EE305 Experiment 4&5. Calculator Design (I)

Faltamo Bereket Yisehak

Post-lab Report

1) Purpose (5%)

1) Understand combinational logic circuits and sequential logic circuits.
2) Design and implement a calculator which performs OR and XOR operations using an ALU, registers, capacitors, and logic gates

2) Methods (5%)

- The grounding resistors and coupling capacitors are designed (all with R = 330Ω and C = 0.1μ)
- The inverting Schmitt triggers and an inverter gate to are then connected to apply hysteresis to the input signals.
- An input system is designed that takes two inputs and feeds the shift register serially is designed.
- A 1-to-2 demultiplexer is designed then to input the two operands to the two shift registers. A D-flipflop is used to get the logic for the control signal of the demux.
- The clear signal is used to clear the shift registers when the operation is finished.
- The outputs of the shift register go first to light LEDs, and two to the ALUs to be computed.
- The ALU computes using the selection signal specified by a no-d-input D-flipflop using the operands and Clear input as set/reset signal.
- The outputs of the shift register are then used by the BCD-to-Seven segment circuit to light the seven segments. The BCD-to Seven segment circuit is designed using *Logic converter* component of *Multisim*.

3) Results (15%)

A.

i. *Explanation of your final circuit with a picture of it – picture can be a capture of simulation result (5%)*

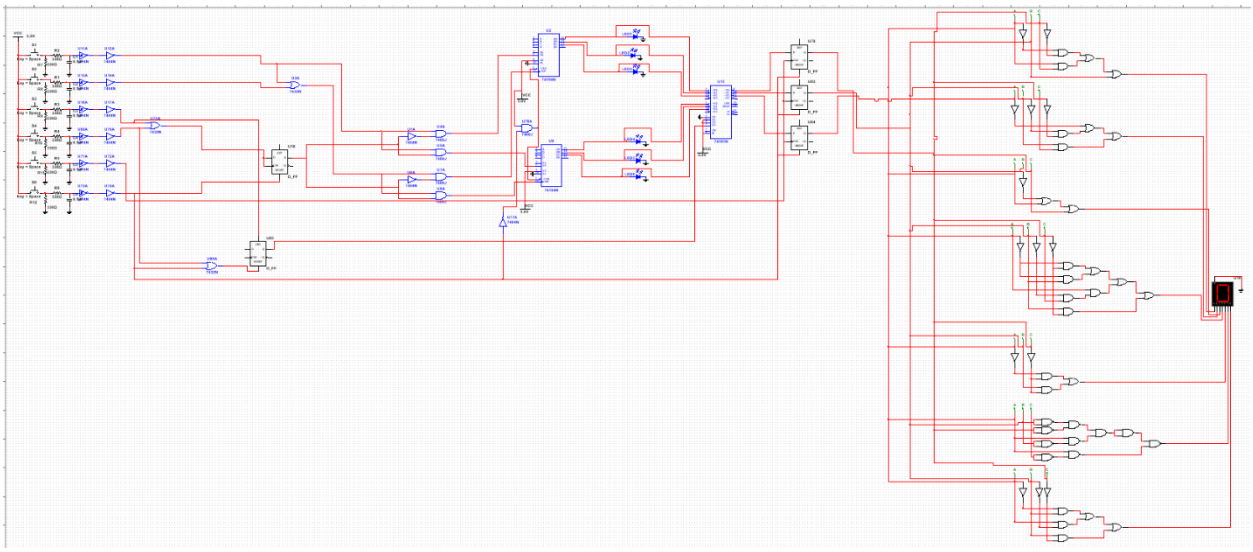

Figure 1 Calculator Circuit

My calculator circuit is as shown in the figure above. For the ease of discussion, I will divide the circuit into 3 units. The components of the elements are as listed below;

Unit 1	Unit 2	Unit 3	Unit 4
Schmitt trigger	Shift registers	ALU	BCD-to-segment sub-circuit
Input system	6 LED	2 D flip-flops	Seven segment display
Selector sub-circuits			
2-to1 demux			

Figure 2

The units are shown in the picture below in rectangular boxes.

