

# Spartan 3 FPGA Tutorial



June 2005

Revision A

Prepared by: David Fritz

Digital Logic Design

Oklahoma State University

## Overview

This document serves as an example based tutorial to programming the Spartan 3 FPGA included in your parts kit using Xilinx ISE Tools. A Basic working knowledge of how to create schematics in Xilinx is required.

The Spartan 3 FPGA can be programming directly from the Xilinx ISE package, greatly simplifying and expediting the design process. Programming the FPGA using a schematic, state diagram, or verilog module is a universally simple and straightforward process.

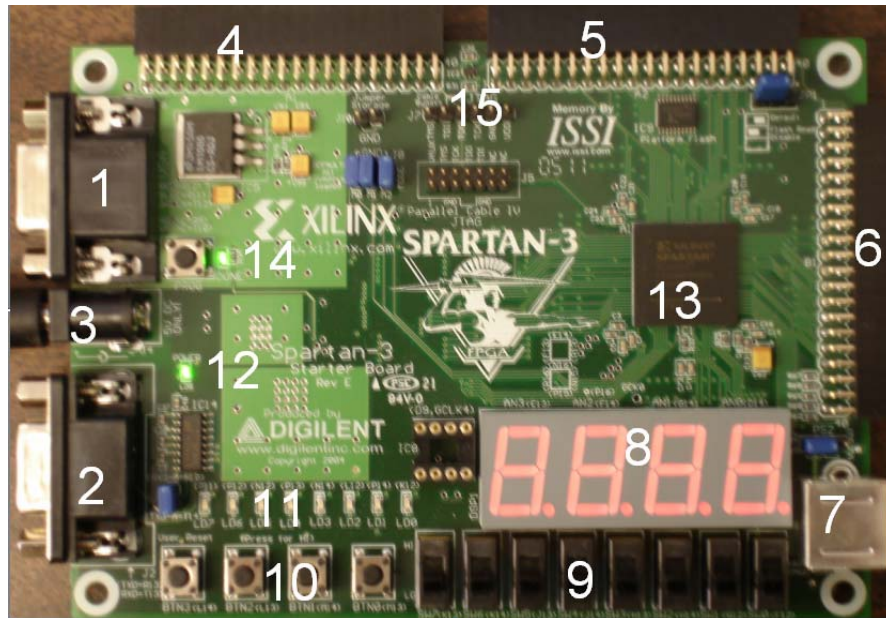
## What you need

In order to complete this tutorial you will need the following skills and resources:

1. Xilinx ISE Software (available in lab and on the web [see Xilinx Tutorial])
2. Computer (available in lab).
3. Digilent Spartan 3 FPGA (available in your parts kit).
4. Digilent Spartan 3 FPGA Reference Manual -  
<http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD-rm.pdf>

## Part I – FPGA Overview

The Spartan 3 FPGA board is a robust board containing many features. A list of key features and their location on the board is listed below, and all of these features are explained in great detail in the manual provided with the FPGA.

