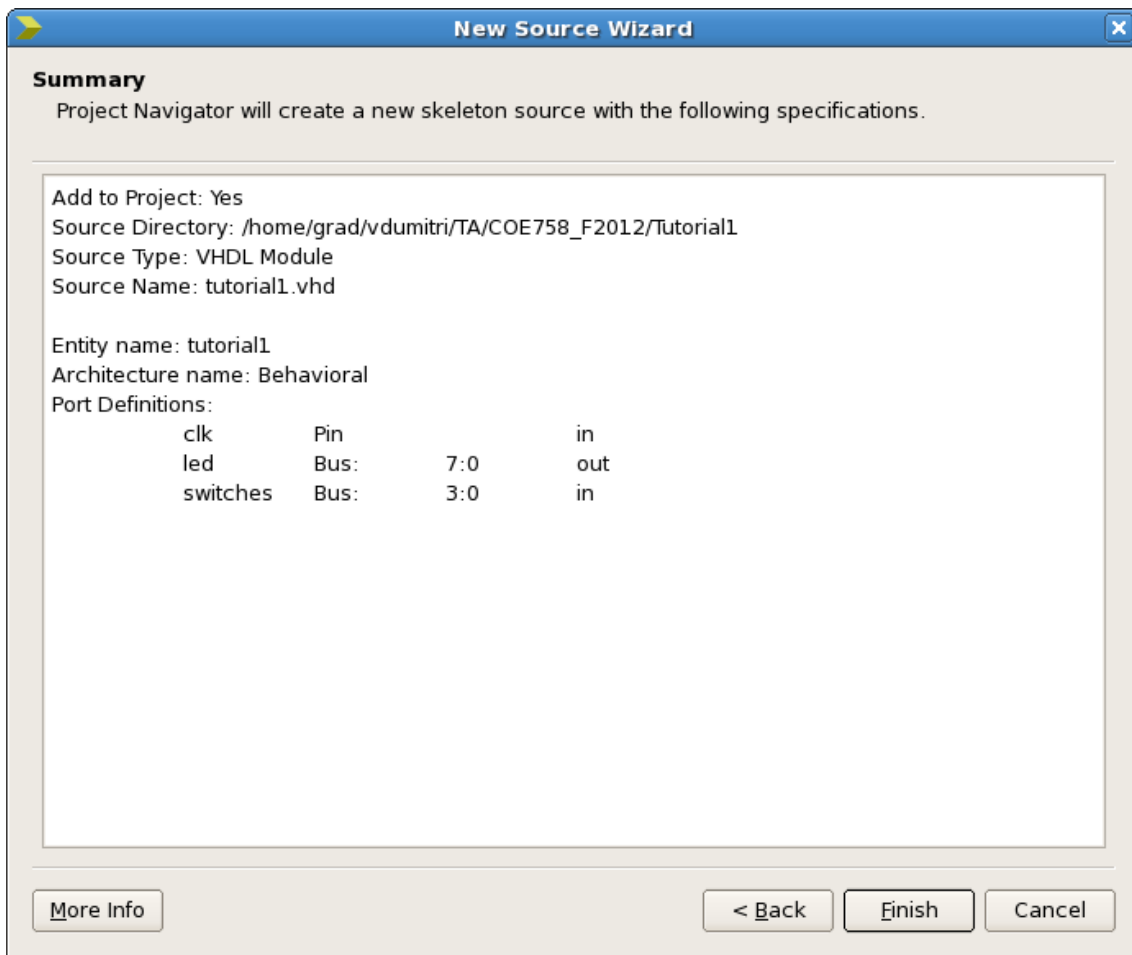
Port Specifications:

clk ➔ *in*;

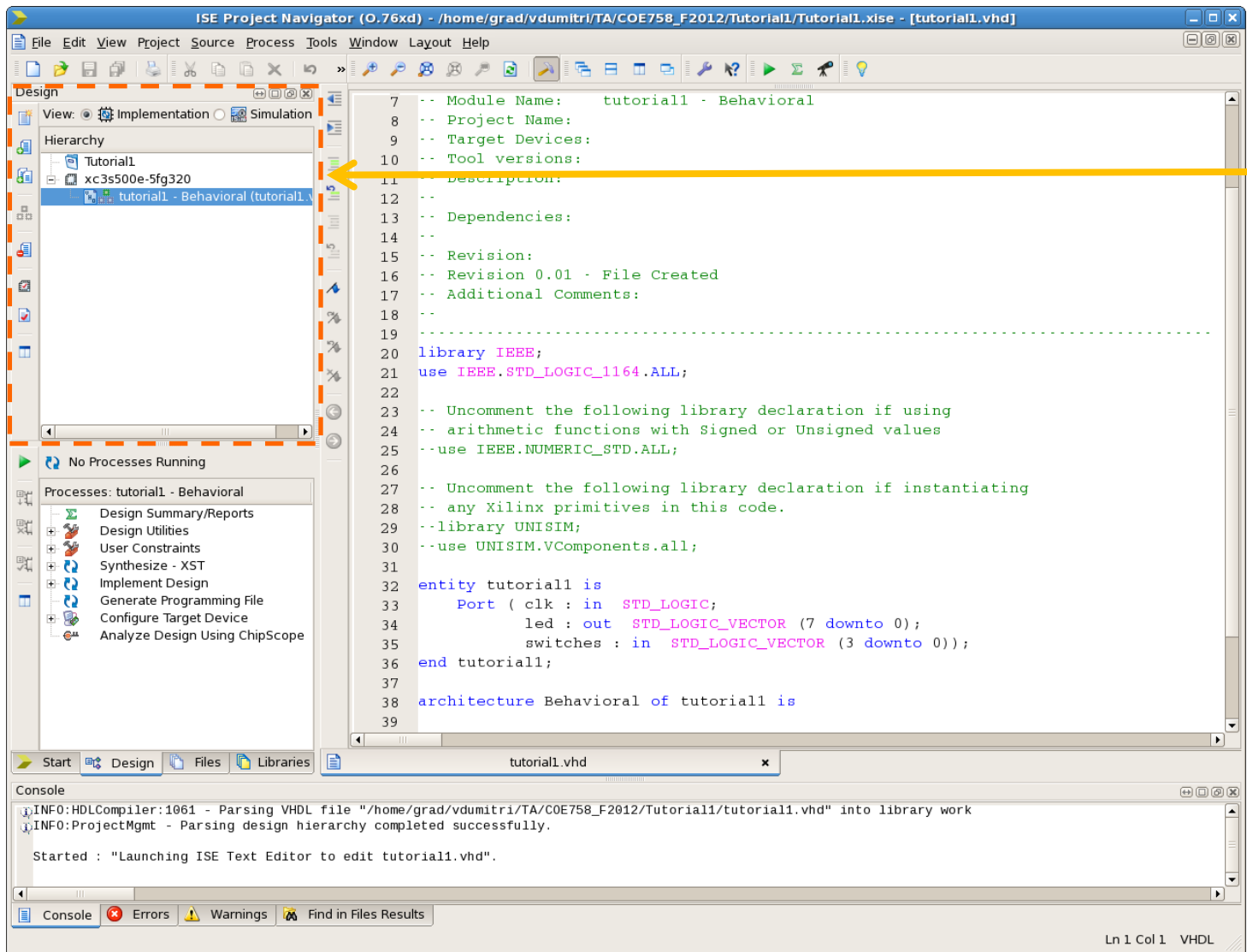
led ➔ *out*; *MSB: 7; LSB: 0*;

switches ➔ *in*; *MSB: 3; LSB: 0*;

When adding a new VHDL source to a project, ISE allows you to automatically generate the entity declaration for the design which will be contained in the file. For the first tutorial, add three ports, as shown above. Finally, click **Next** to continue.



Before the file is actually generated, a summary window will describe the type of file being created, where it will be located, and if it is a VHDL file, the details of the entity declaration. To actually create the file, click **Finish**.

