Ramzey Ghanaim

Lab 3: D-Latches and Registers

Introduction

In this lab my goal was to learn about D latches and flip flops and learn how to construct memory with them. The reason we use flip flops to construct memory is because flip flops allow an output to be “trapped” or “saved” in the logic design. The second objective of this lab was to explore the basics of memory and how the logic of memory elements work.

Part A

In Part A I was required to take a D-latch and find a way to add a Reset button to the logic. To do this I first had to understand how the latch worked. It turns out that when you flip the switch for the data input D, nothing happens unless the clock is activated. If the clock is not activated, no changes can be made in the output. Otherwise, when the clock is activated, if the D is 1 then output Q is 1 (LED is on). If D is 0, Q will be 0 (LED is off). When adding the reset switch, when this button is clicked it’s job to to turn off the Q LED if it is on, or keep it off it the LED is not on. To help in my design, I created a truth table which looks like this:

Data (D)	Clock (C)	Reset (R)	Output (Q)	Not Output (Q’)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1
1	0	0	0	0

As you can see, the output is only a 1 when Data and the clock both equal 1. I added the reset but by OR-ing it with both the clock and the Data. Because these buttons are momentary switches, when R is pressed, the Reset’s 1 is the logic that is followed and clears the data stored in the D-latch. I used OR gates to decide whether to follow the clock or the rest because by default both values are zero. The OR gate will select the switch that is 1 at any particular time. Therefore, if clock is 1, the reset is 0. 1 OR 0 is 0 so the Clock’s output is what is sent to the D-latch to write data D. This same logic applies when the reset is 1 and the clock is 0.

Part B

In Part B I constructed a basic register file to design basic memory. This memory had 2 unique locations to store a value. This is known as address space. Each address space contained the ability to store 4 bits. The word size for this memory I constructed is 4 bits. I created the basic storage element by starting out with 16 flip flops to store each bit value. 4 of the flip flops were used for the first register and the other four flip flops were used to store data for the second register. Just like in part A I have the Data input (user input), Clock input, and a reset button to clear the flip flop, and lastly an output came out of each flip flop. Now since I have 2 registers, I needed to determine a way to decide which register to read and write data to. I did this by combining the clock, write enable, and address selector into a AND gate. If the clock and write enable are on, and the address select is 0. I had an inverter in front of the address select to make the AND output 1 when the address select is zero. The output of this AND gate is the clock input for the first register's flip flops (Register 0). I then created a second AND gate. If the clock, write enable, and address select are all 1 then the data will write to the second register (Register 1). The output for this AND gate is the clock input for the all the flip flops in the second register. Now that I established a clock for the registers we can now determine which register to read/write to.

Whichever register has the clock enabled, will allow the data to pass through and be saved. Now I needed to figure out how to display data from one register or the other. I did this by creating 4 two to one multiplexers. Each multiplexer determines whether or not bit n (n is any integer, 0-3) is sent from register 0 or 1 to the final data register output. The mux decides which register to read by checking the address select. If the address select is 0, then register 0's bit is outputted. Likewise, if the address select is 1, then Register 1's bit is outputted. The four bit outputs go into the Register Output. The 16 different combination of 1's and 0's from the four multiplexer outputs determines a character to display in the 8 segment LED. This Display is the final output of what is stored in a register.

Conclusion

In this lab I learned the basic of memory by designing d-latches, and using flip flops and multiplexers to create a simple memory architecture. I learned about how logic works in memory and storage elements and how they each function individually. I also learned the path of which data flows through in a basic memory structure.