Rashad Kayed

Section 3: TuW 9-11

Due 2/6/2013

Lap Partner: Kyle Novelli

Lab Report 2: Memory and ALU

Purpose:

The purpose of this lab is to build a memory and arithmetic logic unit, that

is 4 bits, capable of performing some functions.

Procedure:

To begin with, you start by constructing the memory portion of the lab.

We started with our first page where we made a keypad input for the input

display and we included the output display there as well, also included the

general clock, reset, and write enable functions. Then we began building the four

banks. From there we connected the banks to the four multiplexers. Also, made

two select lines, which are connected to the address decoder and the

multiplexers.

For the ALU, we started by building out source registers, and destination

registers. From there we connected them to clocks and resets. We also

implemented the first page which consists of the input and output displays. We

build the multiplexers and opcode decoder and got them connected. And then

worked on making the NOT, AND, and adder options of the ALU.

Algorithm & Other Data

**Address Decoder Truth Table**

|  |  |
| --- | --- |
| Input | Output |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AS1 | AS0 | R0 | R1 | R2 | R3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

**One Bit Adder Truth Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| In1 | In2 | CIN | Sum | COUT |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

DeMorgans: (AB)' = A' + B' (A+B)' = A'B'

AB means A AND B

A+B means

A OR B

What Went Wrong or What Where The Challenges:

The most complicated part of the lab was I think keeping everything

organized and maintained in a manner that would help us see everything. On

several occasions my partner and I hooked up senders and receivers to the wrong

pins.

Other Information:

Sequential logic is useful because it creates a simpler user interface, and

more clear and clean schematic to work with, we can set and reset all values.

The inputs and outputs are latched because that allows us to store and

remember the values. A SUB instruction is not included in the LC3 instruction set

because it would cause for a more complicated system to be implemented and

would need another bit. To implement and OR we would have inputs coming in

from the source registers and into the OR, then into the multiplexers. The

purpose of the N, Z, and P, is to tell us if the value we receive in the output

display is negative positive or zero. A carry out is the extra digit that exceeds the

max number of digits for a specific problem, for example the 4th bit in a 3 bit

addition problem. The clock is used to push in a value or store it into a register or

slot. The Reset is used to reset all slots so that one can input new values. A

register file is used to transfer data between memory and proposed functions. It

is basically the combination of all our banks. You tell the register banks to

perform a certain operation by passing it in through the switches on the input

page, where it is then transferred into the source register and from there into the

ALU, for the bank option that is passed in through the input manually by the user

and clocked in. Having the output of the ALU go into the register file is good

because it allows the register file to store the value, and use it to do other

functions if needed later.

Conclusion:

This lab was definitely a big leap from what we have been doing or

responsible for doing in class. But we got it done. Basically in this lab we had to

start by building memory, and implementing that aspect, making sure that we

can input and store values. Then, we had to make an arithmetic logic unit, which

in our case, adds, ands, or ors, two values.

In the final portion of the lab we had to connect the two and make sure

that we can use arithmetic on values in our program and save those values

implementing our memory properties. Everything went as expected except that

we had trouble figuring out how to actually use the new and improved system

which combined the ALU and the memory together. Once figured out it took

practice to try and test the different propositions we were responsible for.