Experiment #2:

Data Storage

Lab Report

Name: Chuangy Zhang

NetID: Czhan30

Name: Homero Vazquez

NetID: hvazqu6

1.Introduction:

The circuit that we built for this lab was a 2-bit, 4-word Shift Register Storage Unit. This means that the memory will have space for 4 addresses in which we store 2 bits per address. This Storage Unit supports 4 instructions: FETCH, STORE, LOAD, and NOP. There is an internal counter inside that keeps track of the addresses. Once the address is matched by the clounter we can either read or write from and to the registers. SBR is a register buffer that can hold memory contents or the data that we are trying to store.

2. Operation of the memory circuit

a. Describe how the addressing is implemented. When does the circuit commit a read or write from/to the input switches and output register respectively?

The addressing is implemented by used FPGA denounced switches for SAR[1] and SAR[0]. SAR which has the value of the address that we will read or write from/to register. The control unit also has the Fetch, Load, and Store switches for users to execute the instruction that they need. When the Store is 1, then SAR[1] and SAR[0] will compare the internal counters. When the values are matched, it will commit a write. When the fetch is 1, then SAR[1] and SAR[0] will compare the internal counters. When the values are matched, it will commit a read.

b. Describe how a write operation is performed on the memory. Describe what switches you flip and in what order. Describe intuitively how the data flows through the circuit at each clock cycle.

In order to perform a write operation on the memory, users need to flip the switch SAR[0] and SAR[1] to match their desired address. The internal counter is keeping track of the address, if it matches the address that the user selected, the write operation will be performed. The internal counter C1 and C0 will constantly compare

SAR[1] and SAR[0] respectively. The comparator is checking if the values are matched at each rising edge of the clock. if it matches the values and the Store signal is 1, the internal signal connects with the 2 to 1 MUX select signal, and it allows the MUX to select the value of SBR and write it to the memory.

c. Describe how a read operation is performed on the memory. Describe what switches you flip and in what order. Describe intuitively how the data flows through the circuit at each clock cycle.

In order to perform a read operation on the memory, users need to flip the switch SAR[0] and SAR[1] to match their desired address. The internal counter is keeping track of the address. The comparator is checking if the values are matched at each rising edge of the clock. if it matches the address that the user selected, the read operation will be performed. The internal counter C1 and C0 will constantly compare SAR[1] and SAR[0] respectively, if it matches the values and the Read signal is high, then 3 to 1 MUX select signal will select the data from the shift register , then the data will store on the SBR.the internal signal connects with the 2 to 1 MUX select signal, and it allows the MUX to select the value of SBR and write it to the memory.

3. Written description and block diagram of memory circuit implementation a. High-level description

The control unit will generate the signals necessary for the operation of memory. The input of the control signal are SAR[1-0], Data[1-0], Load, Fetch, and Store. If the load signal is High, it will send a 3 to 1 MUX select signal to select Data[1-0] and store in SBR. If the Fetch signal is high, it will send a 3 to 1 MUX select signal to select LBR output and store in the SBR. If the store sigal is high, it will send a 2 to 1 select signal to select SBR and write to the memory once the address matched. The most important is the address comparator if the diresed address and counter is matched, then it will authorize the execution.

a. Describe in words what components are necessary to perform the operations described in the written description of the memory circuit operation.

