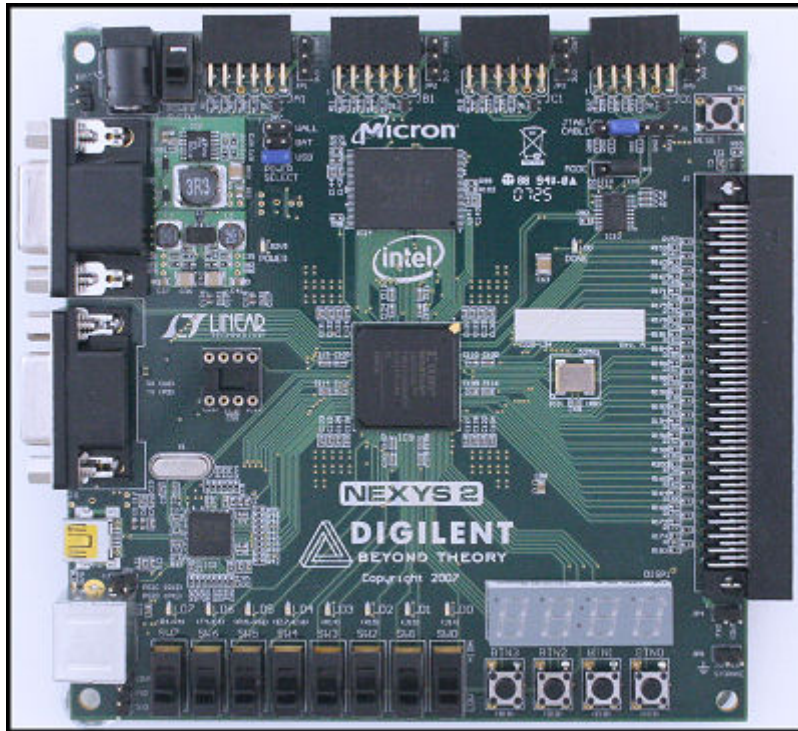


Getting Started With the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack A Beginner's Tutorial



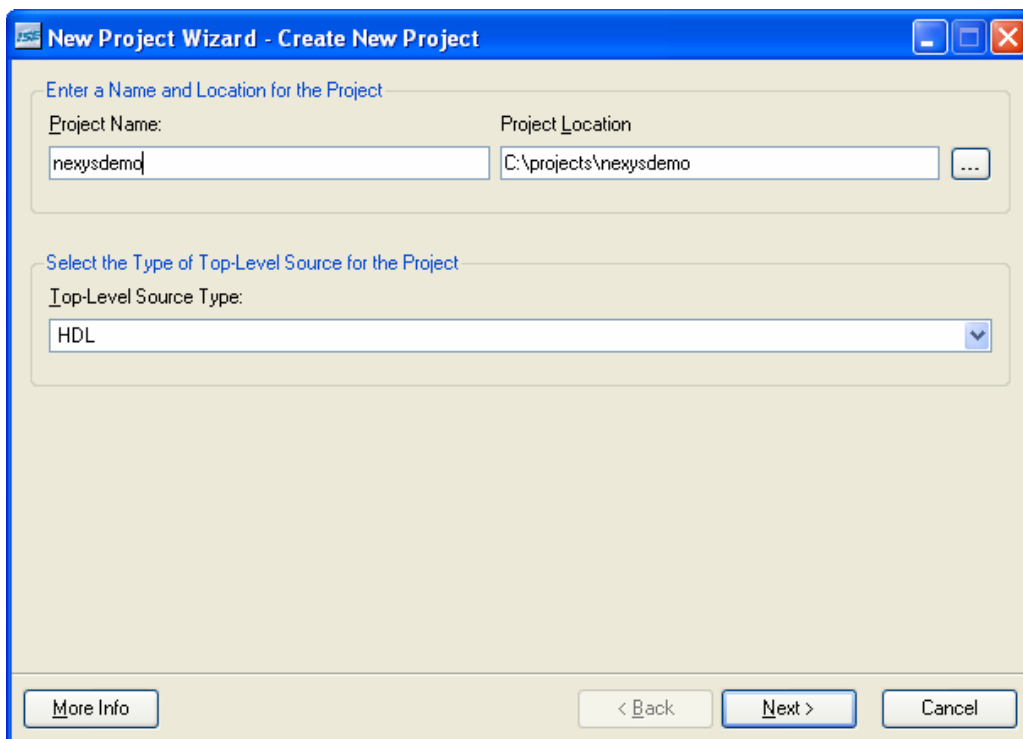
Goal

In this tutorial, we will go through the creation of a simple FPGA design using the Xilinx Spartan 3E on the Digilent NEXYS2 Board. By no means will all features of the board or software be covered, as this is intended to provide a starting point to those unfamiliar with the Xilinx tools and/or FPGA design in general. This tutorial will presume the reader has already installed Xilinx ISE, a free version of which is available at <http://www.xilinx.com/webpack>.