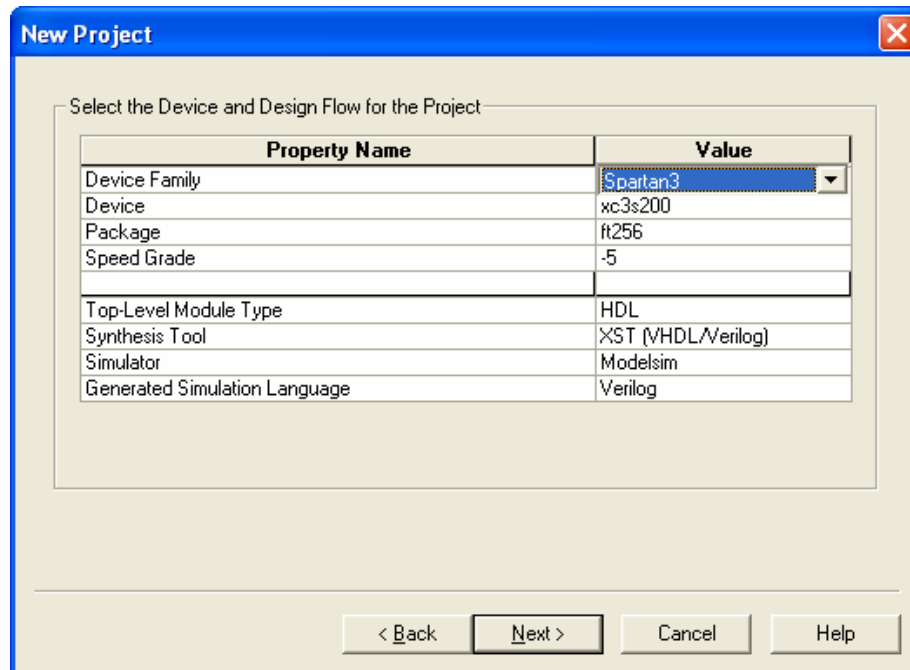


- |                             |   |
|-----------------------------|---|
| 1. VGA (HD-15) Monitor Port | 10. Buttons (4)                                   |
| 2. 9-pin (DB-9)             | 11. LEDs (8)                                      |
| 3. Power Connector          | 12. Power LED                                     |
| 4. A1 Expansion Port        | 13. Spartan 3 FPGA Core                           |
| 5. A2 Expansion Port        | 14. Program LED (Lit when the FPGA is programmed) |
| 6. B1 Expansion Port        | 15. JTAG Port (used to program the FPGA)          |
| 7. PS/2 Port                |   |
| 8. Seven Segment Displays   |   |
| 9. Switches (8)             |   |

You will notice that next to each component on the board, a label and pin number is printed on the board. For example, the leftmost button has “BTN3 (L14)” printed on it, the “L14” is the pin number we will use in Xilinx. There are also 3, 40-pin expansion ports along the edges of the board that we can use for wiring the FPGA to external devices. Note that not all of these pins can be used (see the Spartan 3 Reference manual, page 49 for a complete list).

## Part II – Creating a Test Circuit

Before we can create a test circuit, we need to make sure our project is setup correctly. When creating a new project, make sure to choose the options as shown below.



**Device Family:** Choose Spartan 3, the device we will be using.

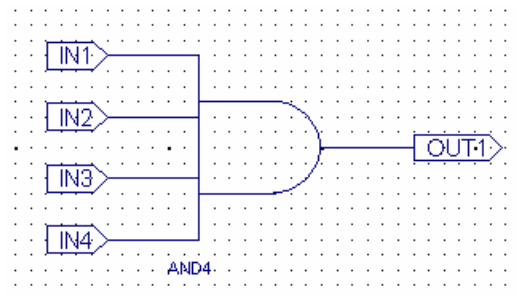
**Device:** xc3s200, the specific Spartan 3 device we use. This is actually printed (very small) on the FPGA core.

**Package:** ft256, this is the package type of our device (Ball Grid Array, 256 pins)

**Speed Grade:** The speed grade for this device is -5.

Complete the new project wizard and create a new schematic (see Xilinx Tutorial).

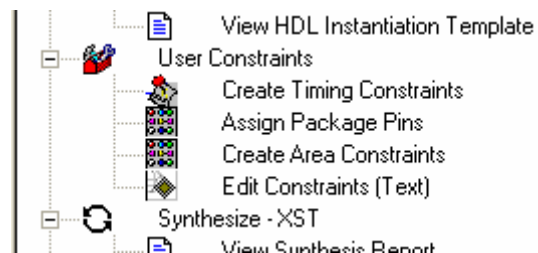
We will create a very simple design, a 4-input AND gate, as shown.



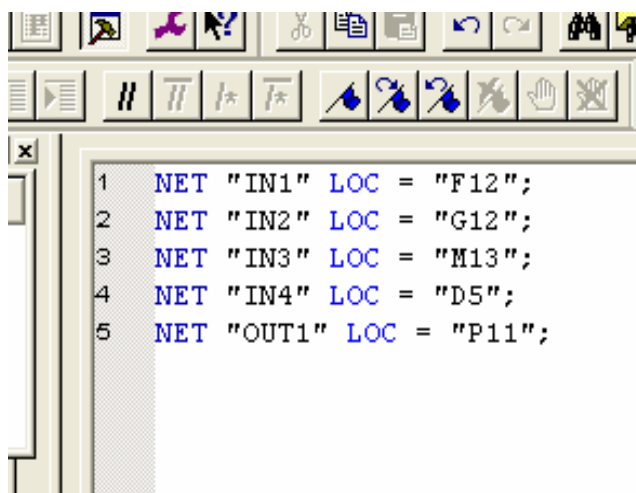
Draw the gate and label the I/O markers as shown. The names are important to remember, as we will see shortly. Save your schematic and return to the main Xilinx ISE window.