Shift Register: enable users to store or fetch the data

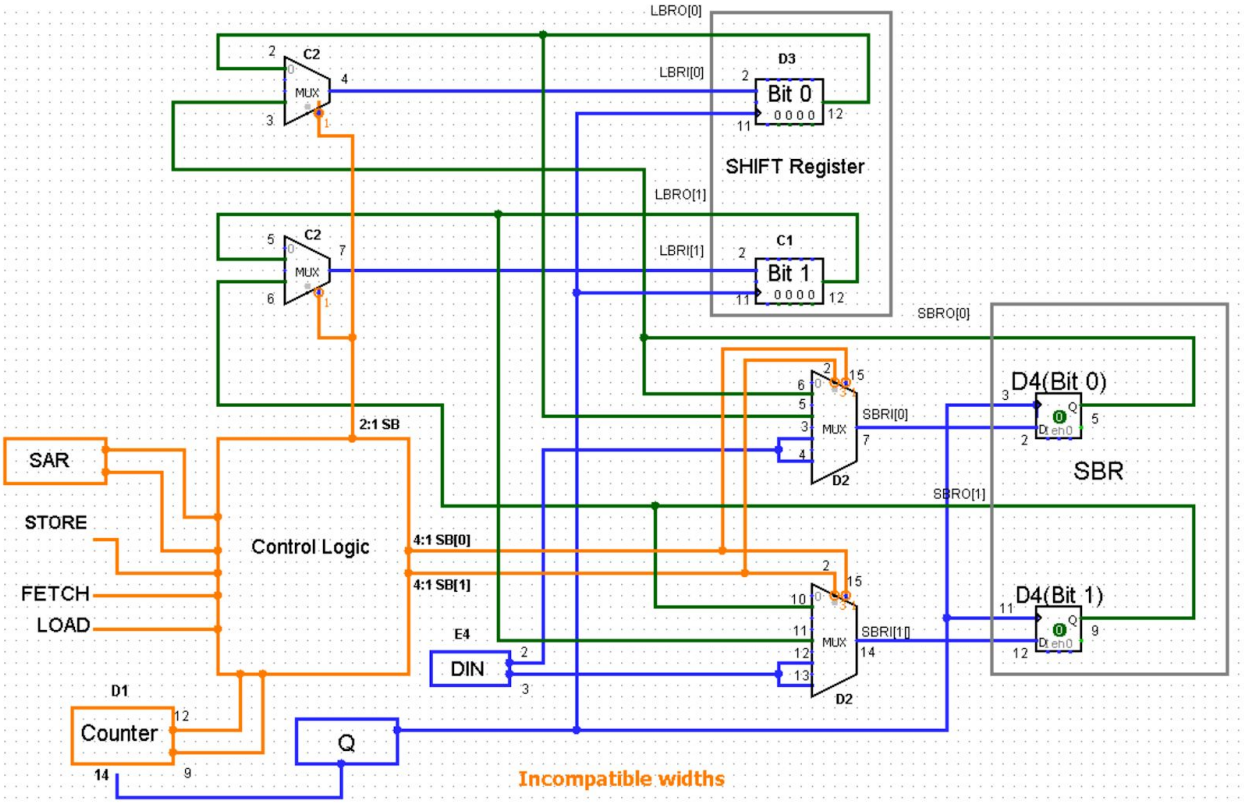
SBR: it acts like buffer register and it can hold Data[1-0] and the value that we fetched

2 to 1 MUX: enable the memory content to circular till we match the address and counter

4 to 1 MUX: enable the SBR to has the three values, SBR[1-0], LBR[1-0], and Data-in[1-0],