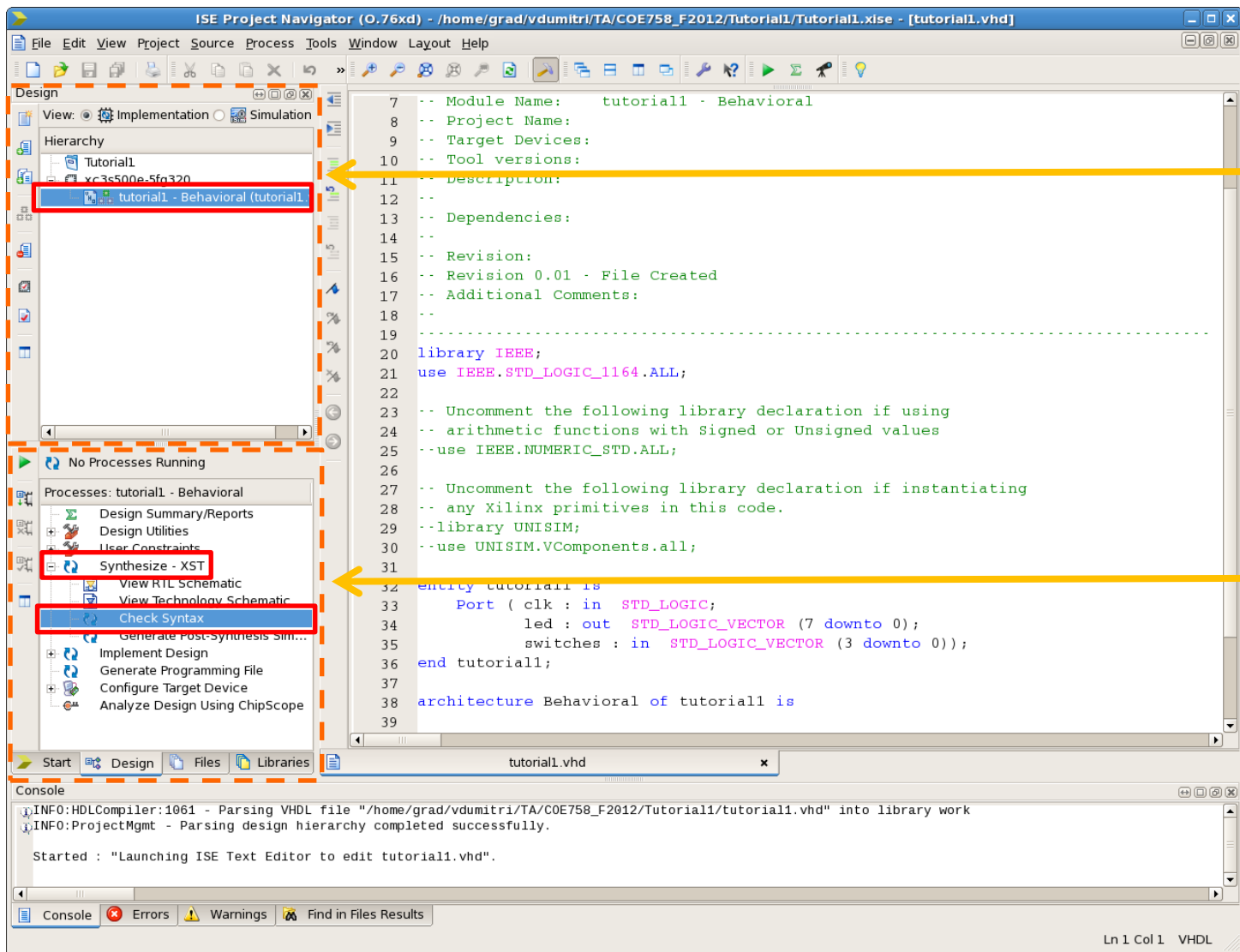


Design
Navigator

Once the VHDL file was added to the project, ISE should look like the above image. You can now add actual circuitry to your design, by editing the file. If the file proper is not open, it can be opened double clicking its name in the design navigator.

library IEEE;		Library declarations.
use IEEE.STD_LOGIC_1164.ALL;		
use IEEE.STD_LOGIC_ARITH.ALL;		
use IEEE.STD_LOGIC_UNSIGNED.ALL;		
entity tutorial is		Entity declaration.
Port (
clk : in STD_LOGIC;		
led : out STD_LOGIC_VECTOR (7 downto 0);		
switches : in STD_LOGIC_VECTOR (3 downto 0)		
);		
end tutorial;		
architecture Behavioral of tutorial is		Declaration of the counter register.
signal counter: std_logic_vector(29 downto 0);		
begin		
process(clk)		
begin		Up/Down counter process.
if(clk'Event and clk='1') then		
if(switches(0)='1') then		
counter<=counter+'1';		
else		
counter<=counter-'1';		
end if;		
end if;		
end process;		
led(7 downto 0)<=counter(29 downto 22);		Counter connection to led port.
end Behavioral;		

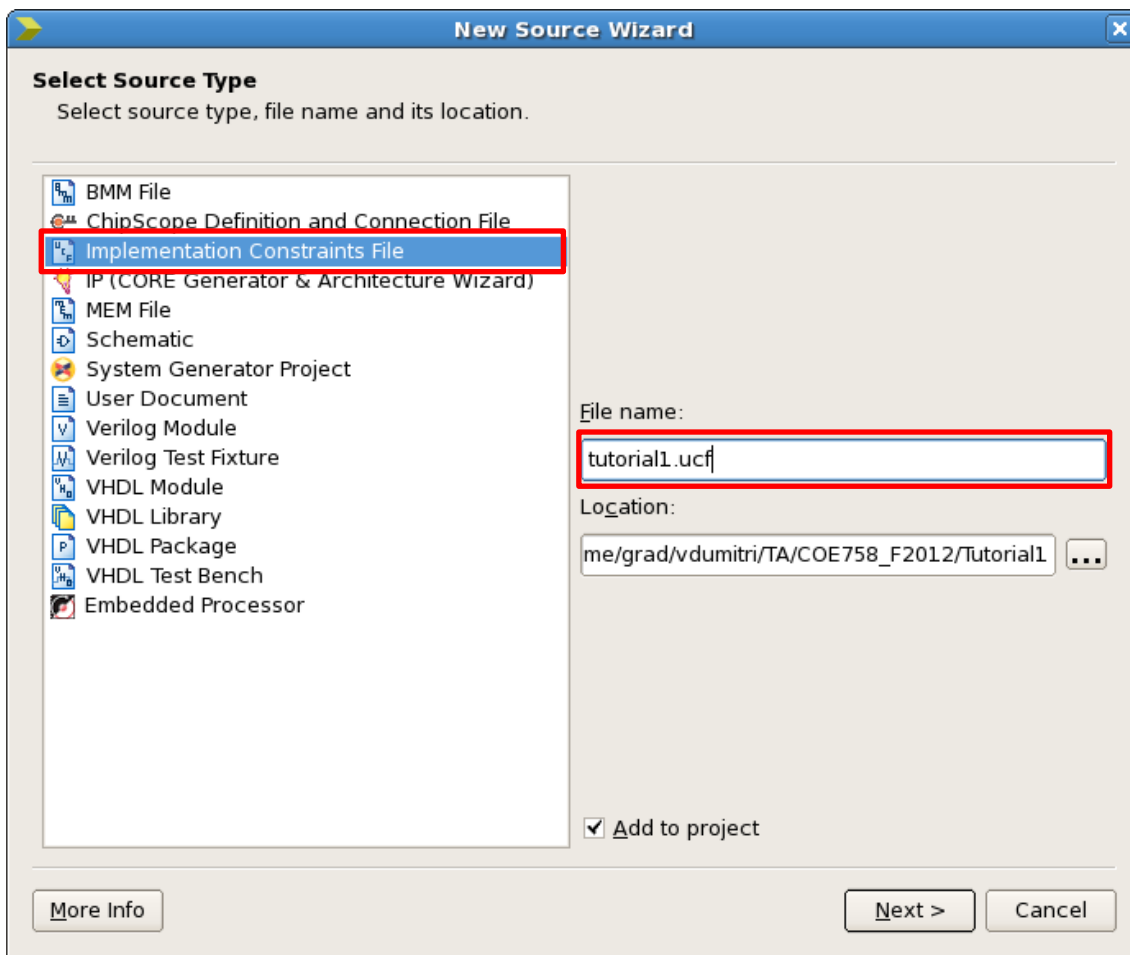
The complete code for the circuit used in this tutorial is listed above. As can be seen, the design consists of a selectable up/down counter. The eight most significant bits of the counter are connected to the **led** output port, and will drive the eight LEDs present on the laboratory development boards.



