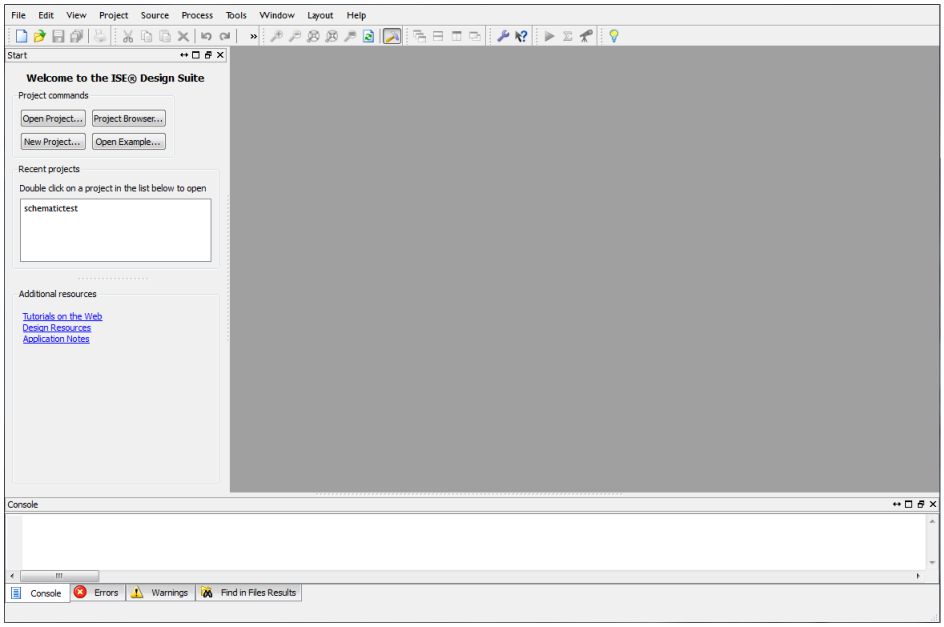
Tutorial on Spartan-3E Starter Board programming with Xilinx ISE

Physics 116B

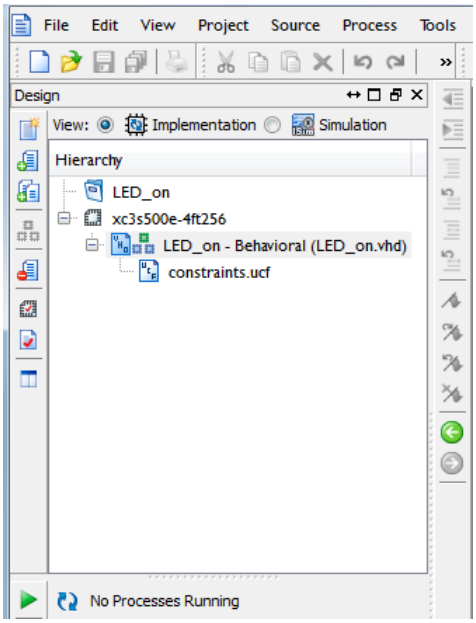
1 Introduction

The Spartan-3E Starter Board by Digilent features a programmable Spartan-3 FPGA (Field Programmable Gate Array) in addition to various inputs and outputs that allow us to interact with the FPGA. These inputes include slide switches, push-button switches, rotary push-button switch, LEDs, and a character LCD screen. In this tutorial, we will learn how to program the FPGA with the Xilinx ISE Project Navigator. The code used to program our Spartan-3 FPGA is written in VHDL (**V**ery High Speed Integrated Circuit **H**ardare **D**escription **L**anguage). Soon, we will see what a simple VHDL design looks like. In the meantime, keep in mind that a typical design written in VHDL is not similar to a program written in C++. Rather, VHDL is used to describe physical hardware, like logic gates, flip-flops, and clocks that interact concurrently in a circuit rather than sequentially. This distinction will become important as you learn how to correctly design your own system architecture. Let us begin the tutorial by opening up the ISE Project Navigator.

