Simple Calculator

Jacob Arabbo, Anthony Barbera, Quinn Krause, Dante Gomez
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
jacobarabbo@oakland.edu
abarbera@oakland.edu
quinnkrause@oakland.edu
dantegomez@oakland.edu

Abstract- **The contents of this report include the process and results of the creation of an unsigned 4-bit simple calculator. The calculator is capable of performing addition, subtraction, division, and multiplication with the use of Vivado and a Nexys A7 board. The results of the calculations performed are printed on the boards 7 segment display while the function performed by the calculator is controlled by switches on the board. The calculations are all performed in order with A as the first input and B as the second 4 bit input. All functions were in working use and any negative calculations would yield a "--" on the seven segment display. The calculator was created using information gathered from class lectures and lab sessions with supplementation of the groups own innovation in order to create a working final product.**

INTRODUCTION

This report focuses on the process and results of creating a calculator in vhdl code. In depth descriptions of the steps taken to create adding, subtracting, multiplying, and dividing functions of unsigned 4-bit values within our calculator will be discussed. The input of this calculator will be operated by the switches on the Nexys A7 and the outputs will be shown on the 7 segment displays. During multiple class periods we learned to add, subtract, and multiply binary numbers. Additionally, during our lab we made an adder and subtractor. Our group combined techniques with what we learned in class, and lab, to create a base for the simple calculator. In order to create more complex functions within our calculator, we had to learn how to code a two input multiplier, learn how to divide in binary, and how to code a divider. Even though this is a simple calculator compared to what is in wide use, in doing this project we understand more of how calculators work, and what design goes into them. With our code and Nexys board we can now have a basic calculator, that can be added on to include more complex operations.

METHODOLOGY

To start this project our group knew we wanted to use concepts discussed within lab and class sections. With this objective, we decided on designing a calculator. In designing a calculator, we took the adder and subtractor we learned from the lab, and concepts taught regarding multiplying binary numbers that we learned in class. We also knew we wanted to have two inputs, since this is what we did in our labs. In creating the calculator this way, we use concepts we have learned in class, the lab, and what we had to learn on our own. Originally, the divider was difficult to code. After figuring out how to display the remainder value, we found it to be a useful addition to our unsigned simple calculator. We also had the idea of creating signed bits, so that we could add, subtract, multiply, and divide with negative numbers. However, the multiplication and division of signed bits turned out to be harder than expected, so we stayed with unsigned numbers. The simplicity of this design led to a problem of its own. When subtracting a larger number from a smaller number, a general solution was required in order to stay consistent within our design. If the solution yielded a negative number, we outputted "--" on our seven segment display. This created a consistent solution to our problem that allowed our project to stay within the scope of its design.

A. *TOP HALF OF CIRCUIT*
(Discusses Figure 1 from Appendix)

As stated before, we had wanted to use two inputs for our project. To start, these two inputs are each inputted into components that add, subtract, multiply, and divide the inputs. These are then outputted on a clock cycle to the mux. The mux is controlled by selectors that are inputted through the board. The selectors are used to determine what input will be outputted to the decoder. The decoder takes the binary inputs and outputs the bcd equivalent.

B. *BOTTOM HALF OF CIRCUIT*
(Discusses Figure 2 from Appendix)

The bottom half of the circuit controlled the 7-segment displays and when to have them on or off. On the left side was a 0 to 2 modulo counter, where every tick increases the output by 1. The output is a 2 bit signal used for the select lines of the 2 MUX's that determine which digit and respective anode is to be active. This counter ran at 1.3ms so all displays can be seen simultaneously. The FSM is used to control which anodes will be allowed to be on, when the clock cycles through. We only want to display the zeros that are necessary. For example: if the result of the calculation is 1 then we want to display '1' instead of "001". On the contrary, if the result is 100, then we want to display those trailing zeros. The FSM filters out unwanted leading zeros based on whether the more significant digits are nonzero or not. The MUX following the FSM can be thought of as a filter for the anodes. As the select line changes value, the MUX will allow through one anode value at a time, whether it be '1' or '0'. The last component is the hex to sevenseg converter for the displays.

EXPERIMENTAL SETUP

Our simple calculator ran entirely on the Nexys A7 50T board, using switches, LED's, and buttons. 3 7-segment displays were utilized to show the result of the calculation. The row of LEDs were used to show the remainder value in binary when dividing. There was a calculate button when dividing, and the switches controlled the calculator. The first eight switches of the board controlled the 2 4-bit inputs in binary, the next 2 switches controlled which method of calculation would be done, and the last switch was an enable switch.

While writing the code, an experimental test bench was used to determine whether the 4 methods of calculation worked. Which allowed us to fix any bugs and errors we were getting, specifically with the division and subtraction components. The code itself was written using Vivado and VHDL, using modified components of previous labs we did.

RESULTS

After bringing our experimental setup and methodology to fruition, our calculator was installed on a board and ready for testing. Results primarily were as expected yielding a successful testing period. Testing showed that our simple calculator was capable of performing addition, subtraction, multiplication, and division between two unsigned 4-bit numbers. When performing division using two values that would result in some sort of remainder, the remainder was accurately displayed on the board without any issues.

Each individual function of our simple, unsigned calculator was based on processes and fundamentals gone over in either lab or class. The ideas and format of our full adder, subtractor, multiplier and divider, are modeled after similar designs found throughout different lab sections over the semester. The FSM implemented into our design that selects how many bits of our display are in use and the ability to select which function is used is what separates our project from the samples found in lab.