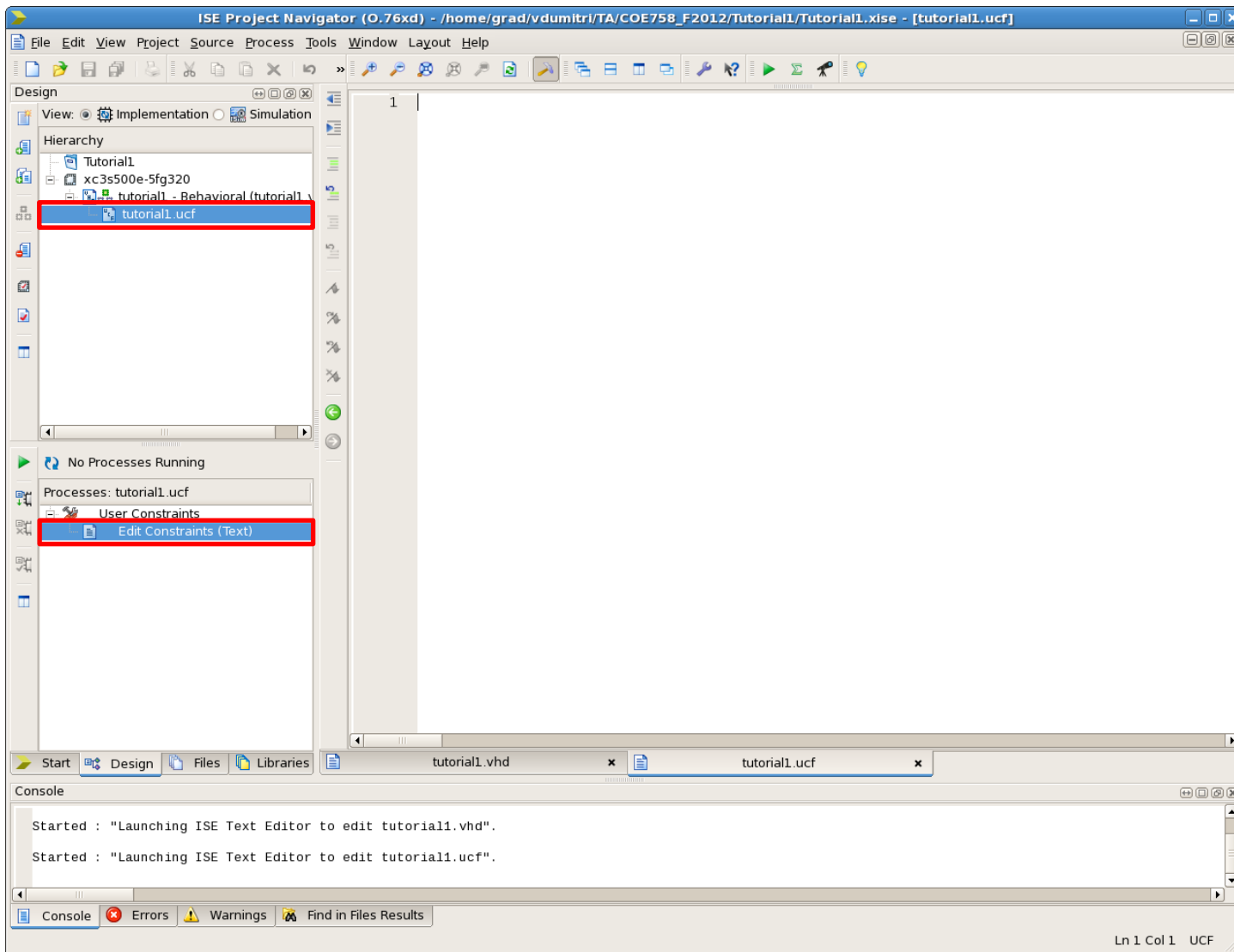
Design
Navigator

Activities
Pane

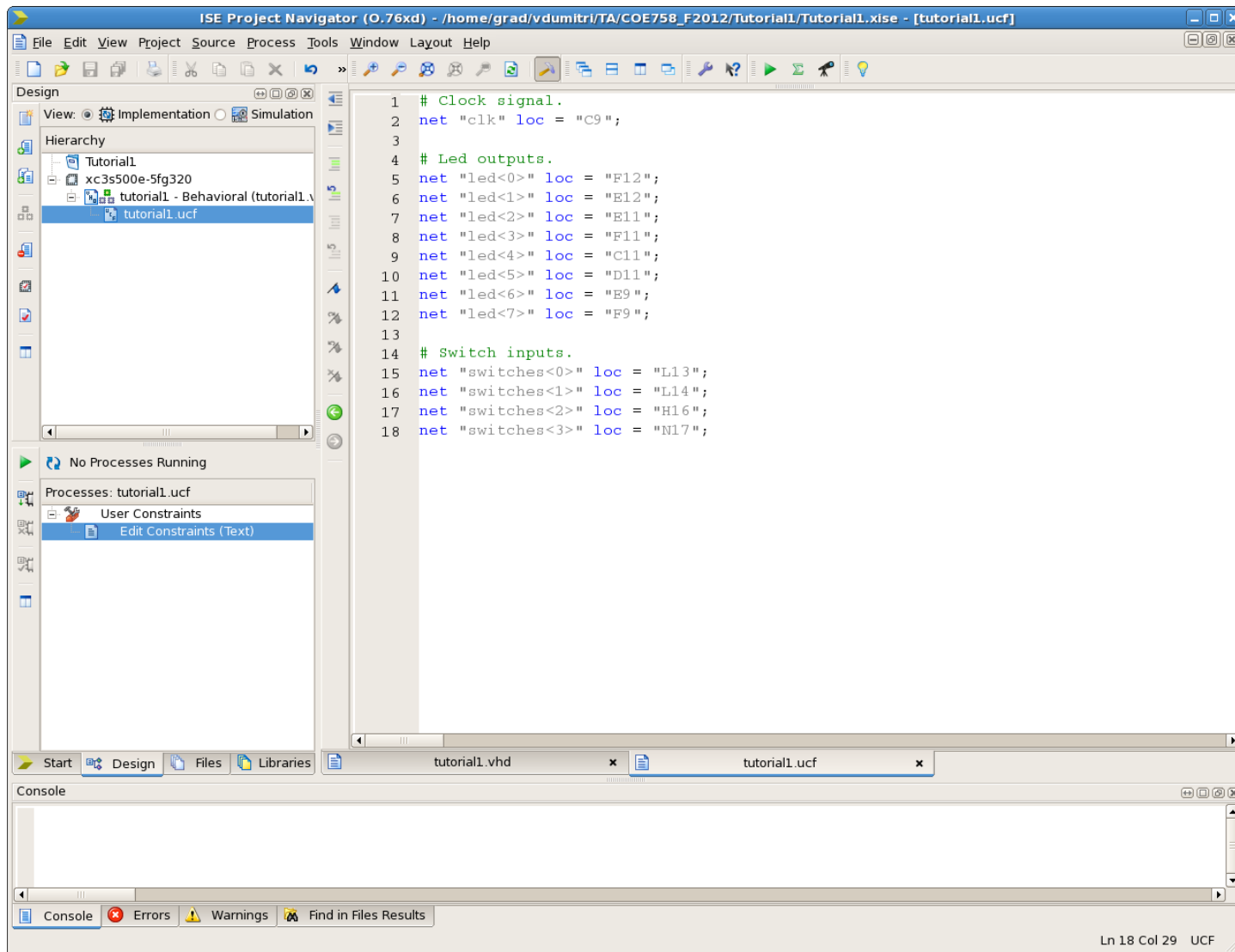
Before a design is fully synthesized, all source files should be verified to ensure that they conform to VHDL syntax requirements. This can be done automatically, by selecting the source file in the design navigator, expanding the **Synthesize - XST** action menu in the activities pane, and activating the option **Check Syntax**; the option can be activated either by double-clicking on it, or by right-clicking on it, and selecting **Run**. If any syntax errors are found, correct them. Note that Check Syntax will only identify syntax-related errors. Other errors may still exist, which will be detected at later stages of the implementation process.