With our schematic drawn, we now want to tell Xilinx what FPGA pins to attach to the I/O markers of our design. For the sake of using different types of input, we will use 2 switches, 1 button, and an external input for the inputs of the AND gate. We will also use one of the on-board LEDs for the output.

Within the main window, with our schematic file selected, choose the “Edit Constraints [Text]” option from within the “User Constraints” submenu. Press “Yes” to the question about adding a UCF file.



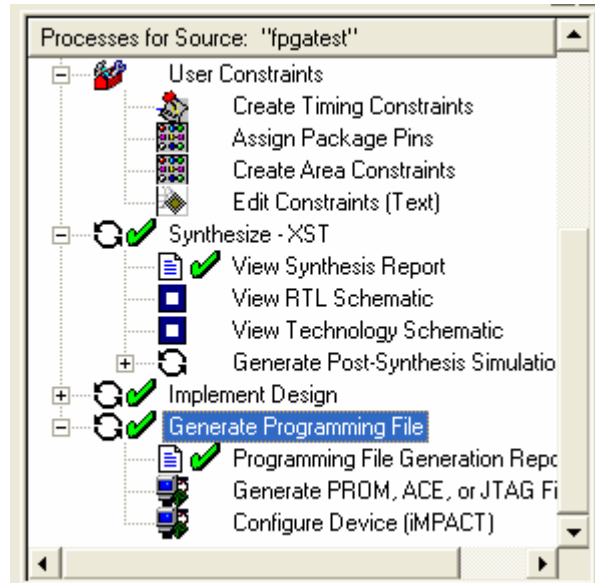
You should be presented with an empty text window. This is where we will type in the relationships our I/O markers have with the FPGA pins. Type the constraints file as follows.



The pin names are tied to specific pin numbers using this format. You can see that the I/O marker from our schematic “IN1” is connected to the pin “F12” which is connected to the rightmost switch on the FPGA. “IN2” is connected to another switch (SW1). “IN3” is connected to the rightmost button. “IN4” is connected to an external input (according to the Spartan 3 reference manual, pin 5 on expansion connector A2).

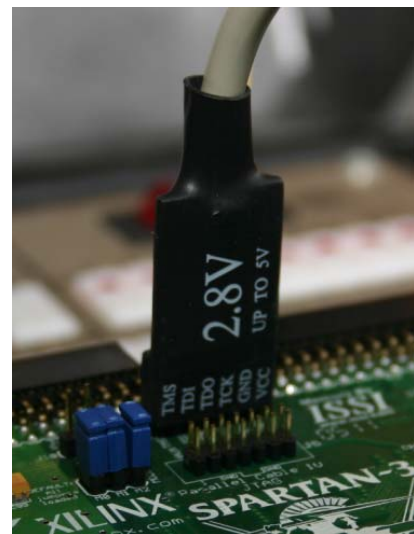
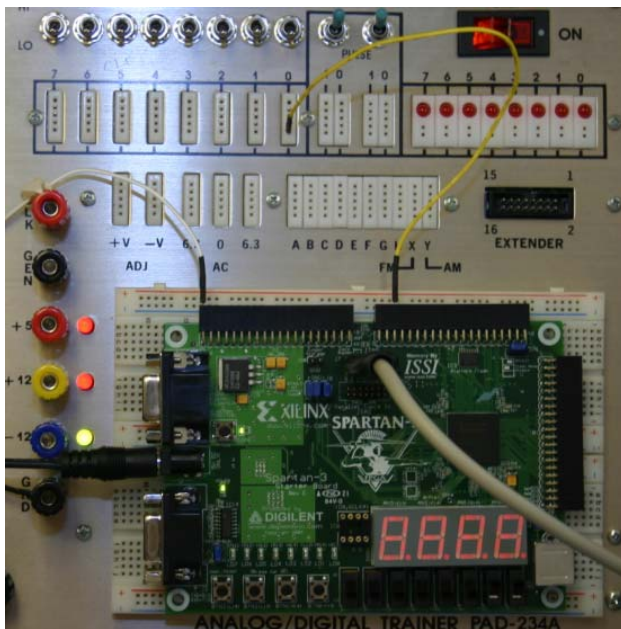
Save the constraints file and run the “Generate Programming File” option.

If all goes well, you should be rewarded with several green checkmarks. Yellow “warning” checks are sometimes okay as well. A red exclamation point means something went wrong.



