Simple Digital Calculator

Yuzan Xiong, Jing Wu, Zongyu Yao
Electrical and Computer Engineering Department

School of Engineering and Computer Science

Oakland University, Rochester, MI

e-mails: yxiong@oakland.edu, jingwu@oakland.edu, yao@oakland.edu

***Abstract* — The purpose of this project is to implement a traditional simple digital calculator that performs the four basic operations; Addition, Subtraction, Multiplication, and Division of 8 bits unsigned numbers. Nexys4 board was used to manage four computations, input functions and output functions. 16 switches on the board will be the input, and output will be displayed on the seven-segment on the board. LEDs on the board will be used to show the sign of results, remainder, and process done or not. The Digital Simple Calculator consists of three main portions in order to achieve the goal. First portion is the input block which will accept inputs from the switches on the board, then transfer to four registers to determine what two 8 bits expressible numeric inputs are and storage of the inputs. The second portion is the Finite State Machine which will determine the state of the inputs and the logic will be responsible for the actual calculation of the result. The third portion will be the output portion which will take the calculated result from the FPGA and display the results on the seven-segment.**

I. INTRODUCTION

Experimental Setup, Methodology, Results and Conclusions of the implementation of the Simple Digital Calculator will be discussed in this report.

The goal of this project is to construct a simple digital calculator with four basic operations and consists of input, computation, and output. Definitely, an organized map of inputs, outputs, logical portions need to be drawn in order to make the simple calculator works properly. In the map of logical computation portion, the calculator cannot function properly if there is a mistake no matter how small it is. So it does not simple as we used in normal lives.

The scope of the project was building a simple calculator, which can calculate signed numbers' simple computation. Seven-segment on the board was used to display the results of operations. The first five seven-segment will show the result, and the remainders will displayed on the last three segments. Initially, group members thought it would be interesting to design a simple calculator by ourselves because the calculator is familiar to us. The way to design the simple calculator requires us to cover several topics were discussed in ECE 378, such as decoders, registers, Finite State Machine (FSM), and Arithmetic Logic Unit (ALU). Those knowledge gained from the labs of ECE 378. For example, Arithmetic Logic Unit were learned from lab 3 to know the basic functions of ALU and how to construct it. In lab 2, it informed the implementation of the multiplication, and lab 4 taught us to construct the register for storing values. The design of division feature of the Arithmetic Logic Unit was mentioned in lab 5, and a way of address registers and the construction of Finite

State Machines in Control Circuit were provided by lab 6.

As it designed so far, the application of the simple digital calculator is only as a traditional calculator, other applications need to explore by our own.

II. Methodology

Arithmetic Logic Unit was used in this project. Arithmetic Logic Unite is a modular design and contains four modules: Control Module, Add/Subtract Module, Multiply Module and Divide Module. This simple digital calculator can process four basic operations between signed numbers or unsigned numbers.

A. Control Module

The control module needs to able to control and select on other modules by using a two-to-four decoder.

B. Add/Subtract Module

Add/Subtract module requires to process 8-bit signed (unsigned) numbers' add/subtract. For a signed number, it needs to extend the numbers to 17 bits by using the sign bit. And subtract is achieved by 2's complement.

C. Multiply Module

If the inputs are two unsigned numbers or two signed number with the same sign, the result does not change. If two signed numbers multiply and opposite in sign, the module needs to invert the result and then plus one.

D. Divide Module

The dividend will be expanded to 32 bits. For example, if the dividend is 01010100, then extend it to 0000000001010100 and assigned it to DATA. Next, compare the DATA's high order's eight digits with the divisor. If dividend is larger, set the quotient as 1 and assign it to the lowest digit of data. Then, let the dividend minus divisor and DATA shifts one digit to the left. After eight times' repeated, high order's eight digits of DATA is the remainder and the rest is the quotient.

III. Experimental Setup

To verify the functioning of the project, different hardware and software were used in the project. For the software, Xilinx ISE 14.7, which utilized throughout the labs to design, synthesize, implement and program the Nexys4 Board. This software was used to code and simulate the project. Individual part/module was simulated on the ISE to make sure it works. After the separately simulations of the files, then those files were combined into a top file. Finally, simulate the top file and to verify the function of it.

For the hardware, Nexys4 (DDR) Board was used to define the inputs and process the operations of the inputs. One of the hardware was used in this project was a PS/2 Keyboard, which was abandoned because of some unsolved issues. The keyboard as the input of the calculator and can convert the decimal to 2C' number. Some protocols which shown in the codes cannot be understood by our own, so the keyboard was not implemented in the project.

Another one is the display choice, group chose to display the results on the seven-segment and use Leds to show the sign of results and remainder. The first five seven-segment will display the results and the remainders will showed on the last three seven-segment. Led0 lighting means the calculate is done, Led1 lighting means the sign of remainder is negative, and Led 2 lighting means the sign of result is negative.

With these components hooked up to the board, and using the switches to identify the register input, the system was able to function as predicted which is the calculation of mathematical outputs based on switch inputs.