Once syntax has been checked, the design should be synthesized; this is done by selecting and activating the **Synthesize – XST** command in the Activities Pane; once again, activation is achieved by double-clicking or by right-clicking and selecting Run. Once this process completes, an *Implementation Constraints File* should be added to the project. This file constrains the implemented design in various ways. Specifically, for this tutorial, the constraints file specifies how the top level ports in the design are connected to the physical I/O pins of the FPGA being used. To add a Implementation constraints file, follow the same process that was used to add a new source file. However, this time, select **Implementation Constraints File** as the source type, and name it appropriately. Click **Next**, which will once again bring up a summary of the file being created, and finally click **Finish** to create the file.



Once the constraints file has been created, it must be edited to contain the appropriate information. To edit the constraints file, select it in the Design Navigator, and then select and activate the **Edit Constraints (Text)** option.



Design Constraints:

Net "clk" loc = "C9" ;

Net "led<0>" loc = "F12";

Net "led<1>" loc = "E12";

Net "led<2>" loc = "E11";

Net "led<3>" loc = "F11";

Net "led<4>" loc = "C11";

Net "led<5>" loc = "D11";

Net "led<6>" loc = "E9";

Net "led<7>" loc = "F9";

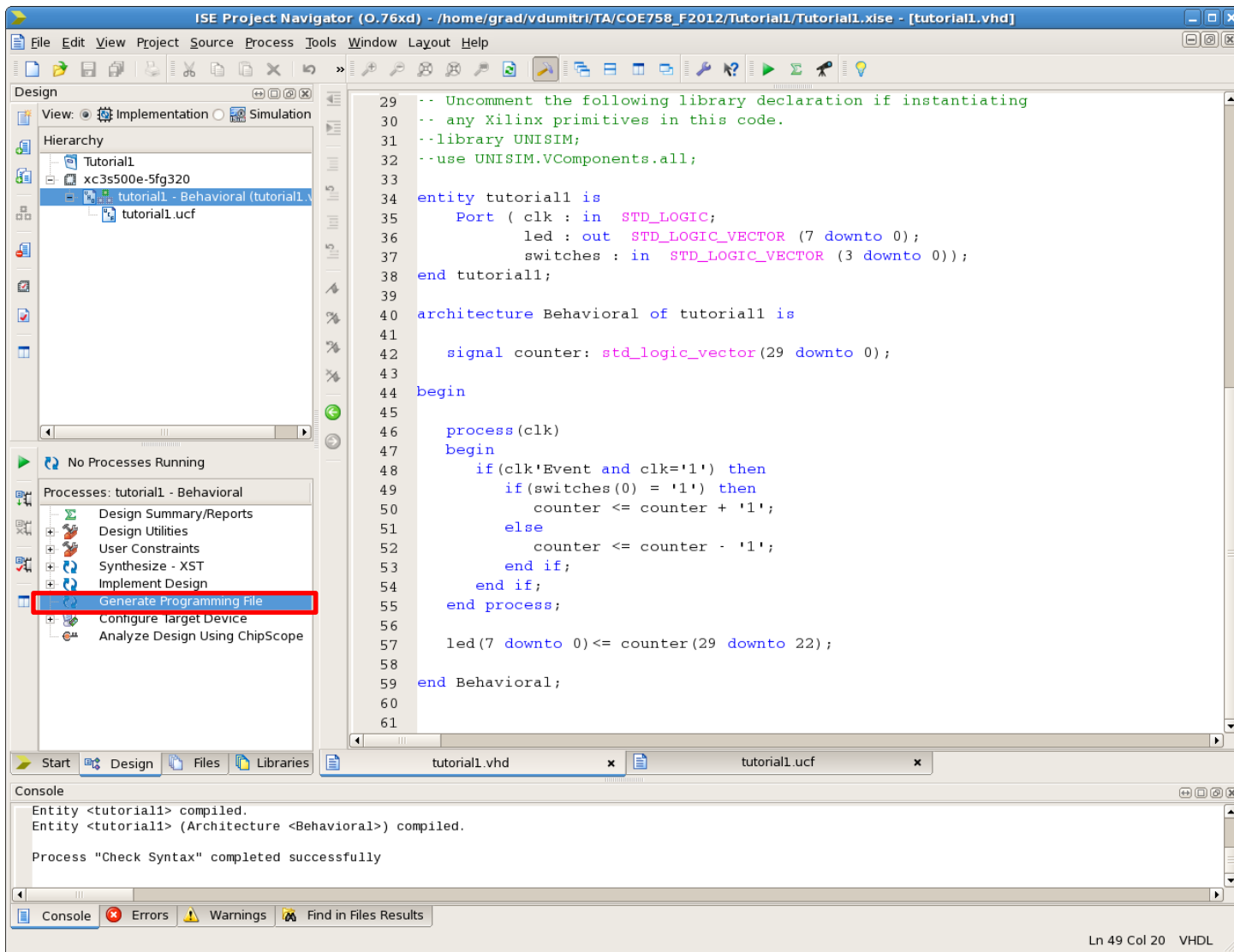
Net "switches<0>" loc = "L13";

Net "switches<1>" loc = "L14";