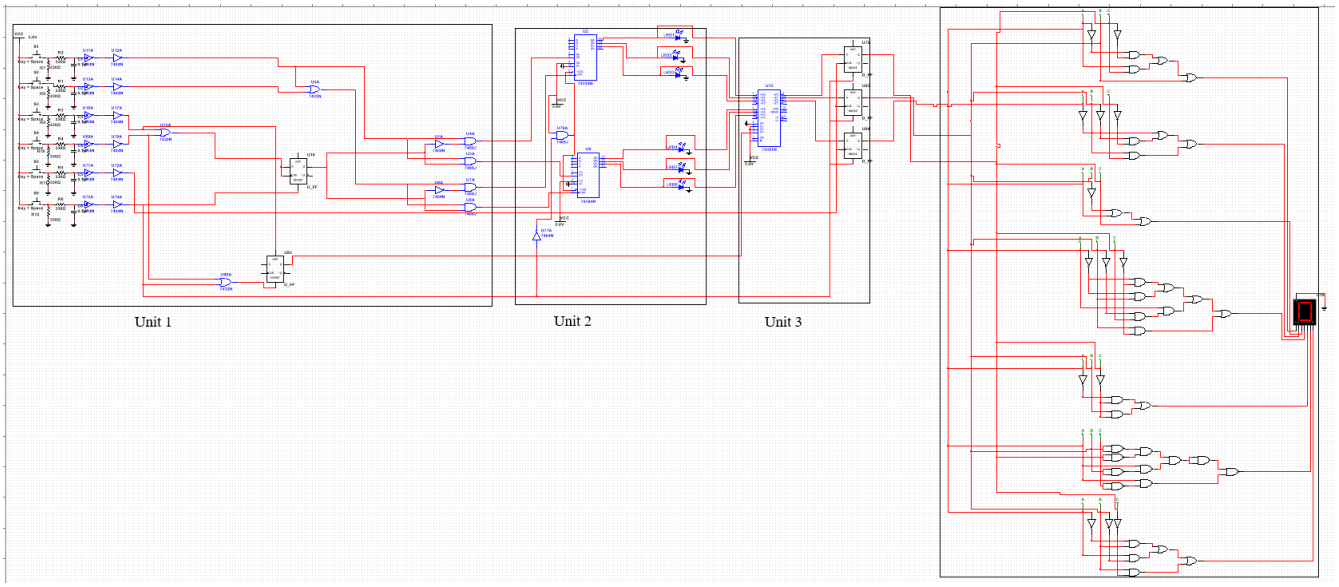

Figure 3

Unit 1's first function is to make sure the data is in digital form using the Schmitt triggers and also prevent switch bouncing using the bouncing capacitors and grounding resistors. Next the input system and demultiplexer process that inputs to the shift register. The final components specify the selector and clear signals to the shift trigger and storage D-flipflops.

Unit 2 directs in parallel the two inputs coming in serious to the ALU which receives parallel inputs, in addition it shows the operands value using the LEDs.
Unit 3 computes the operation and stores it. Finally, unit 5 shows the stored result using a seven-segment display.

ii. Circuit diagram of your final circuit – It will be your simulation file (5%)

⇨ Please, check the attached .ms file.

