ECE 441

Advanced Digital Design and FPGAs

Lab 2

Evan Weiss

Daniel Sciortino

January 30, 2016

Honor Pledge: I pledge to support the honor system of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community, it is my responsibility to turn in all suspected violators of the honor system. I will report to Honor Council hearings if I am summoned.

**Introduction**

In this lab we became more familiar with I/O on the DE2-115. This lab was broken into 3 different parts; each part building on the previous. For all 3 parts of the lab we use the 18 side switches switches and 4 push buttons for input and actions. For the outputs we utilized the LEDR and LEDG arrays as well as the 8 on board 7-segment displays.

**Part 1**

For the first part of the lab we took the input from the 18 side switches and outputted the switch being enabled on the corresponding 18 red lights above it. Then for the 4 push buttons were displayed on the corresponding green led lights above it. There was no design work necessary because the VHDL for this part was provided.

**Fitter Report:**

|  |
| --- |
| Fitter Status Successful - Thu Jan 28 20:47:31 2016  Quartus Prime Version 15.1.0 Build 185 10/21/2015 SJ Lite Edition  Revision Name lab2\_1  Top-level Entity Name lab2\_1  Family Cyclone IV E  Device EP4CE115F29C7  Timing Models Final  Total logic elements 0 / 114,480 ( 0 % )  Total combinational functions 0 / 114,480 ( 0 % )  Dedicated logic registers 0 / 114,480 ( 0 % )  Total registers 0  Total pins 44 / 529 ( 8 % )  Total virtual pins 0  Total memory bits 0 / 3,981,312 ( 0 % )  Embedded Multiplier 9-bit elements 0 / 532 ( 0 % )  Total PLLs 0 / 4 ( 0 % ) |

The only utilization present is that of the Total Pins. 18 pins for switches, 18 pins for the LEDRs, 4 pins for the pushbuttons and the remaining 4 for the 4 LEDGs we used. This is because the design required no logic gates to implement, just connections from one pin to another, specifically the switches and push buttons to the LEDs. The FPGA routed these connections without using any logic elements.

**Part 2**

For the second part of the lab we were to take input from 16 of the side switches and show the corresponding hex values on the four leftmost 7-segment displays. Then we were to add these two numbers and display the sum on the remaining 4 7-segment displays. We used four switches to make a nibble and then displayed the nibbles hexadecimal value on one 7-segment display. To accomplish this we made individual processes for each 7-segment display that had a sensitivity list of the 4 bits corresponding to the input switches for that nibble. At the end of each process it would then output the nibble onto the 7-segment display in hexadecimal form.

In order to convert the 4 bit input to a 7 bit output corresponding to the 7-segment display we made a function. It would take 4 bits as the input, find the appropriate value in the case statement, and return the new 7 bit std\_logic\_vector. At the end of the function we had the return value inverted to correct for the 7-segment displays being common anode and thus active low.

Once these small sections of the code were written we had to add the two bytes together and output the resultant on to the last 4 seven segment displays. The issue that we ran into was the adding the 2 8 bit std\_logic\_vectors was too short for the 4 remaining seven segment displays. From this we had to sign extended each operand to create the required length. Because we treated our inputs as always positive rather than 2’s compliment numbers this was easily accomplished by concatenating eight zeros to the left side of our operands.

**Fitter Report:**

|  |
| --- |
| Fitter Status Successful - Thu Jan 28 21:22:14 2016  Quartus Prime Version 15.1.0 Build 185 10/21/2015 SJ Lite Edition  Revision Name lab2\_2  Top-level Entity Name lab2\_2  Family Cyclone IV E  Device EP4CE115F29C7  Timing Models Final  Total logic elements 51 / 114,480 ( < 1 % )  Total combinational functions 51 / 114,480 ( < 1 % )  Dedicated logic registers 0 / 114,480 ( 0 % )  Total registers 0  Total pins 86 / 529 ( 16 % )  Total virtual pins 0  Total memory bits 0 / 3,981,312 ( 0 % )  Embedded Multiplier 9-bit elements 0 / 532 ( 0 % )  Total PLLs 0 / 4 ( 0 % ) ` |

For part 2 of the lab we had only 51 total logic elements used for the function and the overall architecture. The pins used increased to account for using the 7-segment displays rather than just the LEDs.

**Part 3**

For this final part of the lab we had to build on the single addition function with more arithmetic and bitwise functions. We were required to have addition, subtraction. exclusive-or, AND, OR, Multiplication, and division between the two 8 bit inputs. This required a separate process with a sensitivity list for 3 of the push buttons which were used to select the calculation performed. When a push button or multiple push buttons were activated the system would go into a process that would change the mathematical operation applied. For all of the operations, except the multiplication it was required to have sign extensions to have the appropriate length of bits. When the operation for division was required the process had to verify to see if the user was trying to divide by zero. If they were the last 4 7-segment displays would go dark. This would occur until the dividing value was greater than 0 or the operation was changed. To achieve this we had a if , else, flow control that would check the divider. This BlankOut signal was a boolean, and if the divisor was equal to 0 then it would be set to true. This BlankOut signal then would be passed into the hex\_display function, if the value was true it would return the inverse of “00000000”.

**Fitter Report:**

|  |
| --- |
| Fitter Status Successful - Thu Jan 28 21:26:52 2016  Quartus Prime Version 15.1.0 Build 185 10/21/2015 SJ Lite Edition  Revision Name lab2\_3  Top-level Entity Name lab2\_3  Family Cyclone IV E  Device EP4CE115F29C7  Timing Models Final  Total logic elements 462 / 114,480 ( < 1 % )  Total combinational functions 462 / 114,480 ( < 1 % )  Dedicated logic registers 0 / 114,480 ( 0 % )  Total registers 0  Total pins 86 / 529 ( 16 % )  Total virtual pins 0  Total memory bits 0 / 3,981,312 ( 0 % )  Embedded Multiplier 9-bit elements 1 / 532 ( < 1 % )  Total PLLs 0 / 4 ( 0 % ) |

Because of the logic that was required for the function, process, case and overall architecture the number of logic elements is almost 9 times that of part 2. Quartus decided to use one of the embedded 9- bit multiplier for the multiplication function rather than make a combinational multiplier out of gates using logic elements

**Analysis**

The utilization of the FPGA increased from part to part with the added complexity of the added functionalities. The first part went from simply making connections from inputs to LEDs while the second and third part required the use of logic elements and even one of the embedded multipliers. All of these designs functioned without the use of clocks and thus there was no added complication there, just combinational logic.

**Summary and Conclusions**

The 3 parts of this lab were to first make connections to the inputs and outputs on the DE2-115 board using VHDL, secondly create greater functionality behind the inputs and outputs using VHDL to add two operands together and display their output on 7-segment displays, and finally to increase the functionality by using additional inputs to change the function performed on the two inputs. Each part built on the previous one adding a new functionality each time.

In this lab we worked on interfacing with the input and output capabilities of the FPGA and how they can work inside of our VHDL model. The use of the qsf file to handle the assignments made accomplishing this significantly easier by allowing us to use meaningful names of our choosing in the VHDL code while allowing the software to make the connections on the FPGA. This allowed us, the developers, to quickly move between designs and improve upon them, a valuable capability in prototyping a design.

This lab also demonstrated the ability for the software to take functional behavior models and turn them into gate equivalents rather than only working if we have a structural or gate equivalent model. This also is very valuable in the testing and altering of designs.