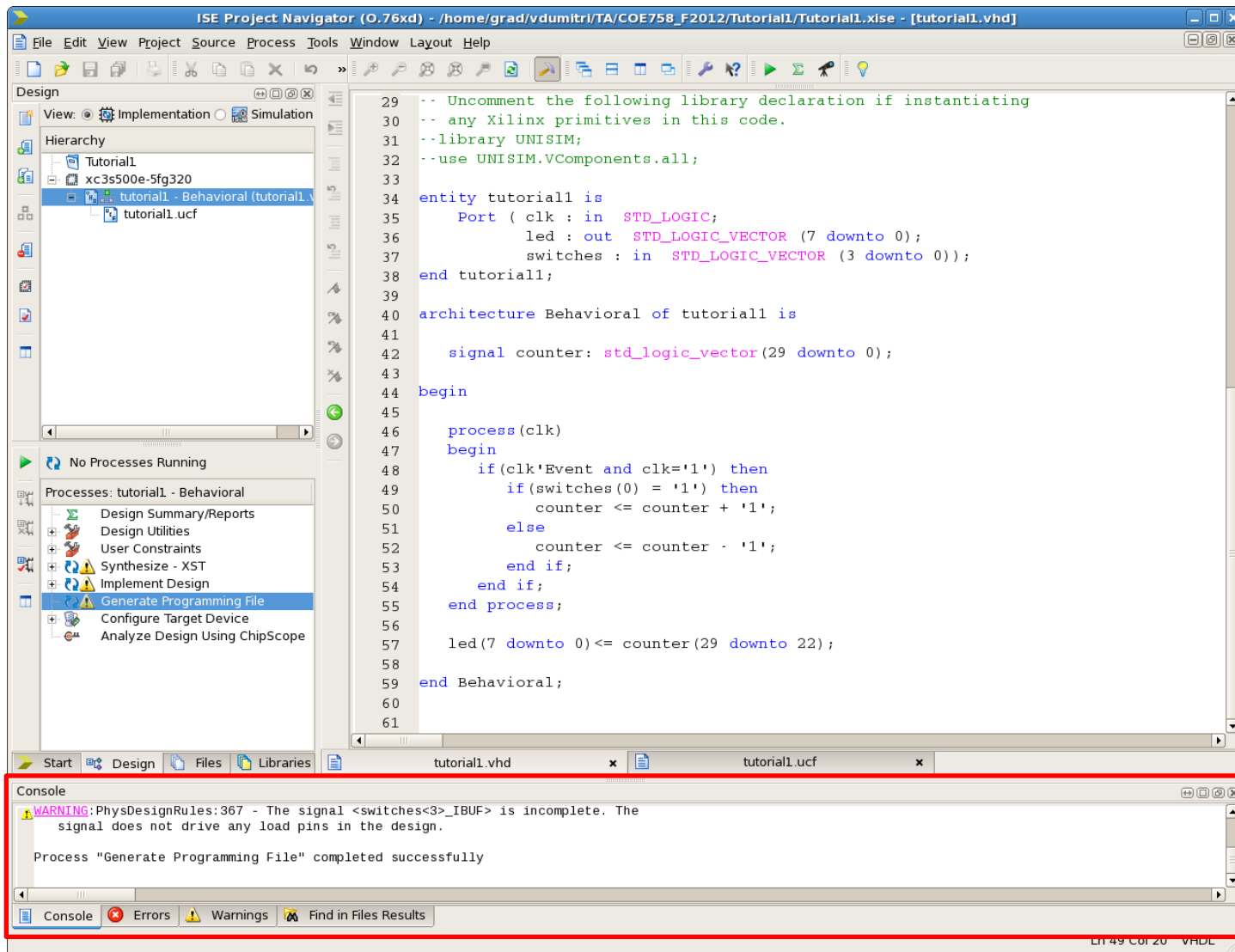
Net "switches<2>" loc = "H16";

Net "switches<3>" loc = "N17";

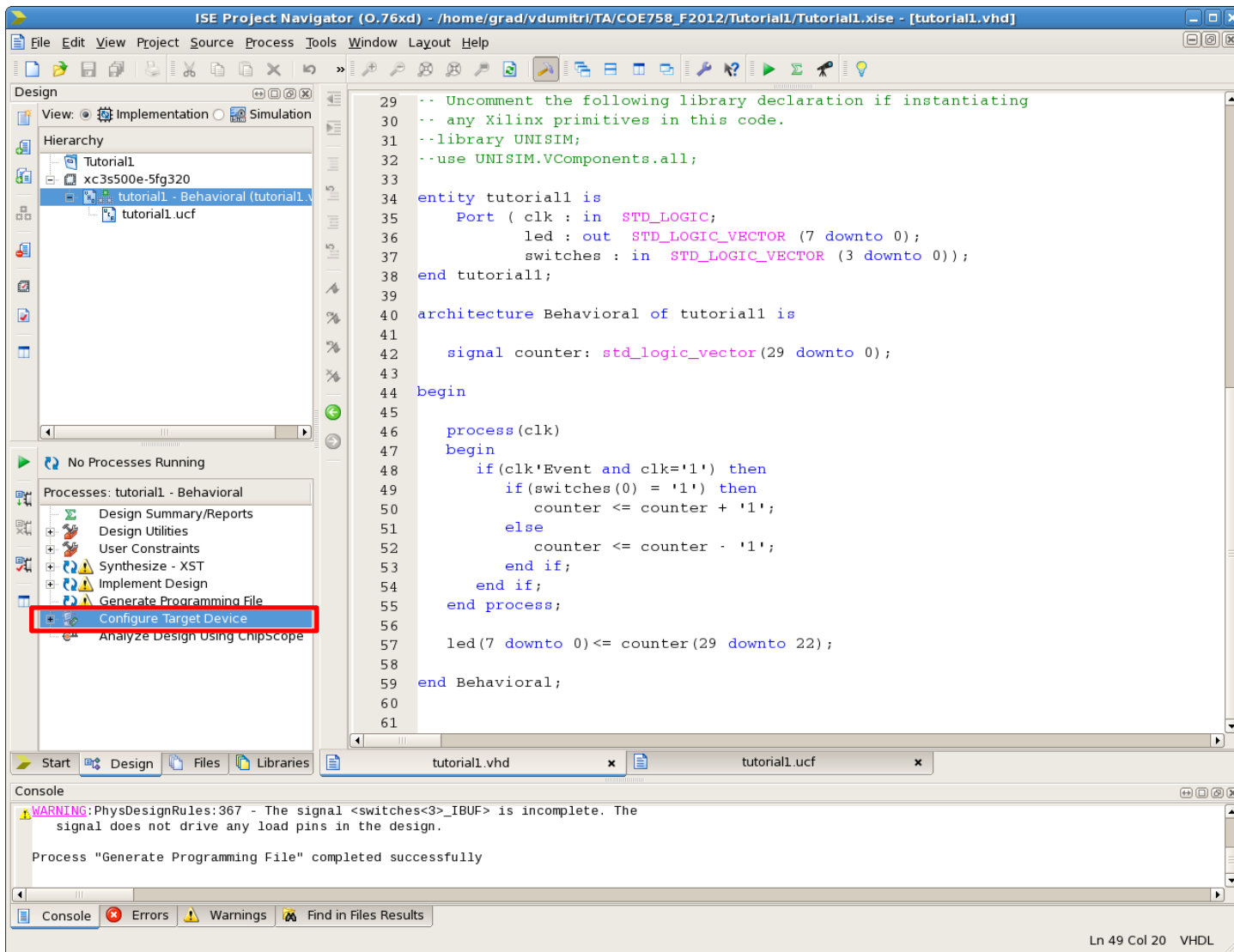
Enter the I/O constraints specified above, and save the file. The design is now ready for implementation.