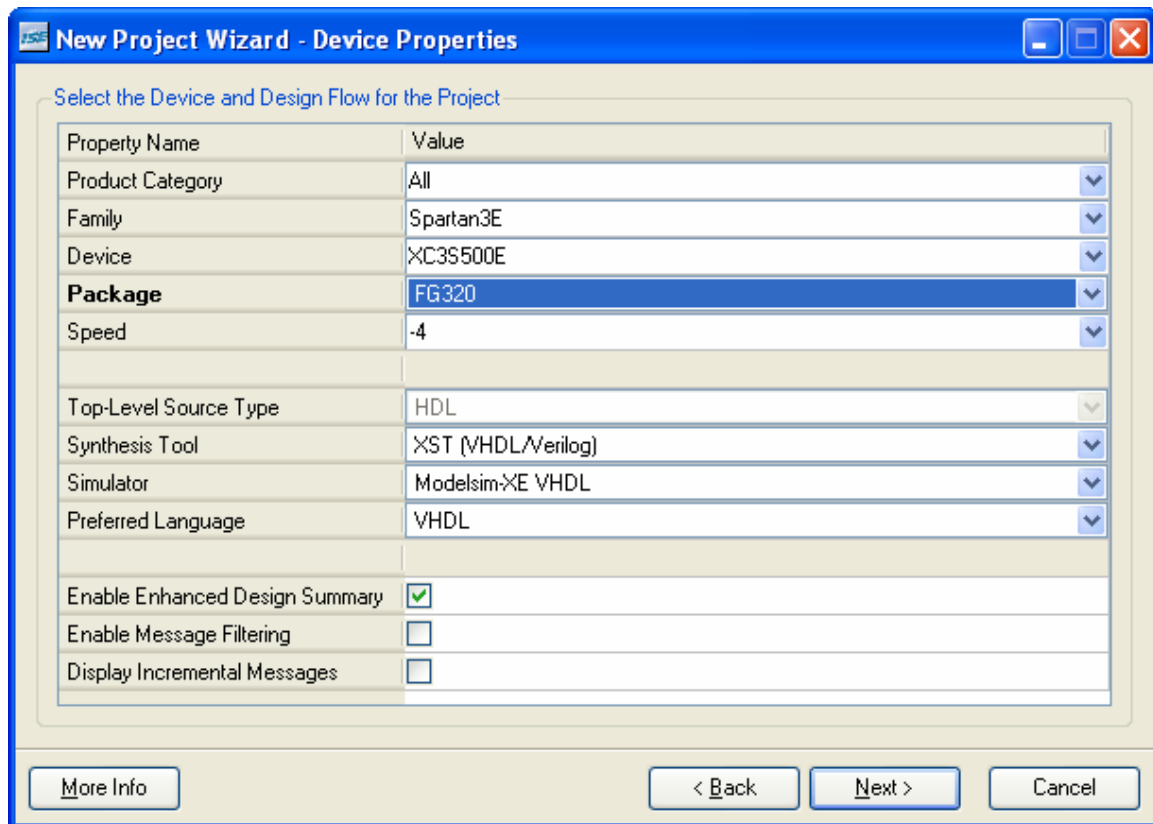
The Project

For this lesson, we will tackle the most basic of devices on the NEXSYS board, the slider switches and 7-segment LED Displays. Our goal is simple enough : use the 4 slider switches SW3 down to SW0 to represent a 4-bit binary number whose value can obviously be hex 0-F. We would like to place that value on the rightmost 7-segment display, and leave the rest blank.

To start the design, first start Xilinx ISE, and create a new Project with File → New Project. When specifying the project name and directory, pick a location without any spaces in the path (i.e. not the desktop) You will often find the Xilinx tools behaving very strangely with difficult to diagnose errors if you have spaces in the project path.



The next step will be to specify the FPGA you are targeting. This is obviously the one that is on the NEXYS2 board.



