

**ECE 4250/ 7250: VHDL and Programmable Logic Devices
Laboratory**

Lab #: 03

Lab Title: 4 Bit Full Adder Implementation on Xilinx Spartan-3

Group #: 5

Names: John Kelly | Zach Rump

Teaching Assistant Use Only:

Points Earned

Reasons for Deduction

Pre-lab:

Post Lab report:

Demonstration:

Final Lab Grade:

Comments to students:

Objective

The objective of Lab 03 is to become familiar with the Xilinx ISE and Spartan-3 FPGA by learning how to implement and synthesize a design onto the board. This lab uses a 4-bit full adder as an example project. The adder count is displayed on two seven segment displays.

Implementation

This lab is split up into multiple different files that all perform a specific function but work together as a whole. AnodeControl.vhd toggles cycles through each 7 segment essentially performing time division multiplexing. FullAdder.vhd describes the entity and architecture for a Full Adder, which is used in fig2_04.vhd to implement a 4-bit Full Adder. Dec_7seg.vhd describes which signals need to be sent to the 7-segment display for a given hexadecimal digit. The Adder4Bench.vhd file describes a test bench which is used to drive the 4-bit Full Adder. This test bench is used in conjunction with the Adder4Bench.ucf file in order to interface the hardware with the test bench. The end result is that the Spartan-3 hardware drives test bench which drives the 4-bit Full Adder. The LEDDisplay.vhd module basically ties everything together, and represents the highest level in the program hierarchy.

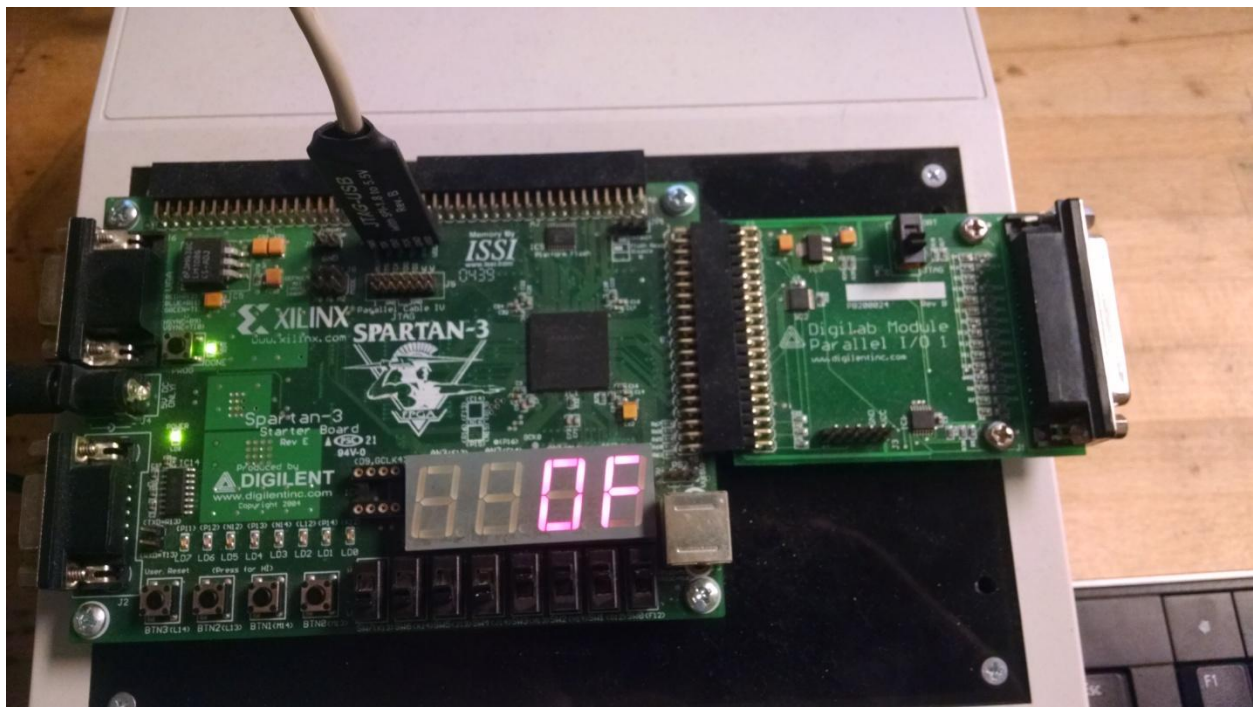


Figure 1: Demonstrates the synthesized code running on the Spartan-3 board.