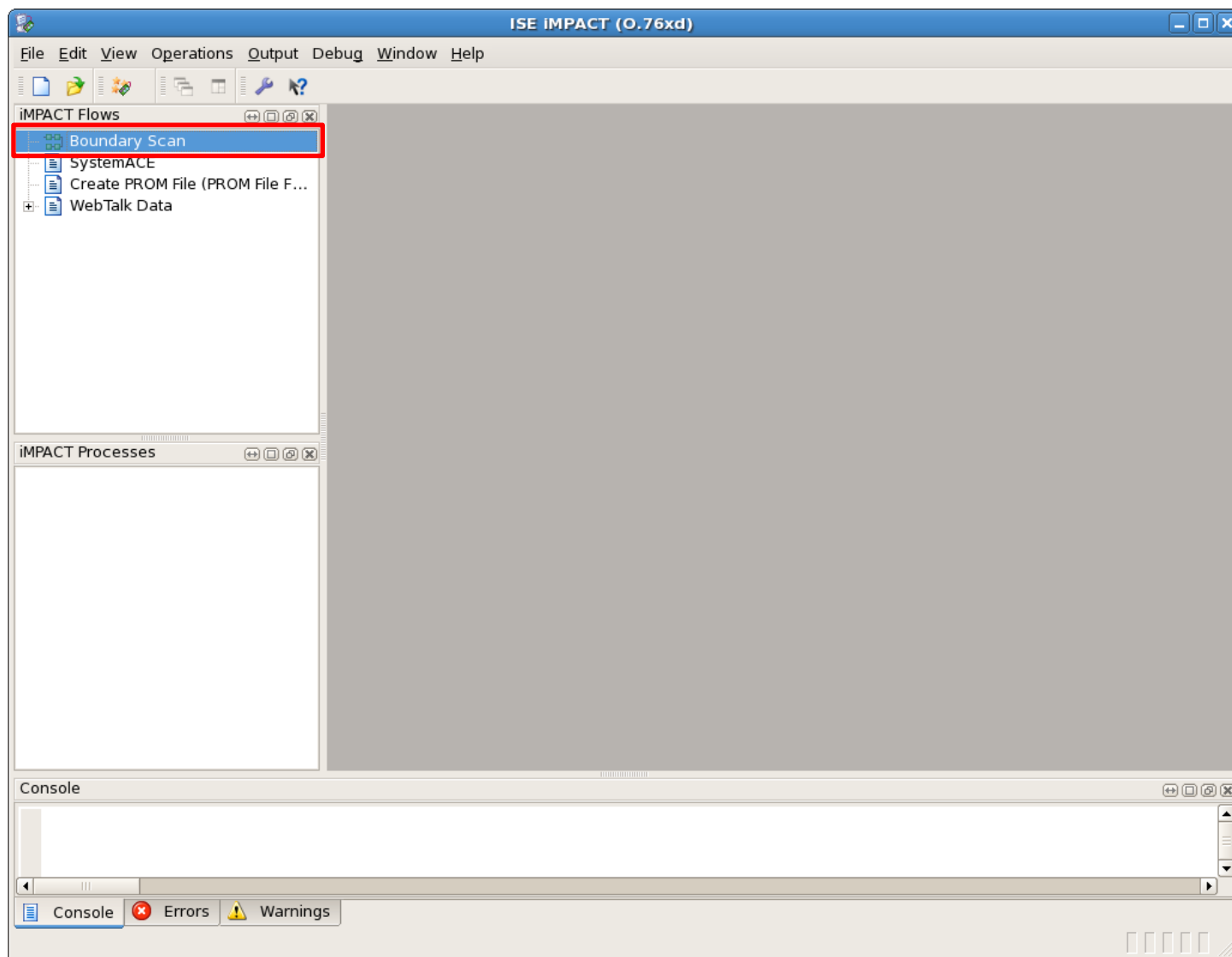
To fully implement the design, select the option **Generate Programming File** in the activities pane. If any errors occur during synthesis or implementation, correct them accordingly.



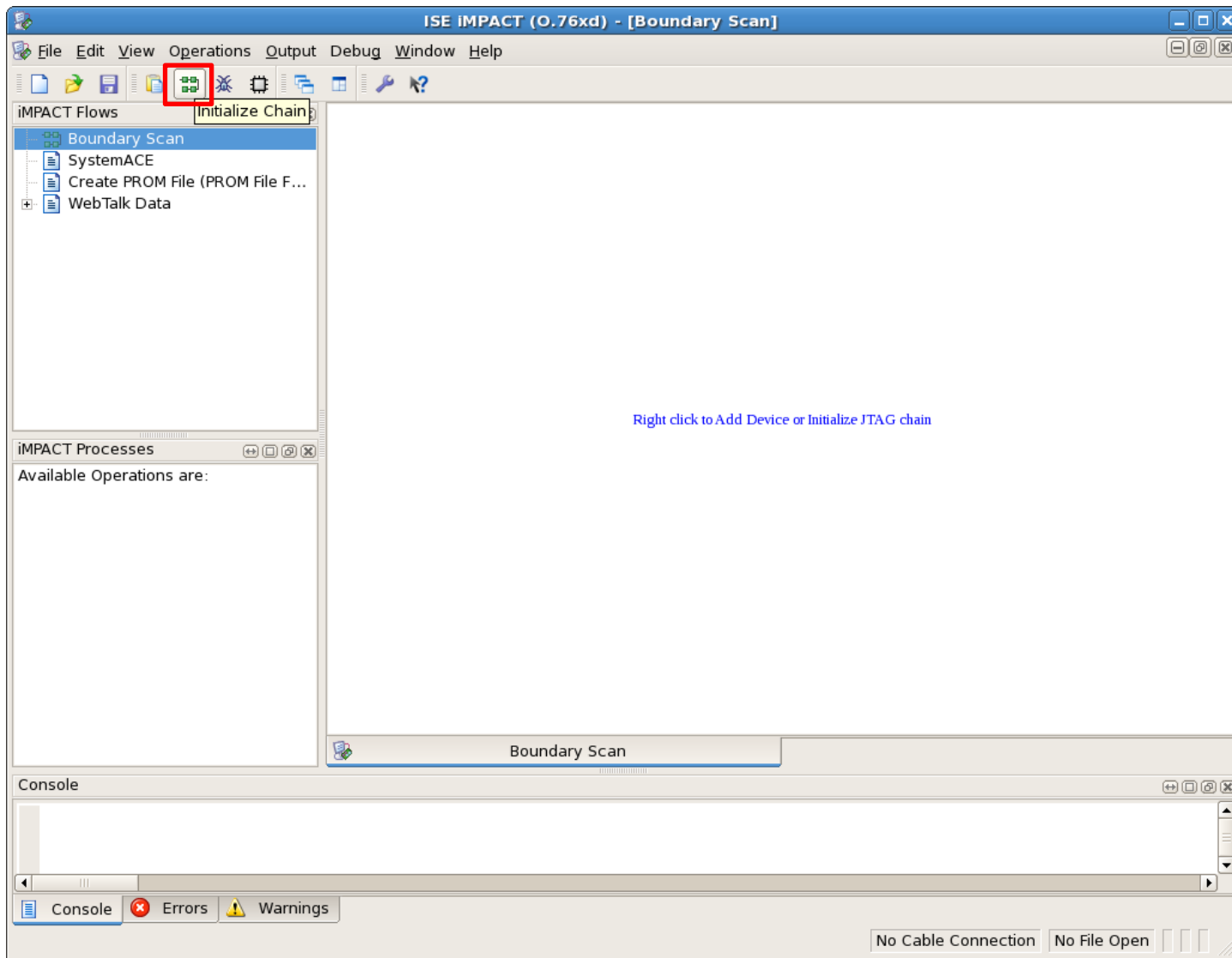
Once the implementation process completes, a message will be displayed in the ISE console. At this point, a bit-stream configuration file has been created. This file, when downloaded to the board FPGA, will implement your design.