B. Simulation results on given input tables (5%) (+: OR ⊕: XOR)
i.101 + 110 (1%)

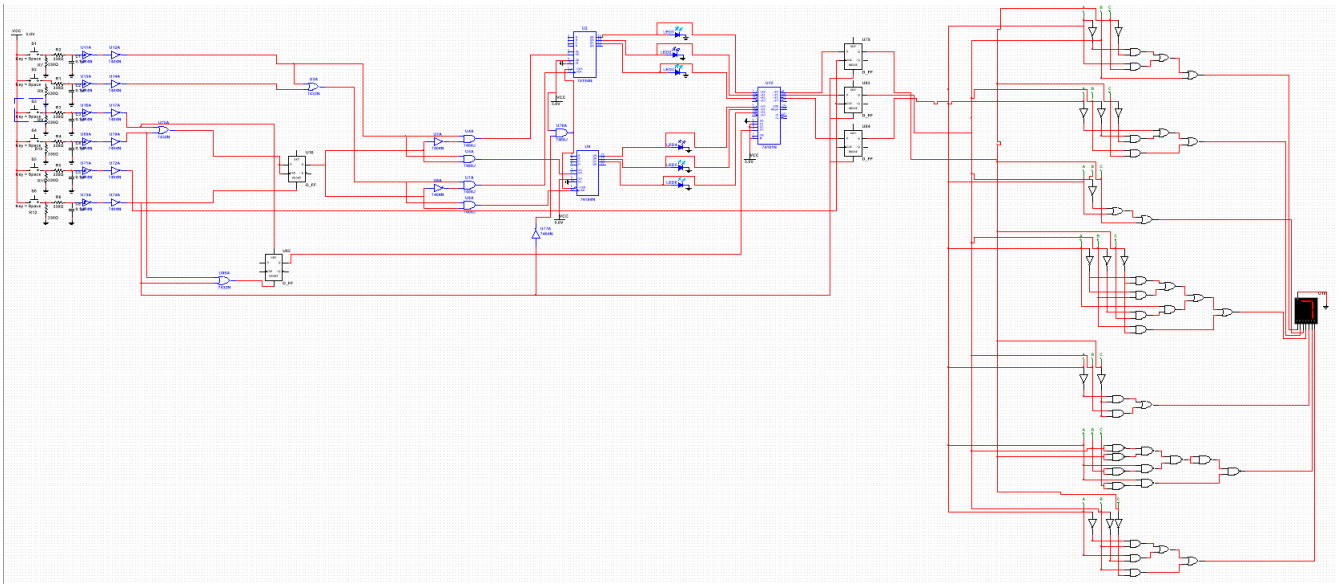

Figure 4

ii. 111 ⊕ 110 (1%)

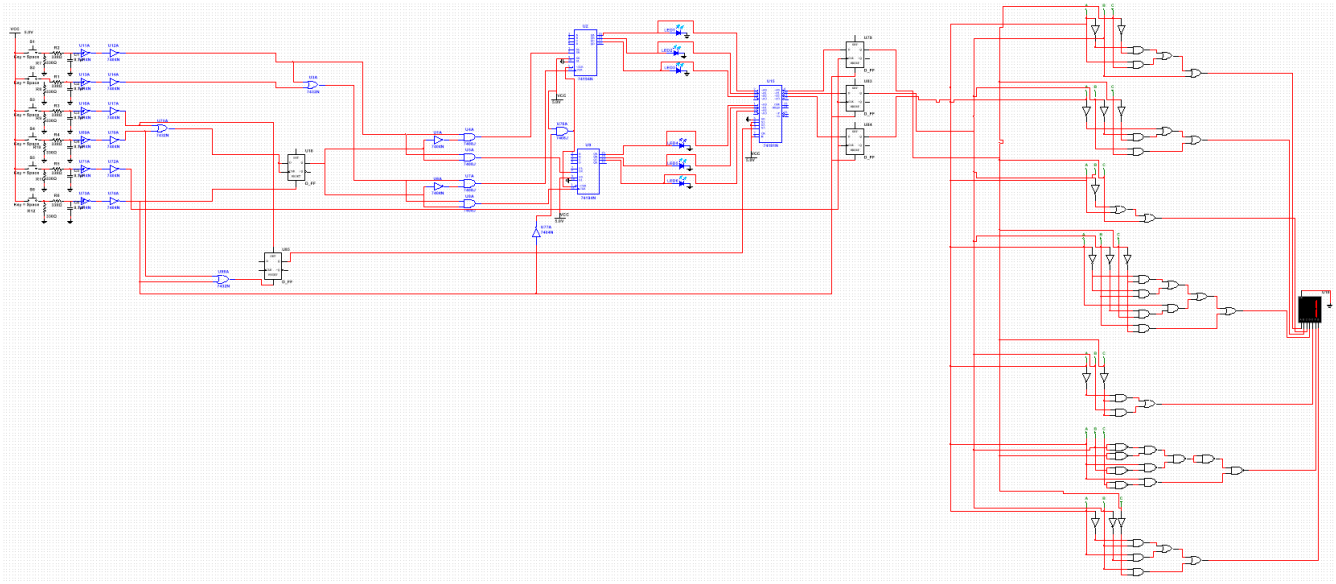

Figure 5

iii. 10 + 101 (1%)