Counter: enable us to compare the user diresed address, store and fetch the correct to/from address

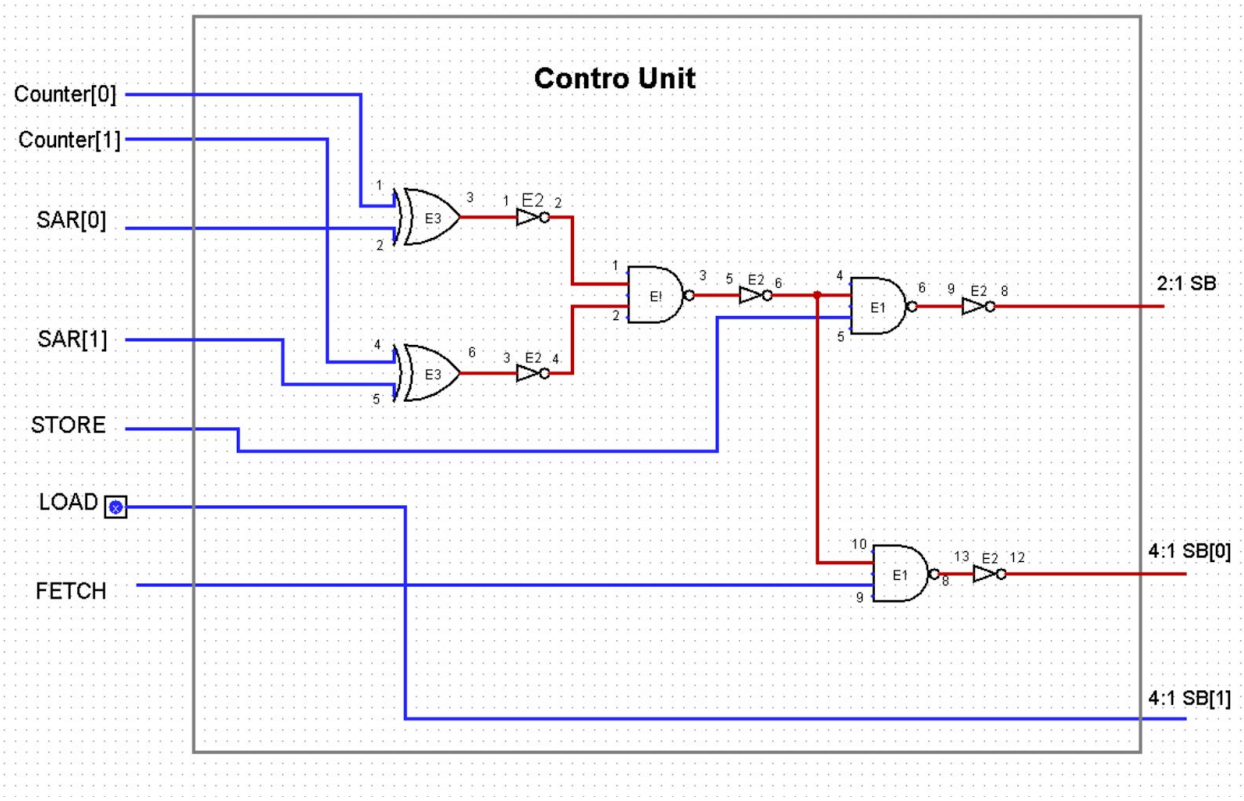
b. Include a high-level block diagram like figure 3 in the lab manual. This diagram should contain components on the granularity of registers, muxes, and blocks and include few/no individual gates.