IV. Results

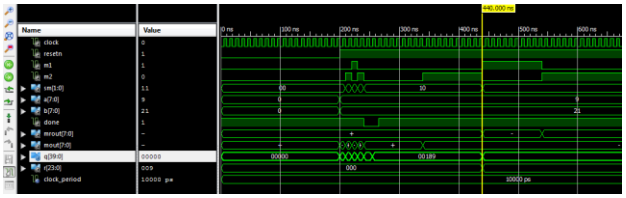


Figure 6: The result of Multiplication

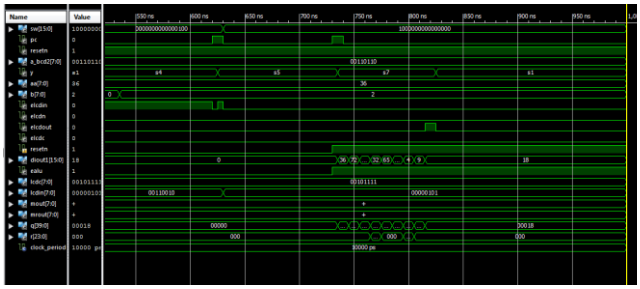


Figure 7: The result of Division

Figure 6 and Figure 7 shows the functional simulation by using Xilinx ISE 14.7 of the Multiplication and Division of the inputs.

In Figure 6, inputs were 9 and 21, and the Nexys4 board made a multiply between them, get the result 189, which was correct. In Figure 7, the inputs are 36 and 2. 36/2 = 18, and the remainder is 0.

The functional simulations show the coding works are correct and it works properly.

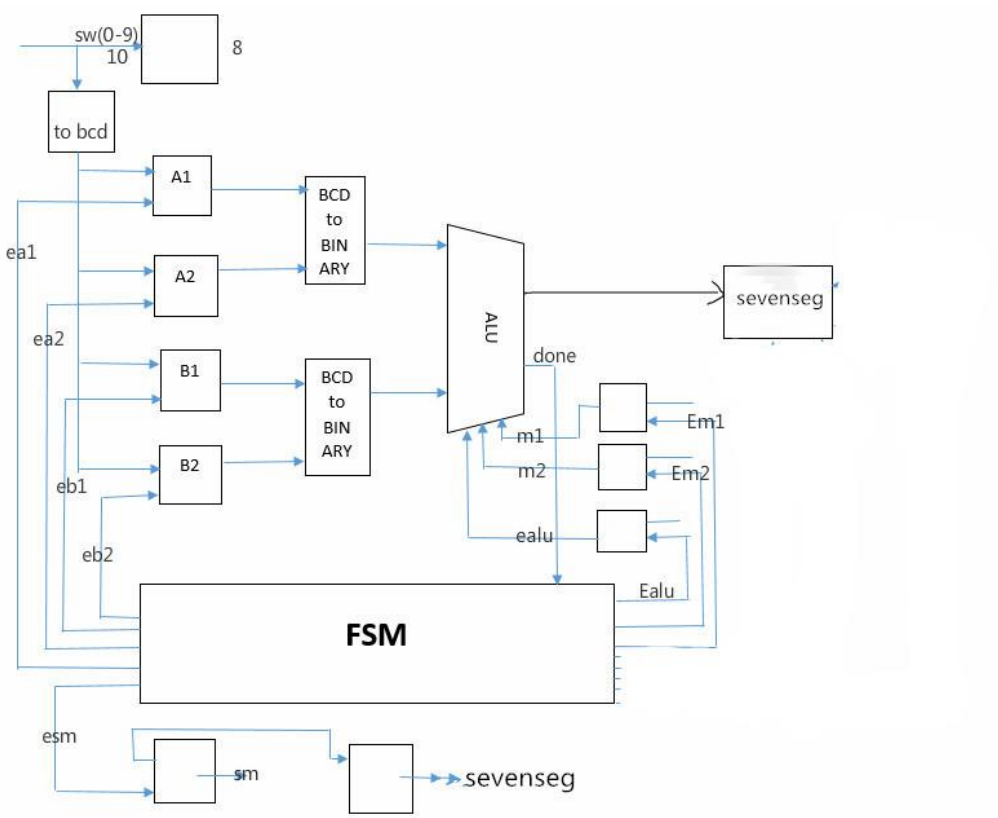
V. Conclusions

In conclusion, this project successfully demonstrated a working 4 function simple 8-bit calculator and the complexity of the very simple operations involved in adding, subtracting, multiplying, and dividing two 8-bit unsigned numbers. The process requires very careful recording of inputs and outputs, translation of signals, and output of the results.

The main functions of the calculator function as designed however many improvements could be made. The calculator could make the task of inputting data simpler so users would not be tasked with the work required to store data in specific registers. Also the calculator functions could be expanded to include values that require greater than 8 bits to represent. Finally, the data output could be improved to allow users to see the values they input and the output could be translated into decimal values that are more understandable.

VI. BLOCK DIAGRAMS

OVERALL DESIGN


sw(0-9)
10
8
to bcd
A1
A2
B1
B2
BCD
to
BIN
ARY
BCD
to
BIN
ARY
ALU
sevenseg
done
Em1
Em2
m1
m2
ealu
Ealu
ea1
ea2
eb1
eb2
FSM
esm
sm
sevenseg

FINITE STATE MACHINE