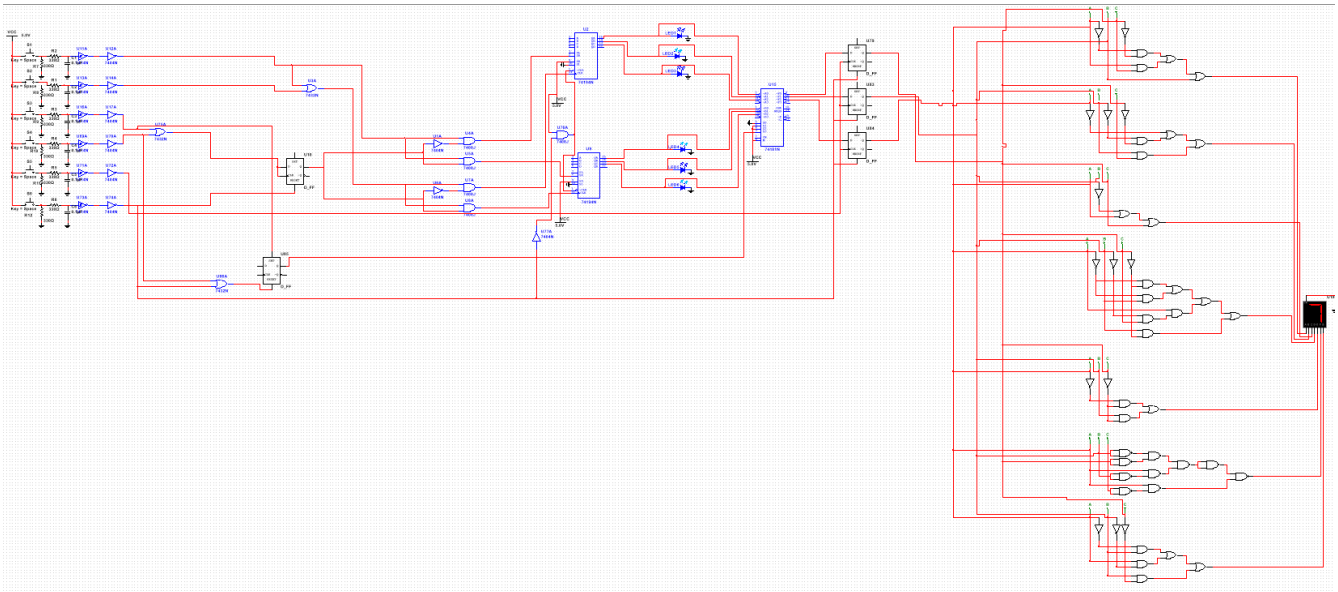

Figure 6

iv. 110 ⊕ 1 (1%)