## Part III – Programming the FPGA

Now hook up the FPGA to the PC using the supplied JTAG cable as shown below. We will also use a black box to supply our external switch. Hook one of the black box switches to our input on connector **A2** (at **pin 5**). Be sure to have a **common ground** between the FPGA and black box by connecting a wire from the ground on the black box to **pin 1** on any of the expansion connectors.

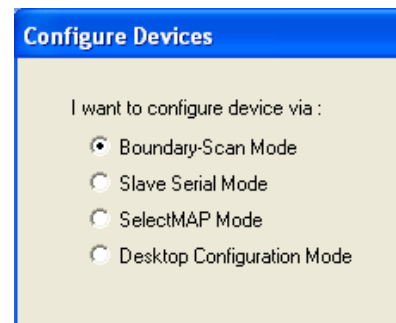


With the FPGA powered and ready, follow the steps to program it with your generated programming file.

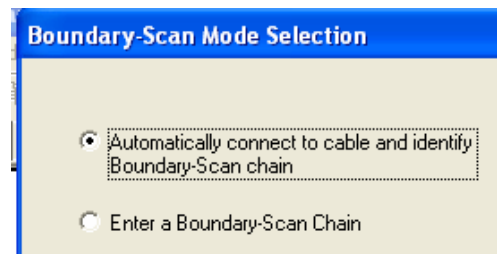
1. Start the iMPACT program by selecting it in the main Xilinx window under “Generate Programming File”.



2. When the “Configure Devices” dialog appears, select “Boundary-Scan Mode” and press “Next”.



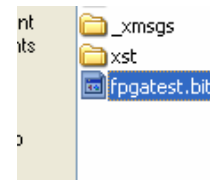
3. In the next dialog select “Automatically connect to cable” and press “Next”.



4. iMPACT should autodetect the FPGA and alert you that 2 devices were found, press “OK”.



5. When the select file menu appears, select the .bit file you generated earlier. You should have only one of these files in your project directory and it should have the same name as your schematic. Select that file and press “Open”.

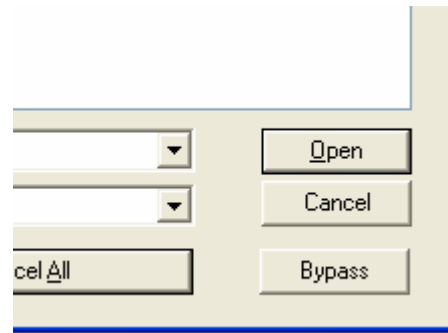




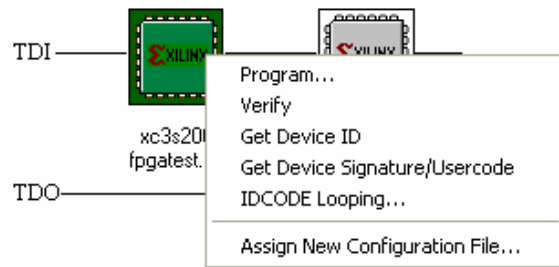
6. If iMPACT warns you about the “JTagClk” being changed, simply press “OK” and move on.



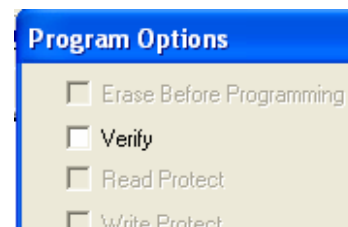
7. Another “Open File” dialog will appear. This is used for programming the Flash-rom on the board and will not be used here. Select the “Bypass” button on this dialog.



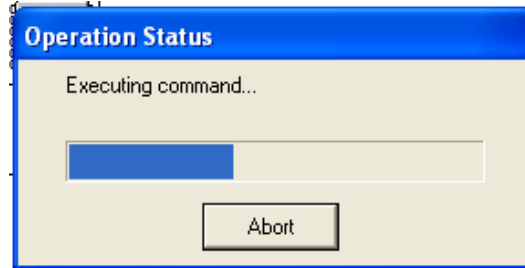
8. With the programming files assigned, right click on the leftmost device and select “Program..”



9. Within the Program menu, unselect the “Verify” check if it is checked. If you try to program this FPGA with verify turned on, the programming will fail.



10. Select “OK”. The FPGA will take a moment to program, and should return with a “Programming Successful” message.



xcf02s  
BYPASS

Programming Succeeded

At this point the FPGA should be programmed and ready. Test this by turning on all of our switches and button. You should see LED7 turn on.