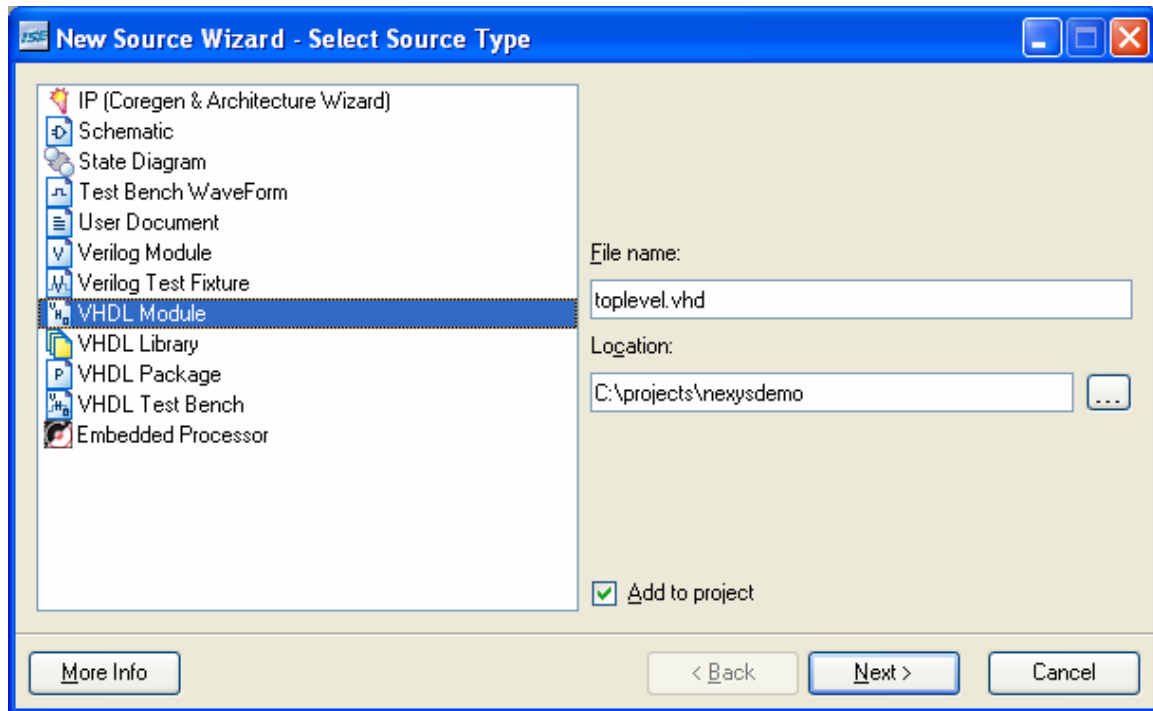
Skip over the next two pages, and click Finish to complete the project creation and begin the design process.

Entering VHDL

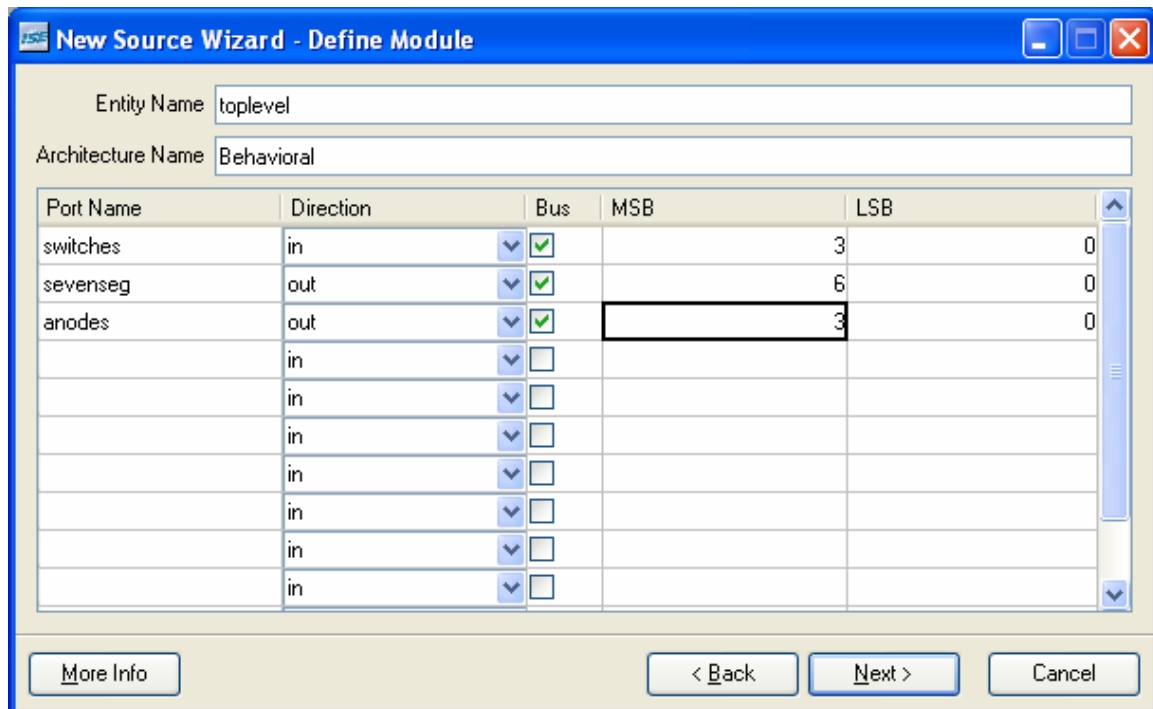
To describe the design, we will add one simple VHDL file to our project. Though this can be done with any text editor, it is convenient to allow the ISE tools to create the framework for us since we are starting the file from scratch. Adding an existing file is similarly easy, and can be done with “Project → Add Copy of Source”.