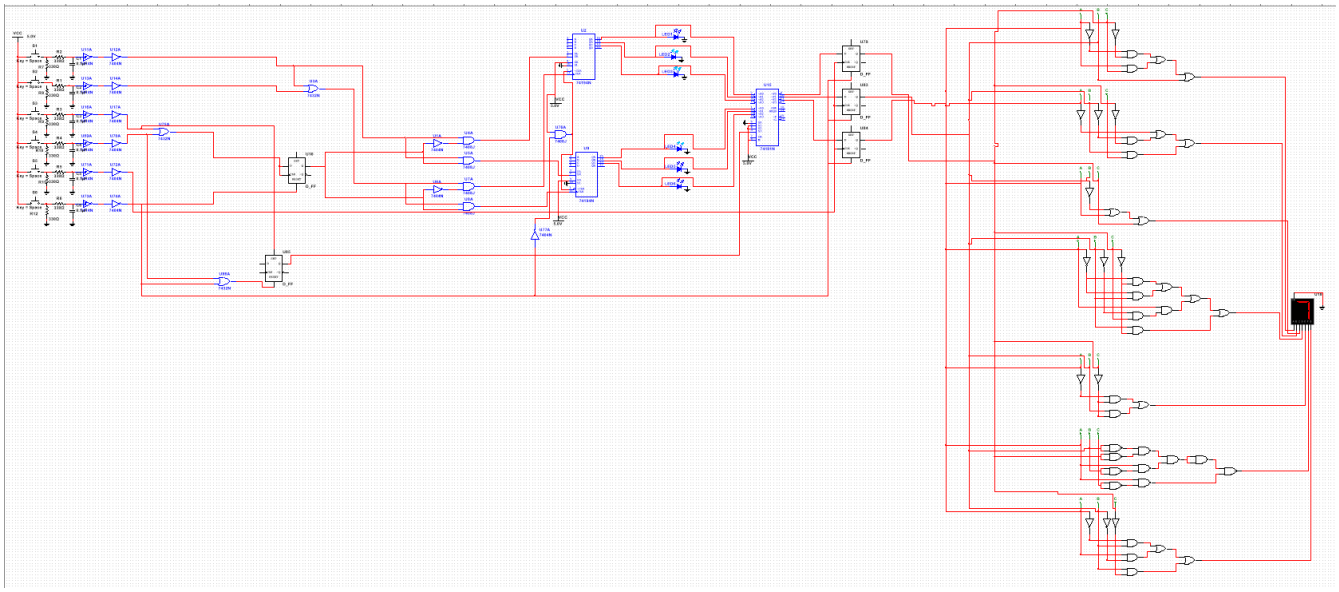

Figure 7

v. 101 Clear 11 ⊕ 110 (1%)

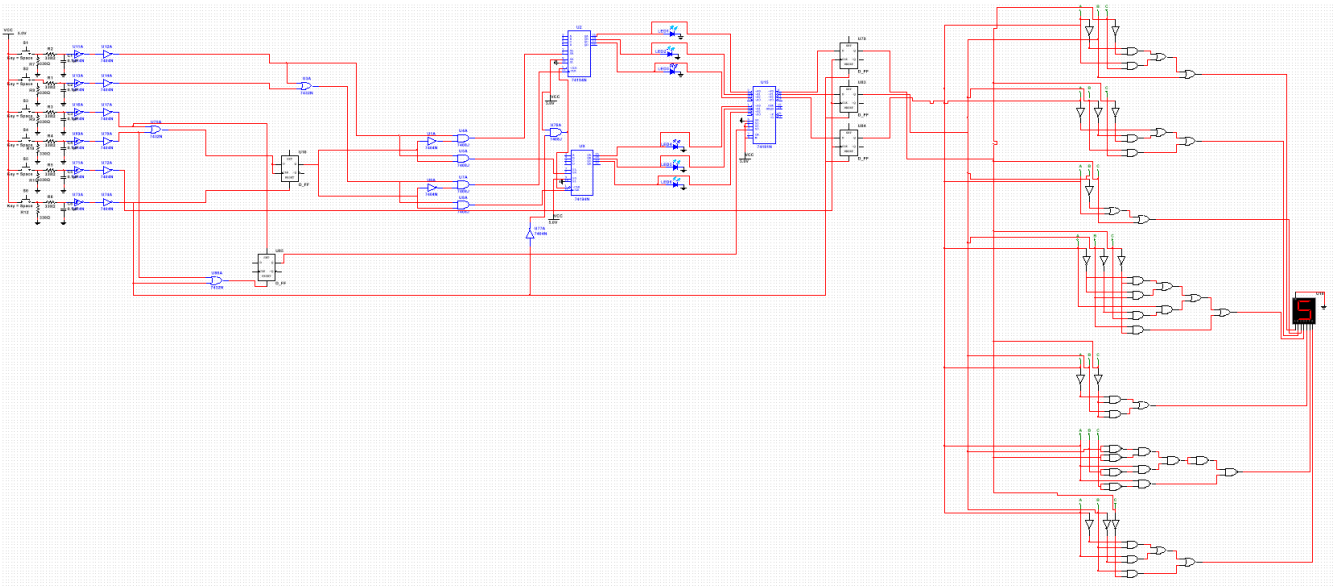

Figure 8

4) Discussion (25%)