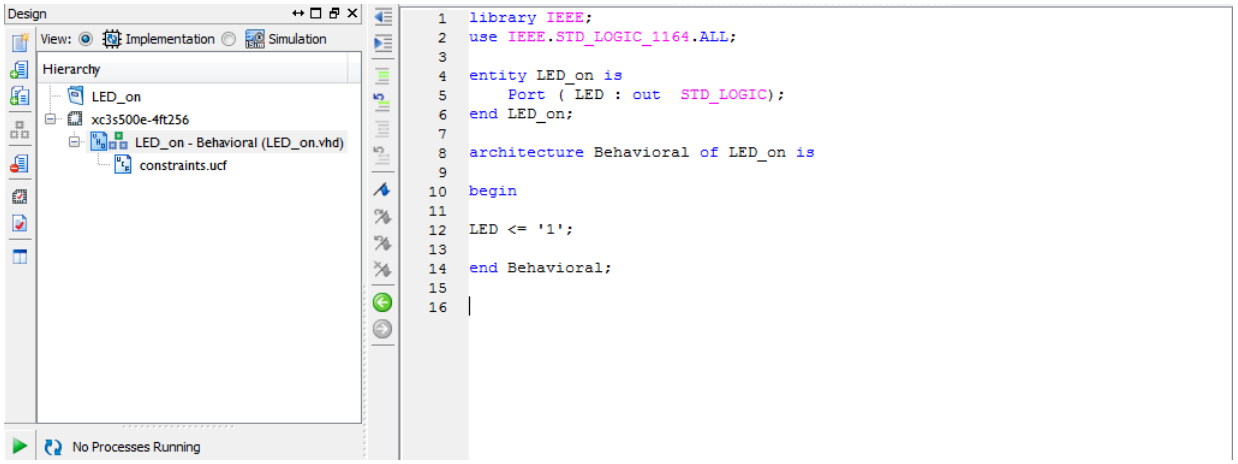


1.1 Opening a project in the Xilinx Project Navigator

Let's start by opening a very simple project in the Xilinx Project Navigator called "LED_on.xise". From the file menu open the project (you can download it from the SmartSite Resources center) and verify that it loads correctly. Double click on the highlighted line "LED_on - Behavioral (LED_on.vhd)" in the "Hierarchy" pane shown below.



The greyed out window should now fill itself with the "LED_on.vhd" code. Verify that your window matched up with the one shown below.



Let's take a look at the code, shown below, and try to understand what each line is doing here.

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

entity LED_on is
    Port ( LED : out  STD_LOGIC);
end LED_on;

architecture Behavioral of LED_on is

begin

LED <= '1';

