Ramzey Ghanaim

Lab 4: Building an ALU

Introduction

In this lab, my job was to build an ALU with logic in the program Multi Media Logic (MML). This lab builds upon the previous lab by having user input from a keypad be saved in registers before being sent to the ALU. But in this lab I have 4 registers, not two. These basic ALU functions contain a passer which allows the users input to pass to the final output, a NOTer which inverts the user's input, an ANDer which and's two of the users inputs that are stored in latches A and B, and an adder which adds the two numbers stored in latches A and B. The purpose of this lab is to get a full understanding of the exact processes that take place in the LC-3's ALU.

The Lab

In this lab, I first had to create the front page dedicated for user input. I started with a Input pad which has user's input going to registers just like in the previous lab. However there were a few minor differences in this lab. Since I used 4 registers this time I had to create a second Address Select to choose which of the four registers I am writing data to. The second difference I had in this lab was the fact that the keyboard input did not go directly to the registers. This input went into a mux which the user can control to decide whether they want the keypad input or the ALU output going to the registers. The user can decide with the Input select switch. And just like the previous lab, I had a clock, reset, and write enable switch. If the user wants to write the values of a register to the latches they can once the Latch's write enable is turned on, and the write/read enable switch is turned to read. This is because I want the Latches to read the values in the registers to put them into the appropriate latch. The write/ read enable switch is writing only to the registers. The latches can always access data from the registers assuming the switch is set to read. From here the user can choose to write data to latch A, latch B, or both by turning the latch write enables on and off. From here, the values in the Latches are the values that are sent to the ALU to perform a calculation depending on the opcode the user selects.

The ALU

When I first designed the ALU I created four functions for it. These four operations are controlled by the user with the Opcode Select. If the Opcode select is at 00, then then my ALU Logic is set to NOT the data in Latch A. If 01, the bits in latch A and B are ANDed together to perform the AND function. When the opcode is set to 10, the ALU will perform 2's complement addition of Latch A and Latch B. And lastly at 11, the ALU is set to let the data in Latch A pass through, to become the final output. I used a 4 to 1 mux to decide which operation to take on based off of the user's input. Basically my ALU design works by computing all of the operations and then a mux decides which one of the inputs to output the final result, based on the user's Opcode Select switch. From here the final result is sent to the 7 segment LED on the first page of the schematic. The final output is also sent to the mux

that I discussed in the beginning so the user can chose to write the data in the ALU or data from the input keypad to the registers.

When it came to the adder, I had an over flow detection which detects when there is an overflow. When adding two numbers. The overflow occurs when the sign bit of a number is changed when adding. A carryout happens when and answer contains more bits than the number of bits you have to store it.

Conclusion

In this lab I learned how the basic operations that take place in the LC3 ALU work. I was able to re-create an example of how the ALU processes data along with how it receives and outputs data. Because of this lab I was able to gain a better understanding of how data flows through the ALU, and the LC-3.