Select “Project → New Source”, and fill out the dialog telling the tools to create a new VHDL module with the name “toplevel.vhd”

Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack



On the next page, you will be prompted to enter the ports of your new module. This is strictly optional, and saves you the typing of creating the VHDL entity manually. Whether you fill this out or not, the resultant text file will be totally under your control for later editing.



Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack

We have entered the ports which will be placed into our entity in the VHDL file : the switches as our inputs, and the seven segments to drive on the display as our outputs. Furthermore, we have included the port “anodes” which will be used to enable / disable each of the 4 displays individually. The resultant VHDL file looks essentially like this :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity toplevel is
    Port ( switches : in  STD_LOGIC_VECTOR (3 downto 0);
          sevenseg  : out STD_LOGIC_VECTOR (6 downto 0);
          anodes    : out STD_LOGIC_VECTOR (3 downto 0));
end toplevel;

architecture Behavioral of toplevel is

begin

end Behavioral;
```

Xilinx has conveniently left a place for us to insert the architecture of our design, which will be two very simple statements :

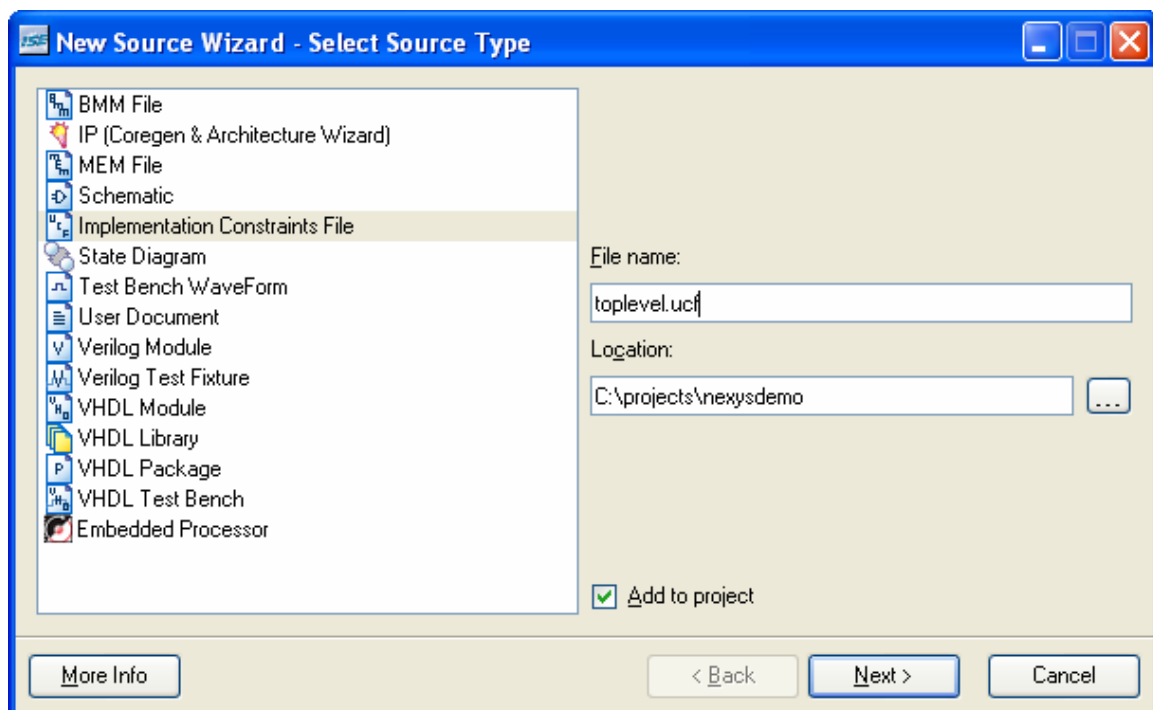
```
    with switches select
        sevenseg <=
            "1000000" when x"0" ,
            "1111001" when x"1" ,
            "0100100" when x"2" ,
            "0110000" when x"3" ,
            "0011001" when x"4" ,
            "0010010" when x"5" ,
            "0000010" when x"6" ,
            "1111000" when x"7" ,
            "0000000" when x"8" ,
            "0010000" when x"9" ,
            "0001000" when x"A" ,
            "0000011" when x"B" ,
            "1000110" when x"C" ,
            "0100001" when x"D" ,
            "0000110" when x"E" ,
            "0001110" when others;

    anodes <= "1110";
```

After entering the VHDL, save the file, and Check the Syntax for errors by clicking on “Check Syntax” under “Synthesize” in the Processes Pane.

Assigning Pins

The next task is to assign our input and output ports to the correct locations. If we neglected this step, the pins would be assigned by the Xilinx software to random locations, which would not be good. To do this, we must first create one more new file, the “Universal Constraints File” (.ucf). This will be done with Project → New Source as before:



Now that we have a constraints file for the project, we can easily assign pins graphically using “Assign Package Pins” under the “User Constraints” section in the Process Pane. This can also be entered by double clicking on the .ucf file that you just created from within Project Navigator. For this exercise though, we are going to edit the .ucf file in text, to get a good feel for how simple it is.

Single Click on your ucf file from within Project Navigator, and then Select “Edit Constraints (Text)” from the Process Pane as below. Then you can edit the constraints as shown below. To figure out the correct pin locations, consult the NEXYS2 manual. Even simpler, you may notice that most pins of interest are printed on the board surface.