end Behavioral;
```

The first line is specifying that we use the IEEE (Institute of Electrical and Electronics Engineers) library package. This declaration should not need to change for your codework in this class. The second line is a use statements specifying which specific package from the IEEE library we are using. For example, the *std_logic*_1164 package includes the *signal* type *std_logic*, which we use in our code. A signal type can be thought of as a data type in other languages you are familiar with.

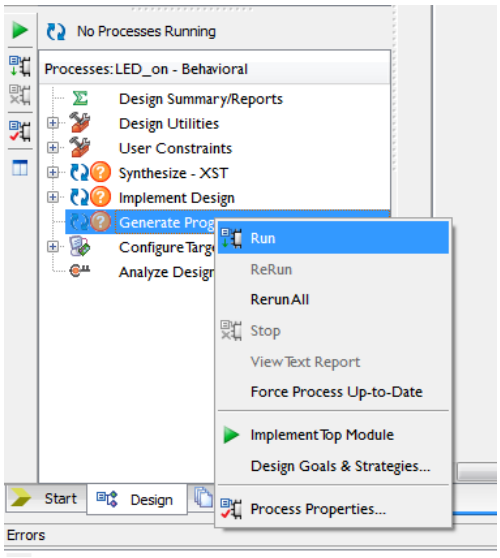
Next, comes the *entity* declaration. An entity can be any "black-box" with inputs and outputs. Your inputs and outputs are declared within the "Port()" statement. For our program, we have no inputs and only one ouput: LED, a single bit standard logic type. For multiple signals, you simply separate them with a comma in between.

After the entity declaration, we have the *architecture* of our entity. Inside the architecture declaration, we describe how our entity (or black box) actually functions. Here, the LED signal is simply assigned a value of "1", corresponding to a logical high value.

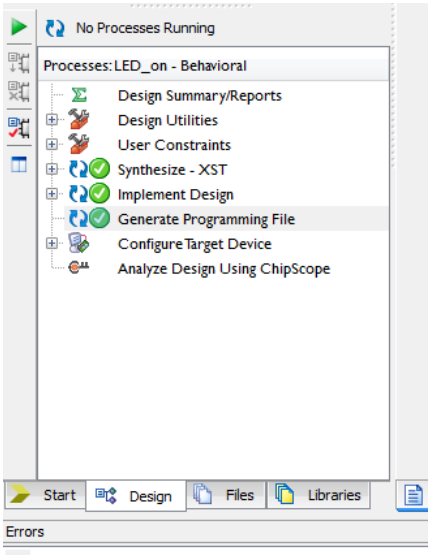
At this point, you should have good idea of what is going on in this code. If anything is confusing, call over you lab instructor and ask some questions. If you're ready to move on, turn the page and follow along with the tutorial.

1.2 Flashing the FPGA

We will now generate a programming file with our VHDL code and flash the FPGA chip with it. Go ahead and right click on "Generate Programming File" in the lower left pane of the ISE Project Navigator and click "Run".

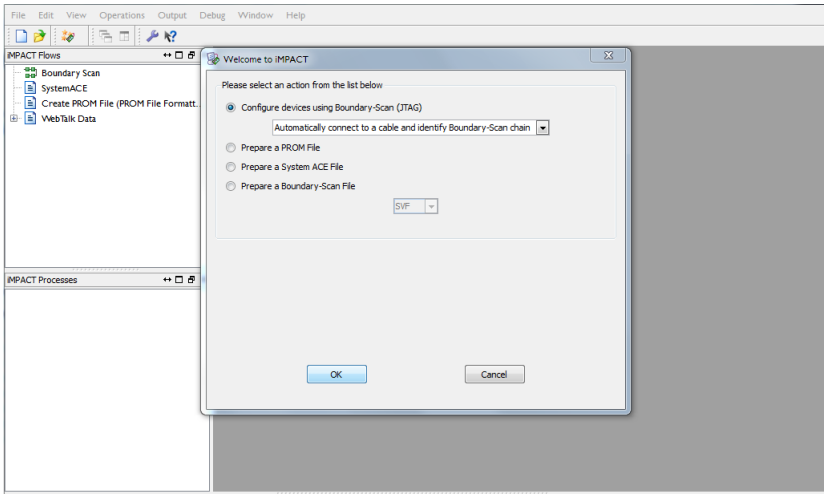


After some time, the program should have compiled successfully, leaving you with the following screen.