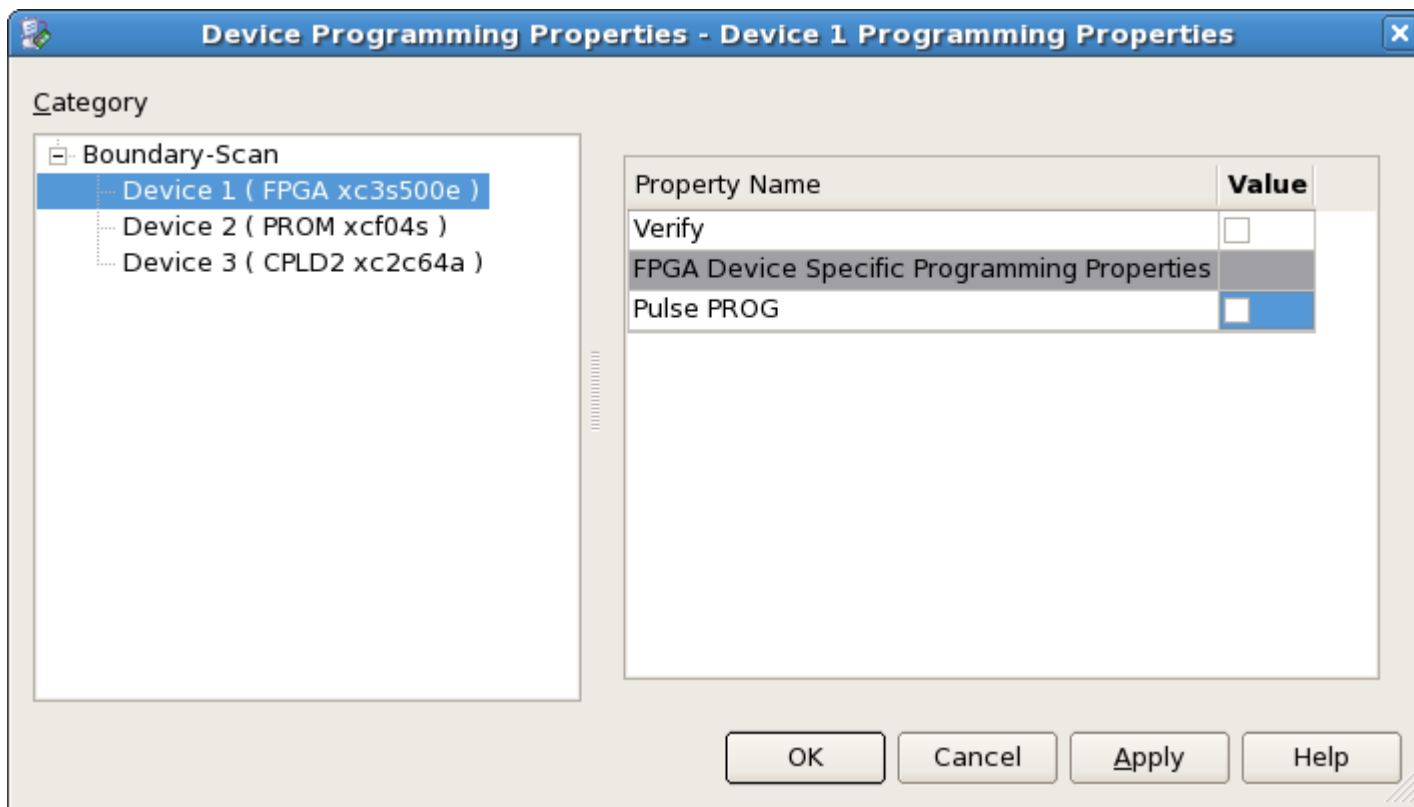
To configure the board FPGA, select and activate the **Configure Target Device** option in the activities pane. This will launch the ***IMPACT*** configuration application. The first thing you will see is a warning stating that no Impact project files exist. Click **Ok** to proceed.



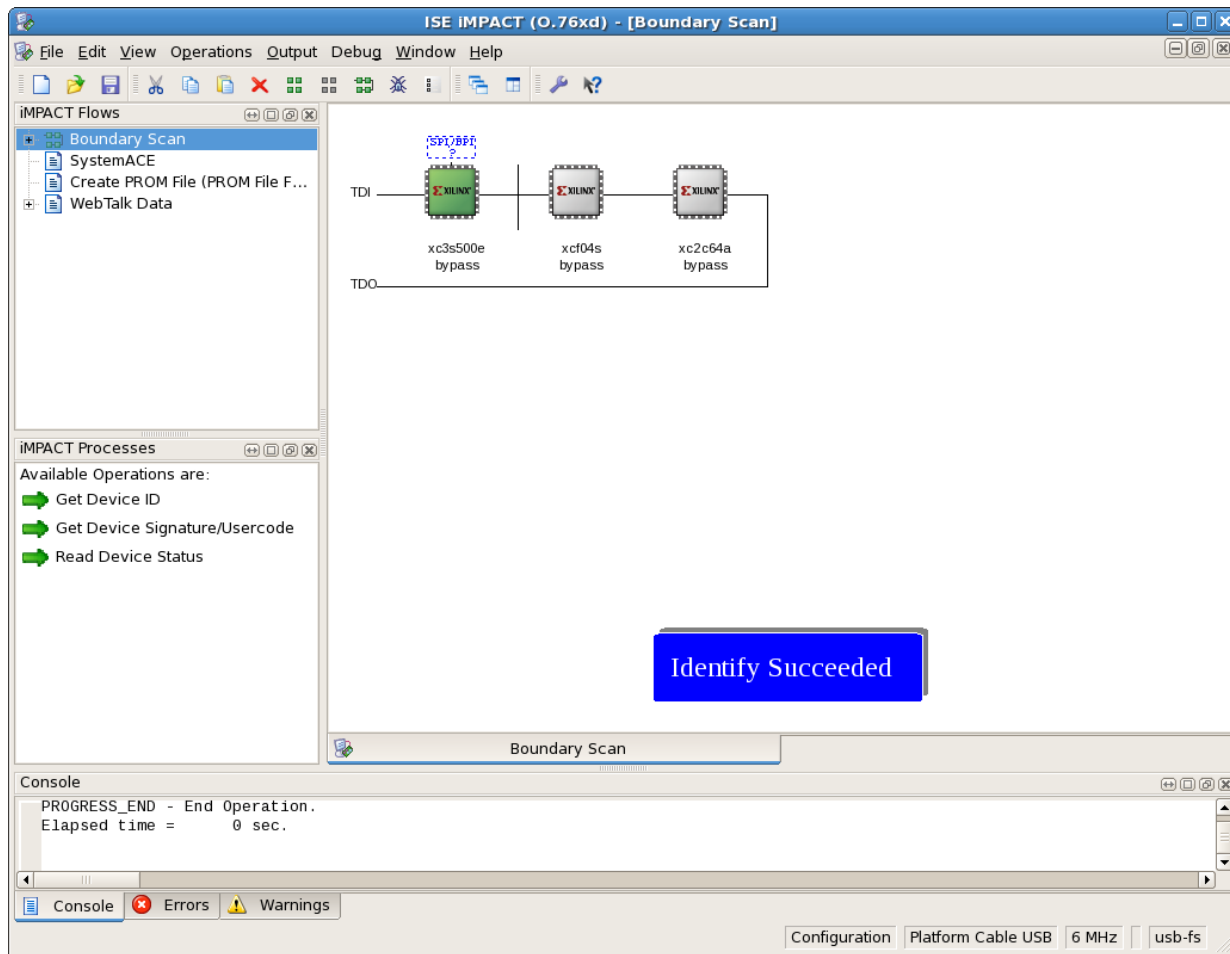
Once iIMPACT is open, you will be presented with a number of configuration flows. Select the **Boundary Scan** flow by double-clicking it in the **iMPACT Flows** tab.