4. Control Unit a. Provide an intuitive written description of your control unit. b. Include a block diagram of your control unit. It is acceptable to turn in either:

a. The control signal has 7 inputs. Counter[0] and Counter[1] are used to keep track of the addresses, and when they match signal SAR[0] and SAR[1] respectively the logic will output high into the 2:1 MUX selector signal whenever the STORE signal is high. The control logic also sends 2 signals to the 4:1 MUX whenever we do the LOAD or FETCH instructions.

b.



5. Design steps taken and detailed circuit schematic

a. If you used k-maps or truth tables during design, include them here. (If you didn't need them, you don't need to include them).

SAR[0]	C0	AdrM
0	0	1

0	1	0
1	0	0
1	1	1

SAR[1]	C1	AdrM
0	0	1
0	1	0
1	0	0
1	1	1

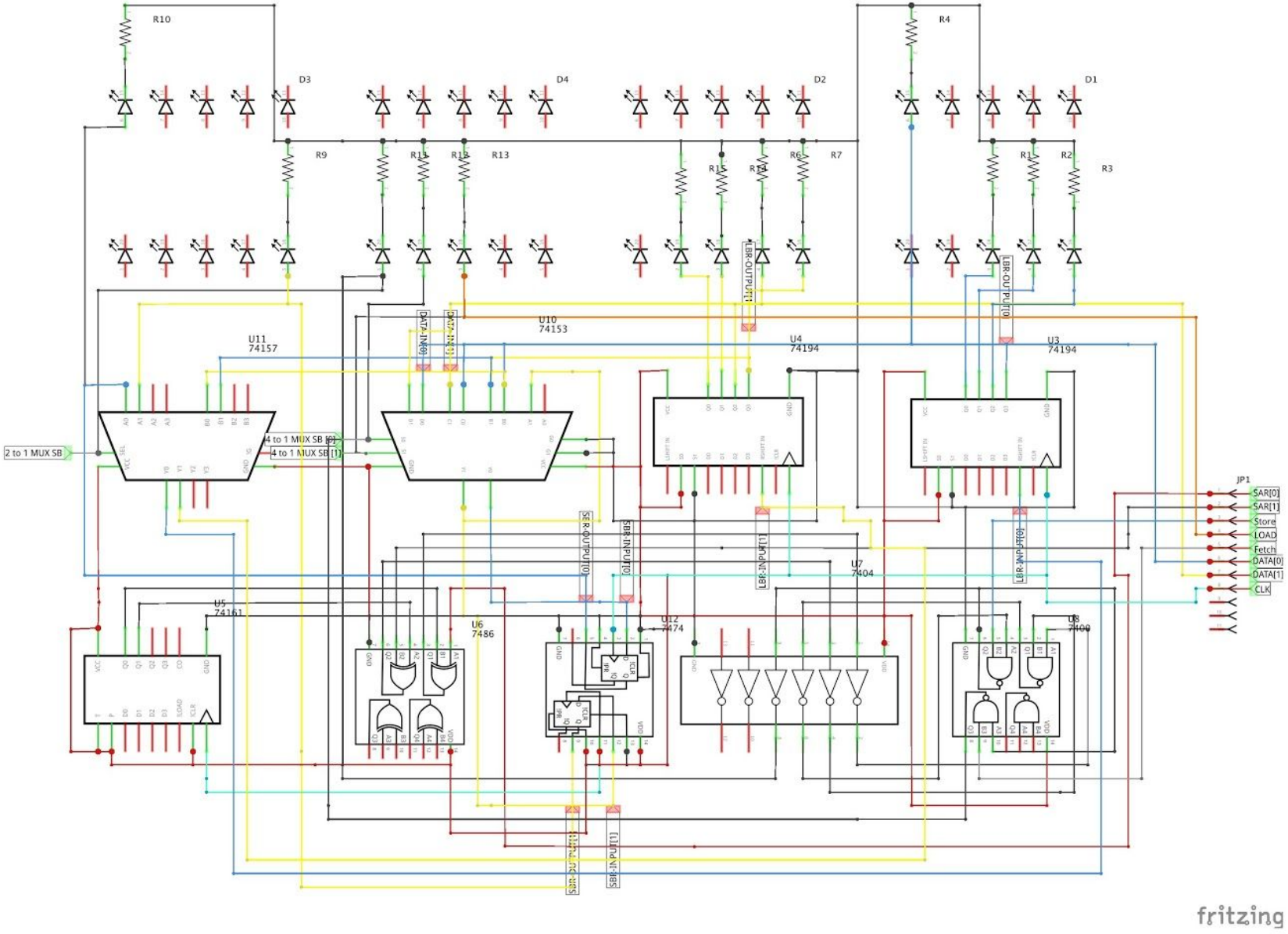
b**. You do not need a state diagram for this lab.**