- *Comparison between the circuit diagrams on prelab and your final circuit (10%)*

I haven't finalized the circuit diagram at the time I submitted the pre-lab report; however, I have ended up modifying a lot from the initial circuit design I made before implementation. I will compare the initial design and the final circuit.

So, the main difference between the two is the number of the shift registers. In the pre-lab design, I have used a single shift register. The purpose of the SIPO shift register is to serially take inputs from the Schmitt triggers and dump them into the two D-flipflops parallelly. The two D-flipflops are then supposed to spore the two operands. However, in the final design, I have used two shift registers. They store the two operands and also help in lighting the LEDs. The LEDs in the prelab design were connected to the D-flipflops, not the shift registers.

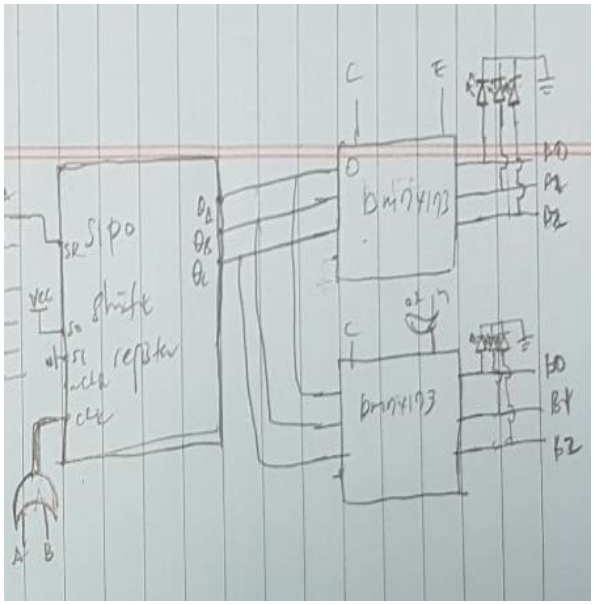

Figure 9

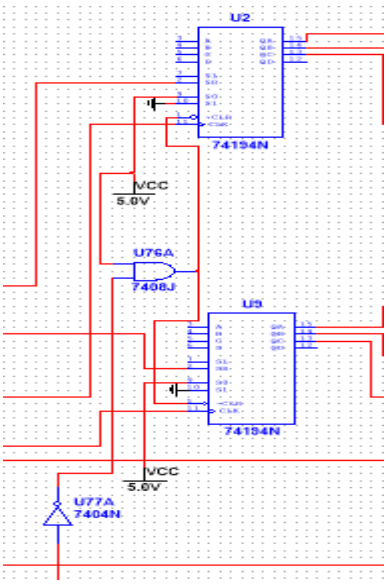

Figure 10

In addition, I haven't used a multiplexer in the prelab design, the inputs are just input into the two storage d- flipflops using clock signal.

Another is the way to store the operator type to the final ALU computation step. In the pre-lab design, I used a D-flipflop as below. The above inputs are the outputs of Schmitt triggers from OR, XOR and CLEAR inputs respectively. The outputs are used as selector(S3) for the ALU. However, it didn't work as expected consistently. I suspect the delay in the number of gates contributed to this problem, by changing the propagation delay in the circuit. In addition, I have tried to fix this problem by using the *Transport Delay* component, but to no avail.