As expected, when performing subtraction that would traditionally result in a negative solution, our 7 segment display showed two hyphens in sequence. This was provided as an auxiliary input to the display to keep results in the scope of our project which solely worked with unsigned inputs and outputs. The sole scenario that would yield and unexplainable result is the specific case where a non zero value is divided by zero. This unique case would result in the value “63” displayed on the Nexys A7 board’s 7 segment display.

CONCLUSIONS

Our final project turned out close to what our original idea was, with some improvements and changes made. In the end, we created a calculator that could add, subtract, divide, and multiply two four bit unsigned numbers. Originally we had wanted to include signed numbers, but in coding the project, we found it would have been too difficult to sign multiplication and division. Creating a simple calculator, we had to learn how to code multiplication, and we had to first learn how to divide in binary, then learn how to code division. Since we never learned how to divide with binary, figuring out how to divide by binary and how to code it, was our biggest obstacle. If we were given more time to create this project, we could have signed the inputs and outputs. This would add a lot to our project, since we could use almost twice the inputs. Also with more time, we could create a better display board instead of what comes along with the Nexys A7. Except for what was stated in making improvements to our project, there were no errors still in our project affected the performance.

REFERENCES

[1] “Oakland University Moodle.” *Oakland University Moodle: Log in to the Site*, moodle.oakland.edu/pluginfile.php/4973230/mod_page/content/8/Basys-3-Master.xdc?time=1546210729156.

[2] LLAMOCCA , DANIEL. “VHDL Coding for FPGAs.” *VHDL Coding for FPGAs*,

www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html.

APPENDIX

Figure 1 Displays the Top Half of the Circuit Discussed in Part A of Methodology

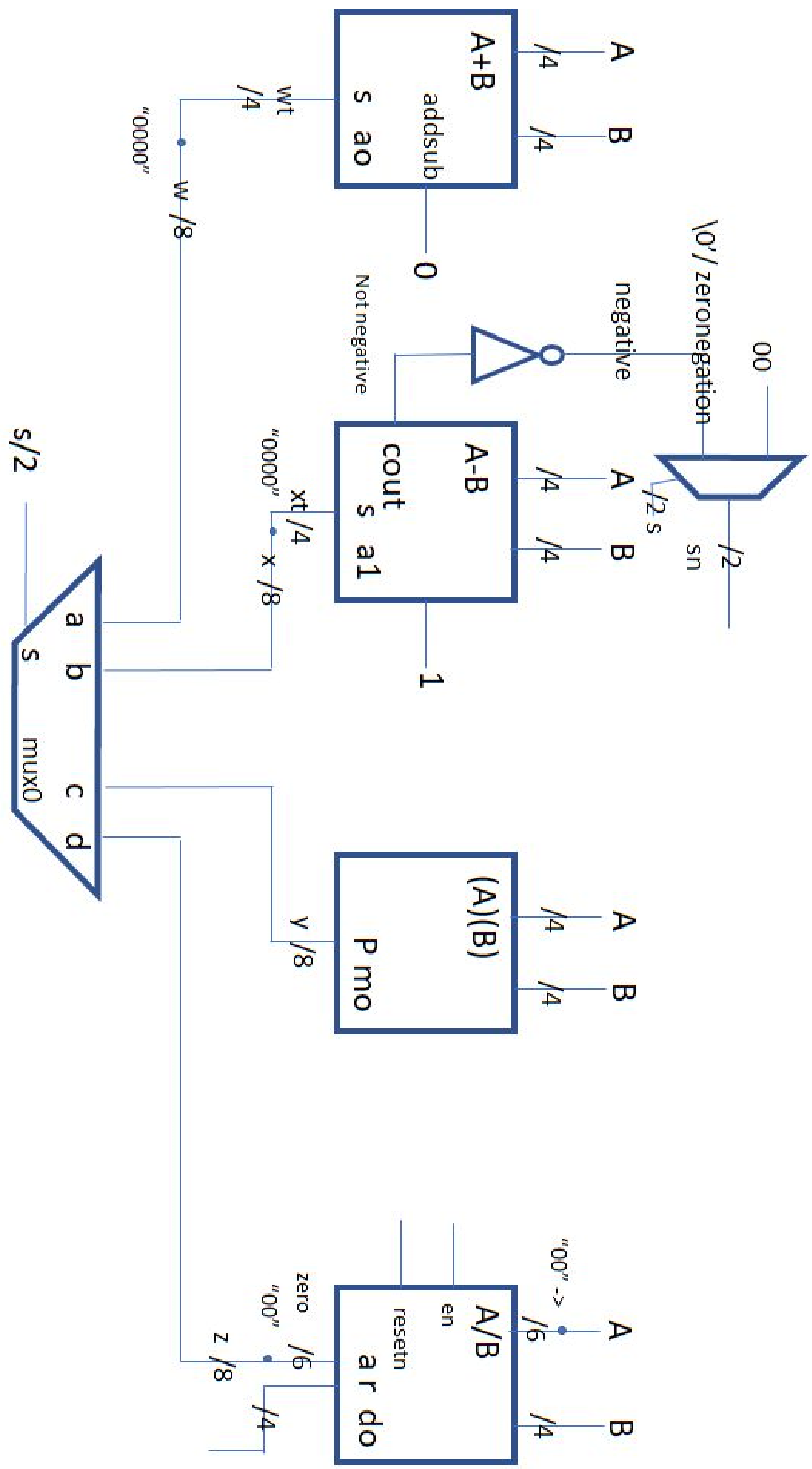

Figure 2 Displays the Bottom Half of the Circuit Discussed in Part B of Methodology