c. Written description of the design considerations taken (did you consider multiple implementations of the same circuit and the tradeoffs of each?)

We did consider multiple implementations of the same circuit. When we were building the SBR circuit, we were choosing the sequential component JK or D flip-flop. The tradeoff of the D and JK flip-flop is connection. Connection for D flip-flop is way simpler than JK flip-flops.

d. Detailed Circuit Schematic

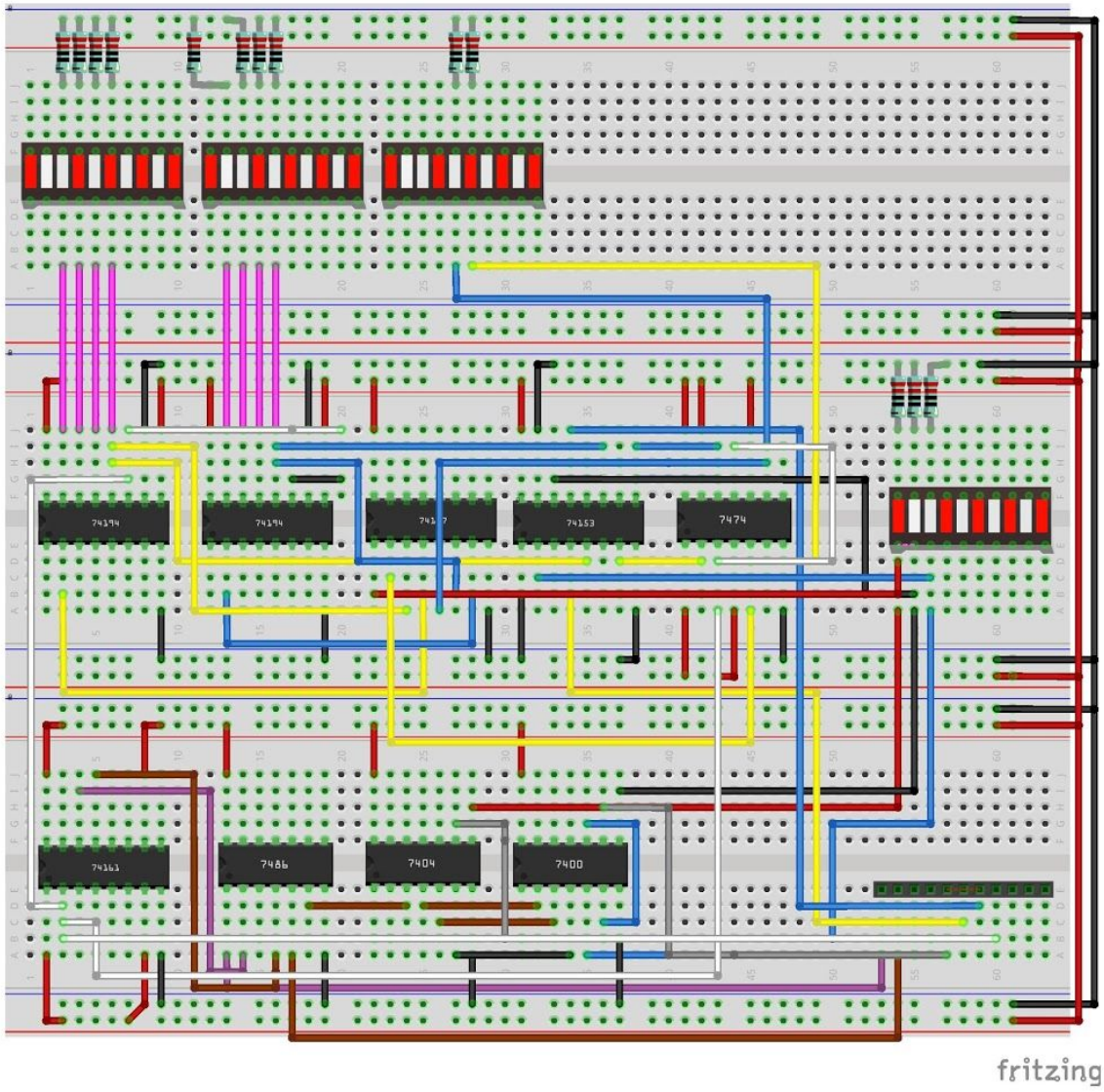
i. This diagram should show all components used and their interconnections down to the logic gate level. Large schematics can be broken down into a hierarchy (for example, in the main diagram, the control logic can be shown as a box with inputs and outputs and a separate detailed diagram of the control logic can be placed below the main diagram). You may omit gate level representations of complex chips which you used (counters, etc.) and instead replace them with a block and all connections. This should be generated with Fritzing, but you must make sure the schematic is well