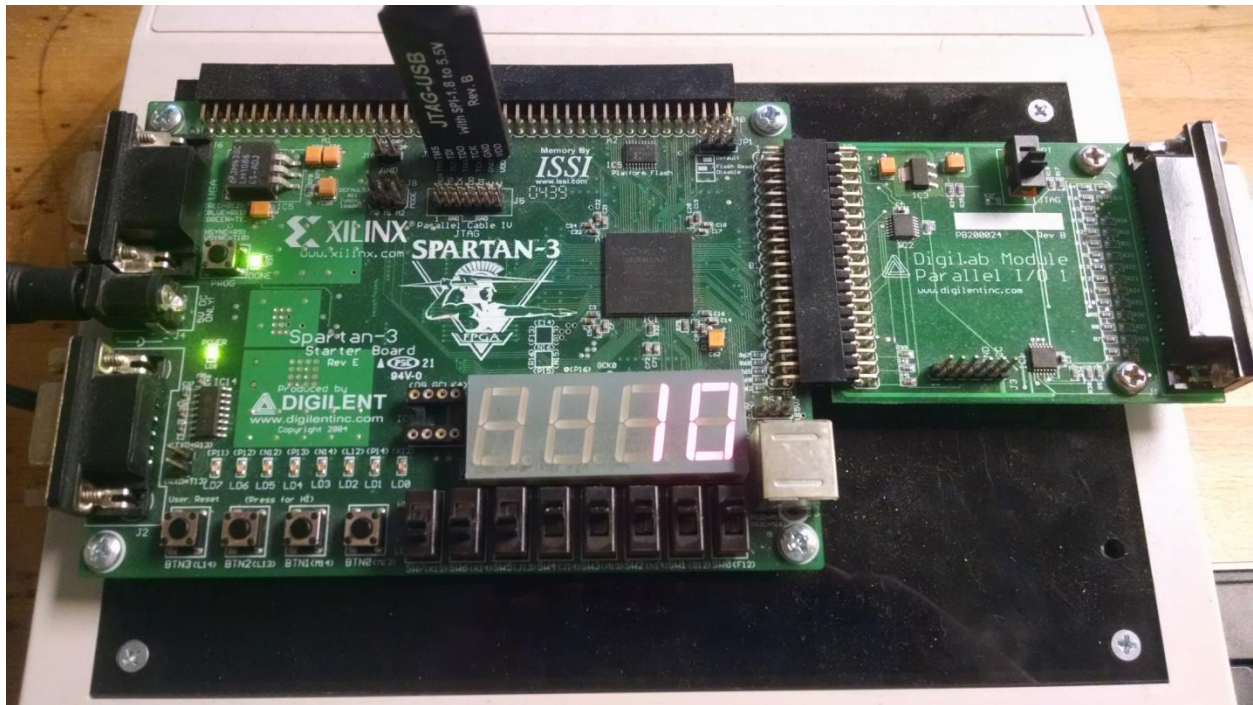


Figure 2: Demonstrates the functionality of the carry out and the transition from “0F” => “10”.

Post lab questions:

- i. **What is the function of the UCF file?**
The UCF file is essentially a netlist. It describes the connections that should be made between ports in the VHDL code and physical connections on the Spartan-3 FPGA.
- ii. **Explain the role of “NET”A<0>” LOC = “F12”;” from the UCF file.**
This maps the 0th bit of signal A in the VHDL code (Adder4Bench) to the FPGA pin F12 which is connected to SW0 (switch 0). I.e. The LSB of signal A in the adder is controlled by the physical switch SW0.

Conclusion

Other than minor issues like syntax errors, the main issue encountered in the lab was with the AnodeControl module. The first attempt at synthesizing worked OK, but the third LCD was turned on for some reason. The TA helped us figure out that this was occurring because of the ‘default’ else statement inside the AnodeControl process. The lesson here is to always initialize values and make sure to explicitly handle the ‘default’ cases when assigning values.