```
NET "anodes<0>" LOC = "F17";  
NET "anodes<1>" LOC = "H17";
```

Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack

```
NET "anodes<2>" LOC = "C18";
NET "anodes<3>" LOC = "F15";

NET "sevensseg<0>" LOC = "L18";
NET "sevensseg<1>" LOC = "F18";
NET "sevensseg<2>" LOC = "D17";
NET "sevensseg<3>" LOC = "D16";
NET "sevensseg<4>" LOC = "G14";
NET "sevensseg<5>" LOC = "J17";
NET "sevensseg<6>" LOC = "H14";

NET "switches<0>" LOC = "G18";
NET "switches<1>" LOC = "H18";
NET "switches<2>" LOC = "K18";
NET "switches<3>" LOC = "K17";
```

Save the file, now we are ready to try the design on the FPGA.

Synthesis, Place+Route, Programming File

The next step in the process is to have the Xilinx ISE create a programming file for you. The whole process can be completed by clicking once on your top-level design in the Sources Pane, followed by a double-click on "Generate Programming File". This will run the whole process, and should complete as in the screenshot below.

The screenshot displays the Xilinx ISE Design Summary window for a project named 'NEXYSDEMO'. The interface is divided into several panes:

- Sources:** Shows the project hierarchy with 'toplevel' selected.
- Processes:** Lists various design processes, with 'Generate Programming File' highlighted.
- Design Summary:** Provides a high-level overview of the design.
- NEXYSDEMO Project Status:** Shows the current state as 'Programming File Generated' with no errors or warnings.
- NEXYSDEMO Partition Summary:** Indicates that no partition information was found.
