

Unsigned numbers calculator

Josue Llamas



February 5, 2016

CAl poly pomona

ECE 424L

Josue Llamas

ECE 424L

Dr. Sacco

2/5/2016

Unsigned Numbers Calculator

1. Theory

In this four-week lab, we learned how to implement an unsigned numbers calculator using the VHDL programming language and the Spartan 3E FPGA board. There were many parts to this project and it all starts with the basics.

Step 1 was basically learning how to output strings of text on to the FPGA board. Having the LCDI code in Verilog, we had to convert it to VHDL by making it a component in the module. Using the LCD character set, I was able to display my name onto the board. The next step was to implement the input interface for the calculator. In this block we used many counters to get the look up table running and get the WADD ports of the ram. The Write Address is used to identify where in the FPGA LCD display is the value going to be written in. RADD (read address) is used to read the address back from control module.

The input interface is connected to the Controller module which is then connected to the Display Interface. Now that they are connected, we made sure that it was working by outputting to the FPGA. The program cycles through the contents of the look up table and outputs to the FPGA. Using the push button, V4, we would de-bounce to the next address while the reset button, K17, would return the output address to “00000”.

Now that we have the contents of the look up table successfully outputting on the FPGA, we can now use those values to do simple mathematical operations. To do this, we have to create a Multiplier, Divider, Adder, and Subtractor with the use of full adders. Since each address in the FPGA board can only hold one character at a time, we implemented a binary to BCD converter, that way we can output a 2 or 3 digit answer onto the FPGA. Having the calculator connected to controller, we finally able to implement and test our unsigned numbers calculator.

1. Design (hardware description code and block diagrams)
   1. Components

|  |  |  |
| --- | --- | --- |
|  | **Components/diagrams** | **Page** |
| 1 | Top Module | 1 |
| 2 | UCF file | 4 |
| 3 | Block Diagram | 5 |
| 4 | Pb-Debouncer | 6 |
| 5 | 26-bit counter | 7 |
| 6 | 5-bit counter | 8 |
| 7 | 4-bit counter | 10 |
| 8 | Look up Table | 11 |
| 9 | RAM | 9 |
| 10 | Calculator | 12 |
| 11 | Multiplier | 14 |
| 12 | Divider | 15 |
| 13 | ADD/SUB | 16 |
| 14 | B2BCD | 17 |
| 15 | Controller | 19 |
| 16 | LCD in Vhdl | 22 |

1. Design Verification (test bench code and simulation results)
   1. Tests

|  |  |  |
| --- | --- | --- |
|  | Tests | Page |
| 1 | Multiplier Test | 23 |
| 2 | Divider Test | 25 |
| 3 | Add/Sub Test | 27 |
| 4 | Controller Test | 29 |

1. Testing Procedure

There are many factors that come to mind when testing this project. Making sure that our mathematical operator work was our first step and then making sure it would correctly output to the BCD converter was our next step. A lot of the testing was done in the main controller, trying to output on the correct addresses.

1. Results

Testing the calculator from the FPGA gave us these results for the following random operations.

Table1. Random Math Operations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sample | B | A | - | + | X | / |
| 1 | 9 | 9 | 18 | 18 | 081 | 01 |
| 2 | 9 | 2 | 07 | 11 | 018 | 04 |
| 3 | 4 | 3 | 01 | 07 | 012 | 01 |
| 4 | 3 | 4 | -- | 07 | 012 | 00 |
| 5 | 5 | 7 | -- | 12 | 035 | 00 |
| 6 | 6 | 6 | 00 | 12 | 036 | 01 |
| 7 | 8 | 1 | 07 | 09 | 08 | 08 |

1. Analysis of results

Based on the results, we can see that B-A doesn’t seem to output a value for cases in which the answer would be negative. 9-9 gave us 18 which is completely wrong. This was the second problem that the Subtractor gave us. We can see that the Subtractor isn’t working correctly 100% of the time. Adding, multiplying, and dividing all seem to work based on the table above.

1. Conclusion

In this lab, I learned a great deal of VHDL, I feel as though I have learned this language better than I learned Verilog. Building upon the project every week made this possible and it is greatly appreciated. The main problem I had when finishing up the project was programming the controller. At the beginning, I could get an operation on the FPGA but the answer would not display on the board. In an attempt to rectify this problem, we added more lines of code on the cases where we would add 1 to WADD. This attempt got us to output the answer on the first 3 FPGA addresses erasing the 3 inputs. With the amended controller block diagram, I was able to fix the answer output. Now we have everything displaying correctly on the LCD display. This fix also outputted the correct quotients.

Another problem, we had that we couldn’t figure out was involving the binary to BCD converter. I was unable to get the Adder and Subtractor to display an answer. After looking over the code various times, I finally noticed that it was missing an input for “sub”, which was the subtraction indicator. After adding this correction, the sum and difference were finally being displayed.

1. Suggestions

Only suggestion I have is to give us a short review on how the Xilinx software works or upload the ECE 205 Verilog Tutorial so that those who haven’t used Xilinx in a while could be up to speed.