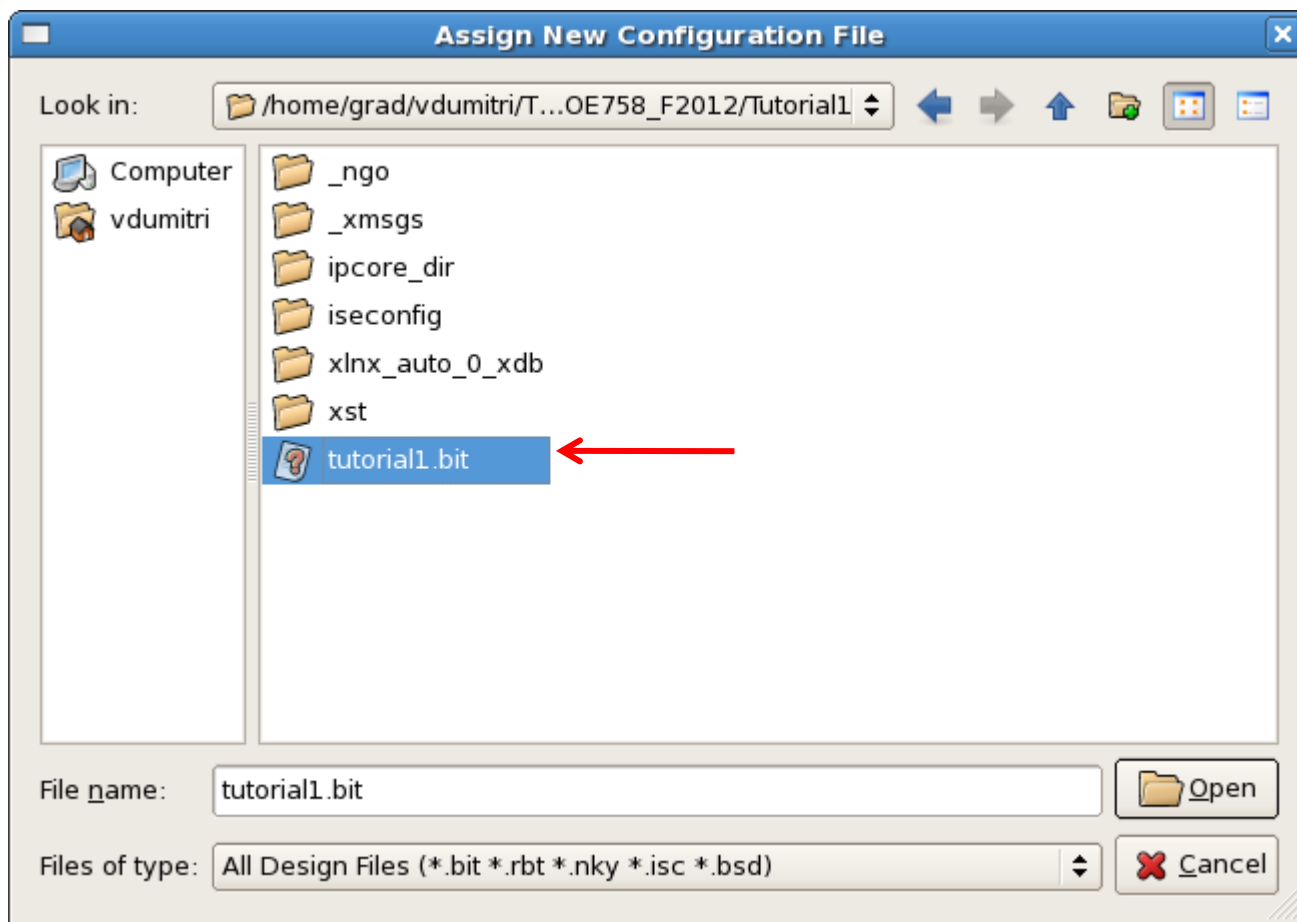
Once the Boundary Scan flow is selected, available devices will be configured using the **JTAG** configuration interface. The first step is to initialize the **JTAG Chain**, by clicking the outlined button. This will send a query through the JTAG interface, and will allow the host computer to determine what devices are present in the chain.



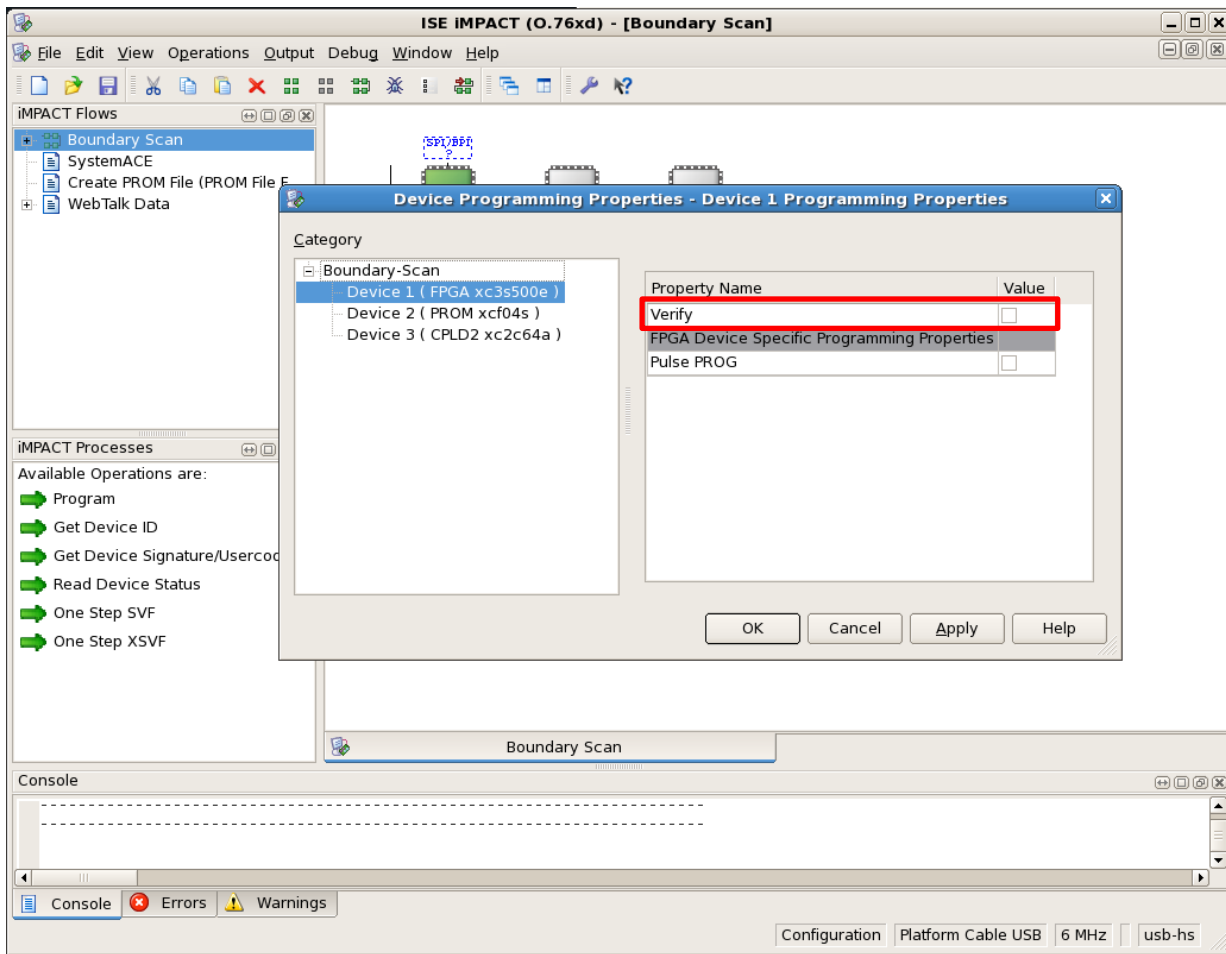
Once the JTAG chain is initialized and all devices present are detected, a window will appear asking you if you now wish to assign configuration files to the detected devices; click **No**. A summary window will then appear (shown above). This window lists the found devices, and allows the user to change certain settings for each device. Click **Cancel** to close the window without making any changes to the configuration setup.



Once the JTAG chain has been initialized, the **XC3S500E** FPGA can be configured with the bit-stream generated earlier. To begin the configuration process select the appropriate device and right-click it; from the menu that appears, select the **Assign Configuration File** option.



A new window will open which allows you to select a bit-stream configuration file. Go to the working directory of your current project, and select the **.bit** file found there. Click **Open** to open the file.



To configure the device with the assigned configuration file, once again select the **XC3S500E** FPGA, right click the device, and select the **Program Device** option. A window will appear asking you if you wish to attach an **SPI or BPI PROM** to the selected device; click **No**. A new window with device programming properties will then appear, as shown above. For this tutorial, ensure that the Verify option is NOT selected. Finally, click **OK** to program the device. Once the programming operation completes, you can observe your counter in action by looking at the LEDs present on the laboratory development boards.

Conclusion

- This concludes the first of the three COE 758 Tutorials. This tutorial has covered the following topics:
 - The creation of a new ISE project.
 - The addition of VHDL source files.
 - The creation of a simple circuit via VHDL entry.
 - The creation and addition of a Implementation Constraints file, necessary to specify connections between design ports and physical I/Os.
 - The implementation of a design and the creation of a configuration file.
 - The configuration of an FPGA with a specific design.
- You are now ready to begin work on the second tutorial, which will familiarize you with the ChipScope integrated logic analyzer. Note that Tutorial 2 will be based on the project which was created in Tutorial 1.