- Device Utilization Summary:** A table showing logic utilization and distribution.
- Performance Summary:** Shows a final timing score of 0 and routing results as 'All Sizes Completely Routed'.
- Detailed Reports:** A table listing various reports such as Synthesis, Translation, Map, Place and Route, Static Timing, and Bitem.
- Console:** Displays the execution log, including the message 'Process "Generate Programming File" completed successfully'.

Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	7	9,312	1%	
Logic Distribution				
Number of occupied Slices	4	4,656	1%	
Number of Slices containing only related logic	4	4	100%	
Number of Slices containing unrelated logic	0	4	0%	
Total Number of 4 input LUTs	7	9,312	1%	
Number of bonded JTAGs	15	232	6%	
Total equivalent gate count for design	42			
Additional JTAG gate count for IODEs	720			

Final Timing Score:	Pinout Data:
0	
Routing Results:	Clock Data:
All Sizes Completely Routed	
Timing Constraints:	
All Constraints	

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Feb 6 11:44:29 2008	0	0	0
Translation Report	Current	Wed Feb 6 11:44:34 2008	0	0	0
Map Report	Current	Wed Feb 6 11:44:38 2008	0	0	2 Info
Place and Route Report	Current	Wed Feb 6 11:44:44 2008	0	0	1 Info
Static Timing Report	Current	Wed Feb 6 11:44:47 2008	0	0	3 Info
Bitem Report	Current	Wed Feb 6 11:44:52 2008	0	0	0

Configuration

Now that the programming file has been generated, it is time to load your design into the FPGA, and test it out. This can be done in many different ways depending on your available equipment and preferences. In this tutorial, we will discuss a few of the most likely used.