R10
D3
D4
D2
D1
R4
R9
R13
R12
R13
R15
R14
R6
R7
R1
R2
R3
U11
74157
U10
74153
U4
74194
U3
74194
DATA-INPUT[0]
DATA-INPUT[1]
SBR-OUTPUT[1]
SBR-OUTPUT[0]
2 to 1 MUX SB
4 to 1 MUX SB [0]
4 to 1 MUX SB [1]
SER-OUTPUT[0]
SBR-INPUT[0]
LBR-INPUT[1]
LBR-INPUT[0]
U7
7404
U5
74161
U6
7486
U12
7474
U8
7400
JP1
SAR[0]
SAR[1]
Store
LOAD
Fetch
DATA[0]
DATA[1]
CLK
SER-INPUT[1]
SBR-INPUT[1]

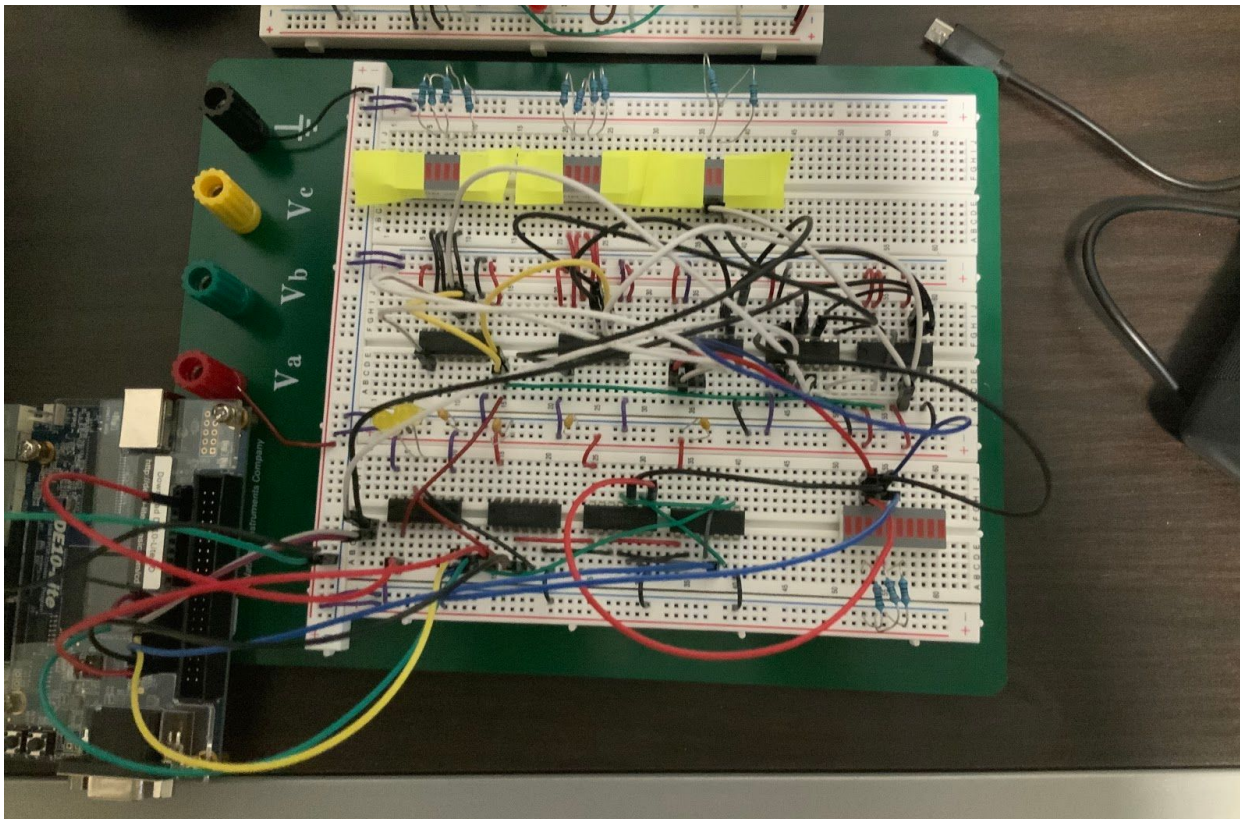
fritzing

6. Component Layout Sheet / Photograph of your breadboard design

a. For the online version of the course, it is required that you use Fritzing for the component view, which will be generated from the breadboard view in Fritzing. You will likely require multiple breadboards. Make sure your breadboard reflects your actual implementation and should look like an illustrated view of your breadboard photo.



b. Your lab report should include a photograph of your circuit as built on the breadboard. Individual wires need not be clearly visible as your component layout suffices for documenting the wiring.



7. Description of all bugs encountered, and corrective measures taken

One of the bugs that we encountered was the SBR didn't correctly get the data. First, we inspected the connection for 4 to 1 MUX and ensured that the select signal is wired correctly. Second, we inspected our connection and datapath flow is correctly wired on the 4 to 1 MUX. We inspected that all the connections were right, but we later found out that the D flip-flop was storing the other data. Then we checked our select signal and it was something with the select signal in the control unit.

8. Conclusion

a. In this lab we were able to learn the functionality of Random Access Memory and how we can use shift registers to create a storage unit. We also learned how to design a

Control Unit that is specific for our logic and that is able to utilize a counter as a way of keeping track of the addresses. We also learned real life applications of the NOP, LOAD, FETCH and STORE operations.

Pre-Lab and Post-Lab Questions:

Pre-Lab:

1. It is bad practice in Digital design to gate the clock because it might cause glitches like multiple clock edges where you expect only one edge.
2. Only the clock needs to be debounced because every time we flip the switch it can cause noise in the pulse therefore giving an incorrect output. And the debounced switch will make sure that our signal does not have any glitches.

Post-Lab:

1. **What are the performance implications of your shift register memory as compared to a standard SRAM of the same size?**

   The performance implications of our shift register memory compared to a standard SRAM is that in the Shift Register Memory we only have access to the leftmost and rightmost bits, and because of that we have to wait until our clock matches the address to be able to access that data. In the SRAM we can access any address at any time so it is faster than our Shift Register Memory. But our design has a lower cost compared to the SRAM and our logic is simpler so we are trading cost and easy implementation with speed.

2. **What are the implications of the different counters and shift register chips? What was your reasoning in choosing the pares you did?**

We have two types of counters which are synchronous up-down and binary counters. They both can fit our purpose for this lab. Synchronous up-down counters have an additional input for the counter which allows us to count up or count down.

We chose the binary counter because it is easier to connect and implement. We only have the bidirectional shift register, and we chose the 74194 also because it is easy to implement and wire our circuit. a 74195 bidirectional shift register also would work too, but it is more likely that it needs more combinations of logic to implement.