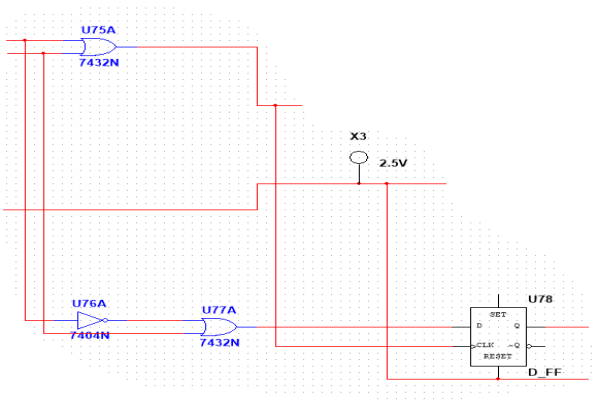

Figure 12

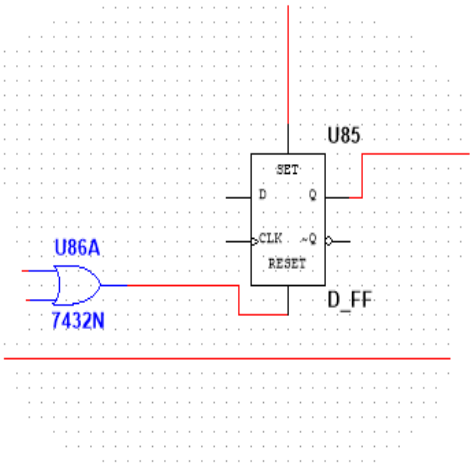

Figure 11

I finally was able to solve the problem by using a D-flipflop with no inputs, but set and reset as the two operand inputs as shown in the figure above.

- *Pros and Cons of your final circuit (10%)*

Pros

- As explained above, the circuit was classifying into units based on functions(modularized). This has helped a lot when debugging as it means we debug smaller circuits which are easier to troubleshoot.
- The clocks in the circuits were made completely from the switch inputs and logic gates.
- Any noises introduced to the inputs are filtered by the low-pass filters and Schmitt triggers. Thus, one less thing to worry about.

- We can add other binary numbers too(2 digit or 1 digit binary numbers) as the MSBs are by default zero, so they stay as is unless otherwise updates. This makes it possible to add a wider scope of number to be added.
- Using a specific logic in the mode selection for ALU, we can calculate a larger number of calculations.
- The inputs are shown in the LED real time. This helps in correcting if any digits are incorrectly input immediately.

Cons

- The circuit wires aren't color-coded. This makes it difficult to debug. It may also be difficult for someone (except the designer) to understand the logic.
- The workspace of the simulation is so long. It maybe difficult to implement the whole circuit in a real physical bread-board.
- The d-flip flops used to stored the output before displaying are individual flip-flops(D_FFs) and not DM74173. DM74173 didn't work for me even if I tried it again and again, so I resorted to three individual D_FFs.

- *Major challenge during the experiment (5%)*
  Honestly speaking, I haven't faced much challenge. The TA session lectures slides were pretty important and I followed the instructions there.
  One challenge I have faced at the beginning of the experiment the *Multisim software* was giving a lot of errors. There weren't many resources online to help fix those immediately. I have to copy-past part of the circuit and try it on other workspace to figure out what the problem is.

- Qualitative discussion

Consider what should be done for changing the choices of operators; For instance, if we want to change the current calculator to that computes addition & subtraction, what should we do?

In the *Pros* section above, we have said this calculator can be modified (or changed) to accommodate other types of operations. To illustrate this more, let's take the case of implementing an addition and subtraction calculation using our calculator.
If we want to calculate addition and subtraction using this calculator, we first need to make a logic that specifies the mode M of the ALU. For arithmetic operations, it is M=L. so M is wire to the ground. Then the two operations have common $s_3 = L$. So, we wire it to ground too. The other mode selects inputs s_1, s_2, $s_3$are specified by using a logic function of the input operands.
We finally then add 1 to the subtraction result as its value isn't exactly *A minus B*, but *A minus B minus 1*.
s_1 and s_2 can be specified as $add'.sub$, and $s_3 = add.sub'$; add and sub being addition and subtraction operand inputs respectively.

5) References

1. http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/flipflop.html
2. Frank Vahid - Digital Design-Wiley (2011)
3. https://www.youtube.com/watch?v=7jb3YHe8haw
4. https://www.youtube.com/watch?v=7jb3YHe8haw