Configuration Using a Xilinx Parallel (or other) Cable via. the JTAG port.

If you have a Xilinx compliant programming cable (such as the one used with the Xilinx S3 Starter Kit), this is an appropriate method. If you would like to program via. the USB port, skip this section.

First, connect your programming cable to both your PC and to the NEXYS2 board JTAG header (J5). Make sure when connecting to use the correct orientation. The labels on the cable and board will assist in connecting VCC to 3V3 and GND to GND.

Next, move the blue jumper on JP9 to the left position, going between the “JTAG” and middle pin. Make sure that the NEXYS board is powered, either via. USB, or AD/DC adapter. The power LED should be lit.

The software that we will use for configuring the FPGA is named IMPACT, and is built into ISE. You can link directly to IMPACT from the Project Navigator by clicking on “Configure Device (IMPACT)” in the process pane. One can also start IMPACT directly, which we will do here to gain some insight into the various steps.

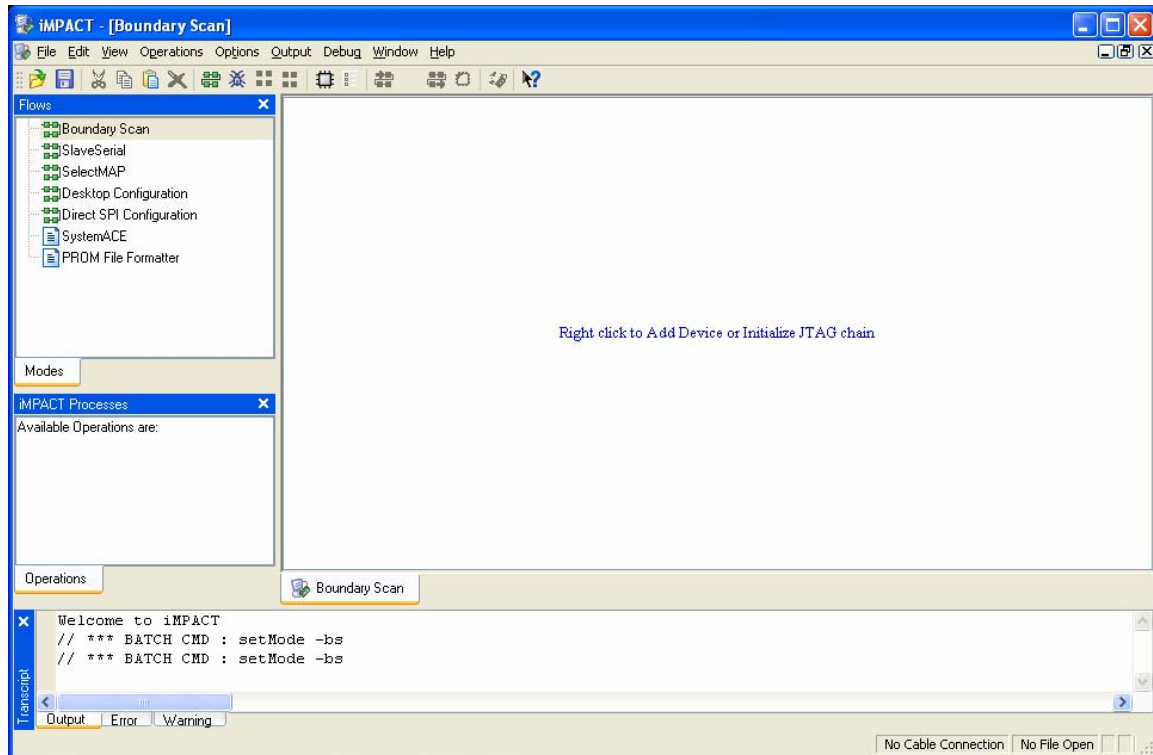
From the Windows Start Bar :

Start → Programs → Xilinx ISE → Accessories → IMPACT

Cancel the first prompt that comes up, as we will do everything manually.

Double Click on “Boundary Scan” to put IMPACT into JTAG mode

Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack

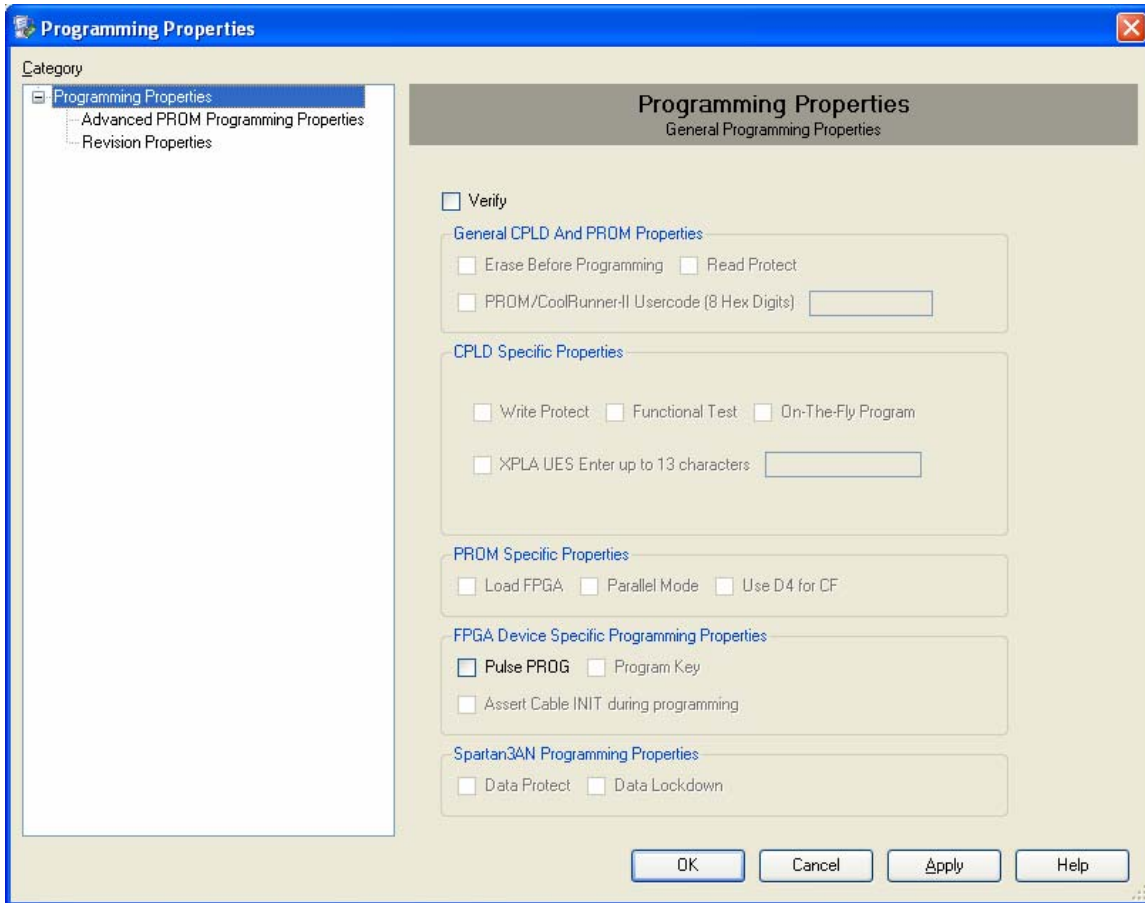


Right Click in the White Center Area and Select “Initialize Chain”. Pictures of your FPGA and a XCF04S configuration ROM should appear, and the tools should start asking for files to associate with each.