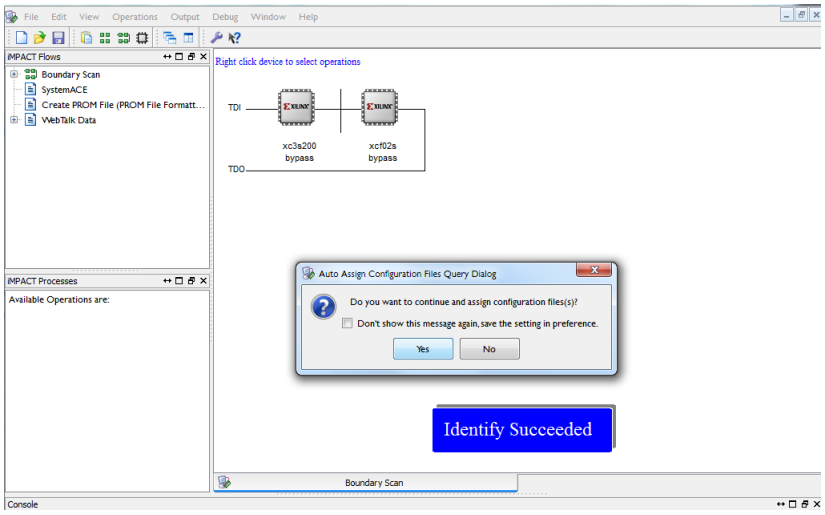


Then, right click on "Configure Target Device", click "Run", and press OK.

We should now have an ISE iMPACT window open. Make sure your Spartan-3 Starter Board is powered and has the USB-JTAG cable connected correctly (match the markings on the board with the markings on the cable). From the file menu, open new project. Press "Yes" and then "OK".

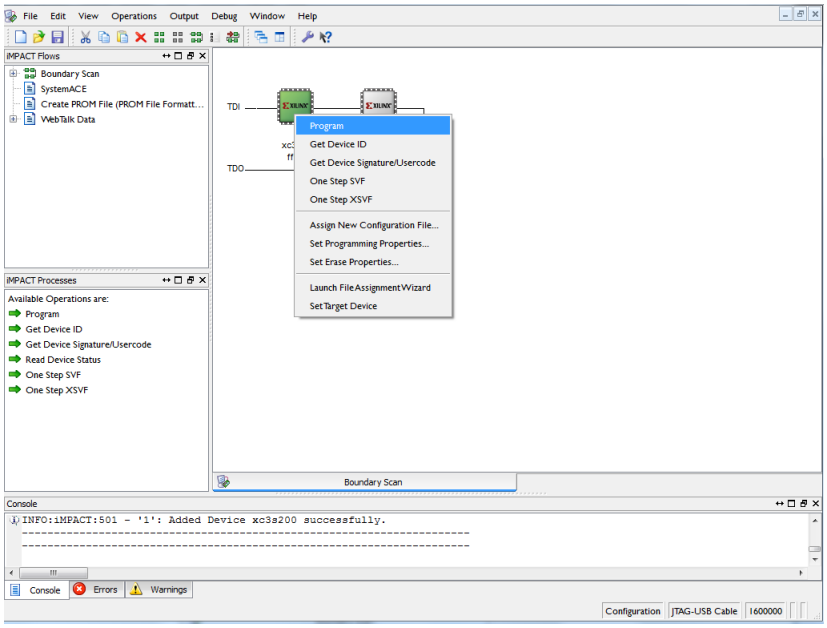


When you are then prompted for assigning a configuration file, press "Yes".



For the first assigment window, select "LED_on.bit" and press "Open. For the second assignment window, press "Bypass". Then press "OK" in the Programming Properties window.

You should now see two Xilinx chips connected together with “LED_on.bit” under the xc3s500e chip and “bypass” under the xcf02s chip. Go ahead and right click the xc3s500e chip and select “Program”.



The ISE iMPACT will execute your command and your board should come to life! Verify that the LED under “F12” is illuminated.

2 Assigment

Your task now is to complete the following:

- Modify your VHDL code so that the output of the LED corresponds to the logical AND, OR, NAND, NOR, XOR, and XNOR of slide switches 1 and 2. You will need to modify the constraints.ucf file by adding the locations of slide switches 1 and 2. Verify that the output gives you the correct results by comparing with a truth table for each logic statement.
- Now, modify your code to create a multiplexer than uses slide switch 0 to control the output from slide switches 1 and 2. That is, when slide switch 0 is in the "off" position, the LED should output the logical value of slide switch 1. When slide switch 0 is in the "on" position, the LED should output the logical value of slide switch 2.