Browse to your project directory, and select “toplevel.bit” for the FPGA, you may click “Bypass” when it asks for files for the configuration ROM (XCF04s).

Next, click, then Right Click on the FPGA (xc3s500e) and select Program. Be sure that “Verify” is not selected in the box below.

Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack



Click on “Ok”

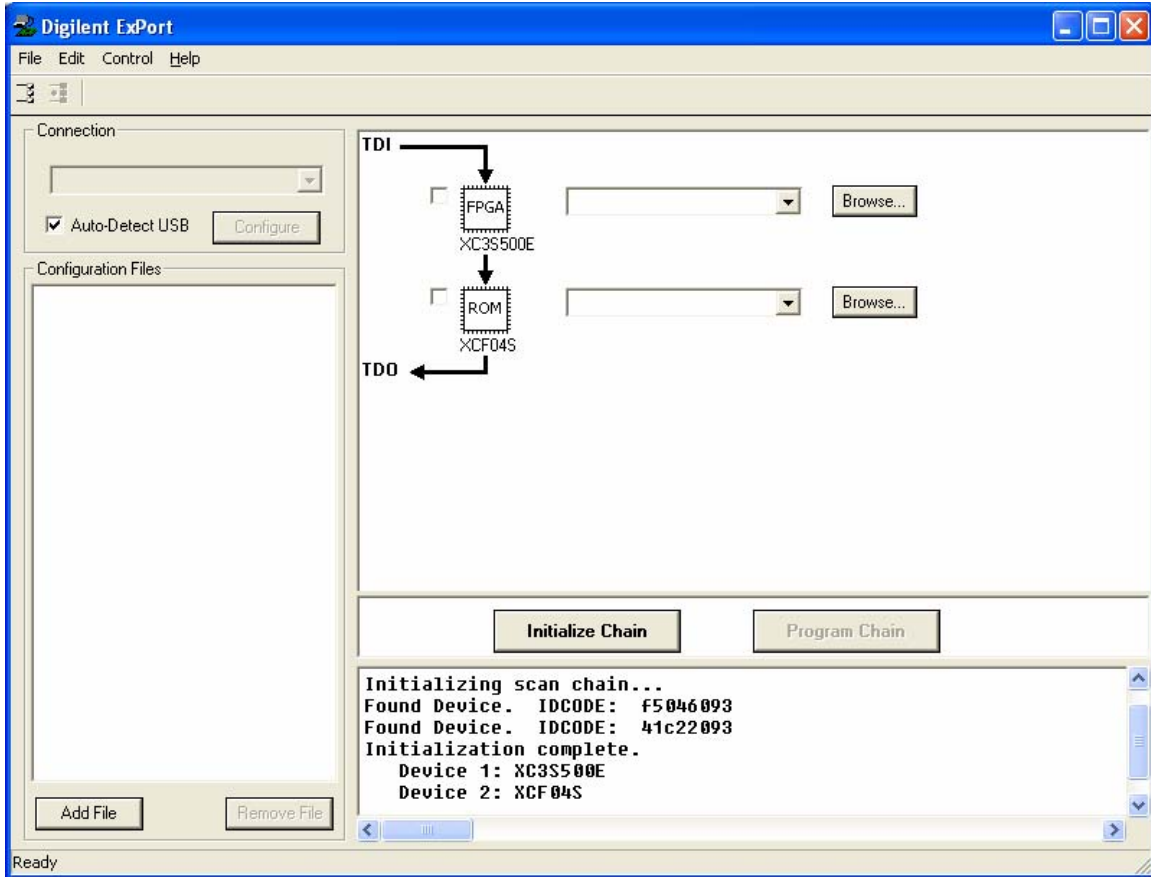
After the process completes, the FPGA should be configured as your simple design, and should be operational.

Configuration Using the USB port and Digilent ADEPT Software Suite

For students without a Parallel port or Xilinx Cable, configuration of the FPGA is done via Digilent’s own tool : ADEPT. This can be downloaded from the Digilent website.

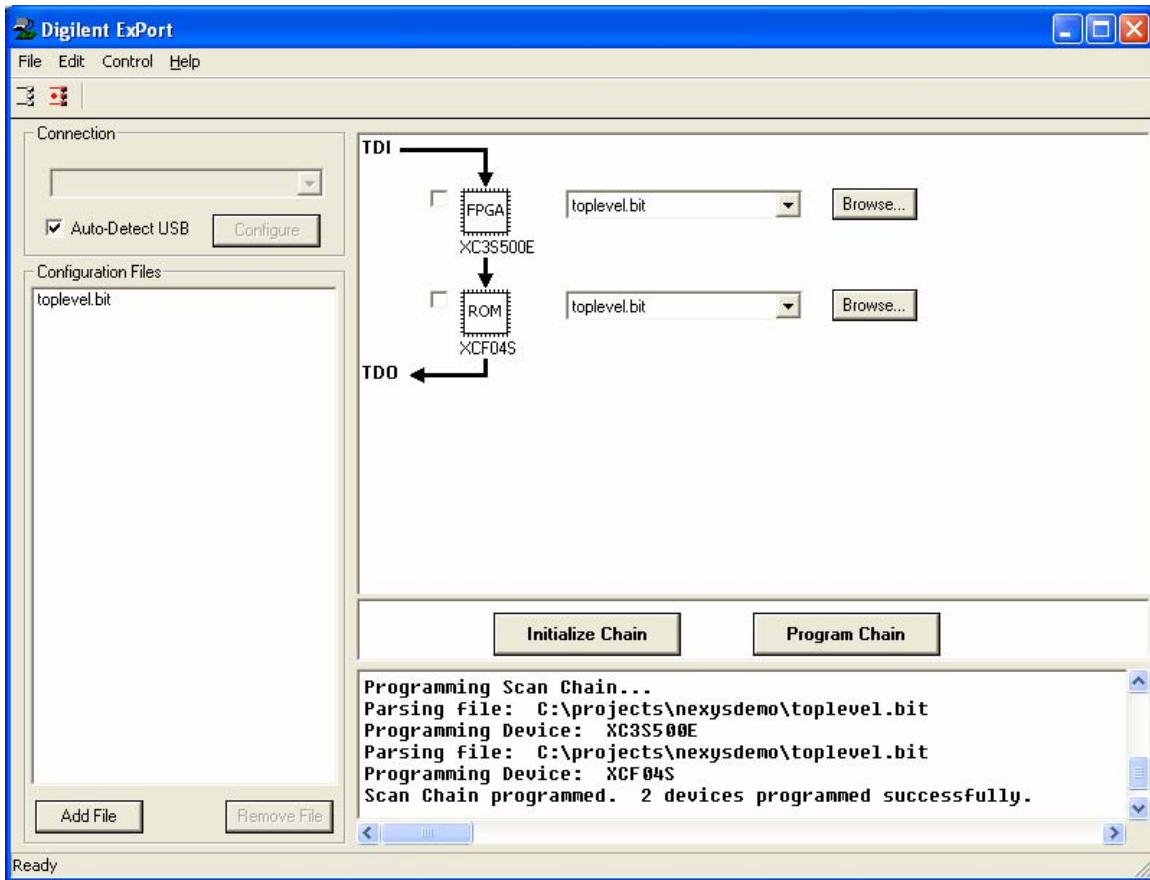
While the USB is attached, click on “Initialize Chain” to identify the devices on the board.

Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack



To program the FPGA, browse to the configuration file and select your toplevel.bit. Similarly, if you would like to program the board to start on power-up program the PROM (XCF04S) with the file as well..

Getting Started with the NEXYS2 Spartan 3E Kit and Xilinx ISE Webpack



Click “Program Chain” and wait. When configuration is complete, your FPGA should be operating as your VHDL specified. If you would like the board to run with this configuration on power-up, just make sure that the Mode Jumper (JP9) is set in the “ROM” position.