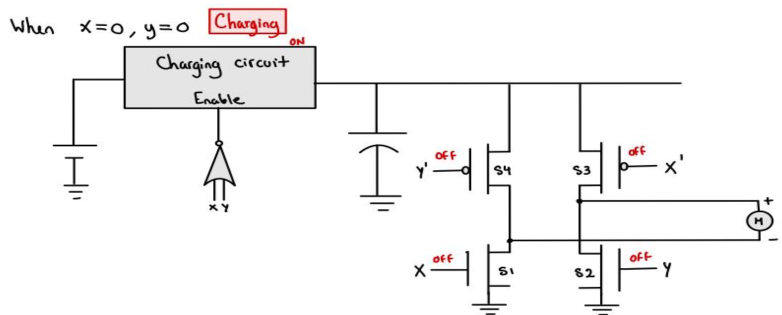
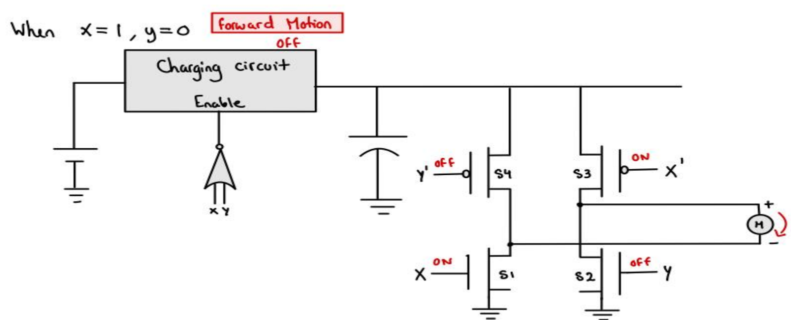
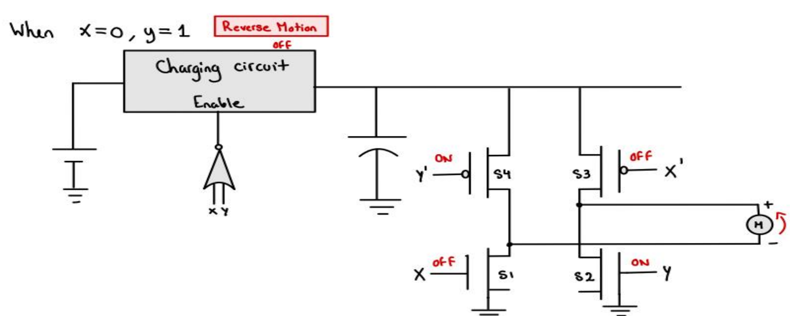
*1.2: Learning about Motor Drive Circuits*

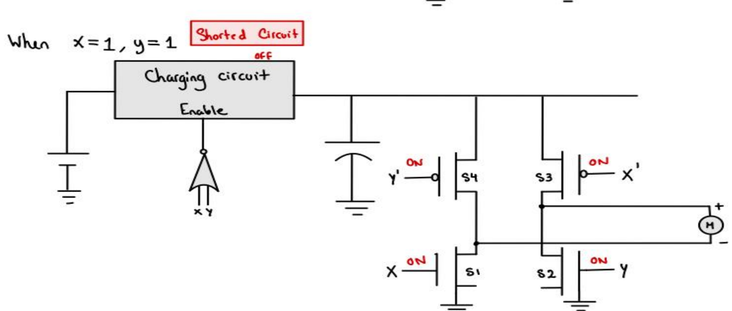
a. For the first lab activity, draw diagrams for each of the four combinations of the X and Y inputs. In your diagrams, indicate which switches are open and which are closed. For example, in the scenario when both X and Y are off, the charging circuit will be on and, therefore, the battery and capacitor are shorted together.

   i. Consider when X and Y are both equal to 1. Is there an issue with this specific combination? Explain.



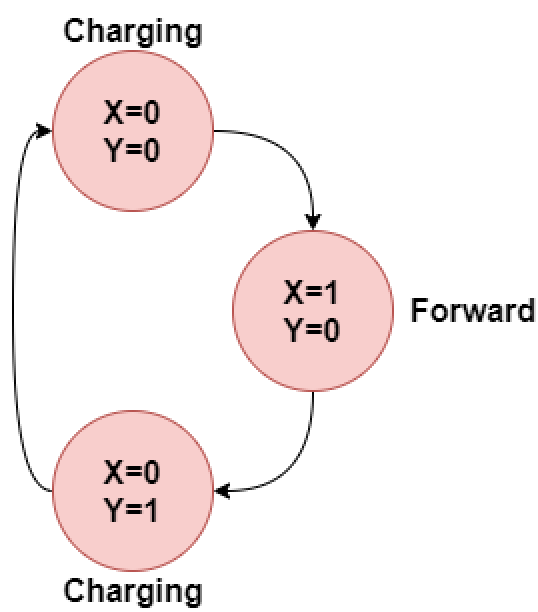
When X=1 and Y=1, all of the switches will be ON and the charging circuit will be off. This will cause the capacitor to be shorted to ground (0 V) causing a large fire that will destroy the component.

b. Translate your diagrams from part a to the truth table shown below. Similar to part a, specify the input combinations for X and Y, the corresponding state of each of the switches (ON/OFF) for each combination, and whether the charging circuit is enabled or not. The first row has been completed for you.

| **Inputs** | | **Switch Outputs** | | | |
|---|---|---|---|---|---|
| X | Y | S1 | S2 | S3 | S4 |
| 0 | 0 | OFF (0) | OFF (0) | OFF (1) | OFF (1) |
| 0 | 1 | OFF (0) | ON (1) | OFF (1) | ON (0) |
| 1 | 0 | ON (1) | OFF (0) | ON (0) | OFF (1) |
| 1 | 1 | ON (1) | ON (1) | ON (0) | ON (0) |

*1.3: Finite State Machines*

a. Draw, on paper, a state diagram like the ones shown in figures 4 and 2 for your motor drive state machine to follow the sequence charge→forward→reverse and repeat the cycle. Remember to label each node/state in your diagram with an appropriate name describing its function and include the binary data corresponding to each state.




b. How many flip-flops will you need for this state machine?

Ceil($\log_2(3)$) = 4 flip-flops

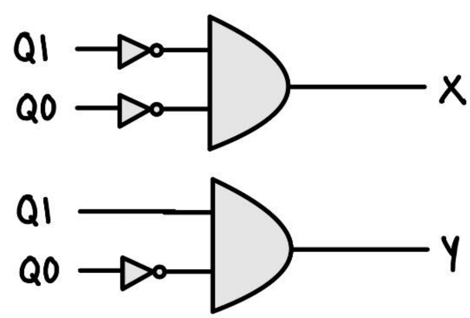
c. Complete a state transition table like the one in table 1 to help you design the combinational logic for your motor drive state machine. After this, write down the logic equations for the X and Y outputs in your transition table.

| Current State | | Next State | |
|---|---|---|---|
| **Q1** | **Q0** | **X** | **Y** |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 |

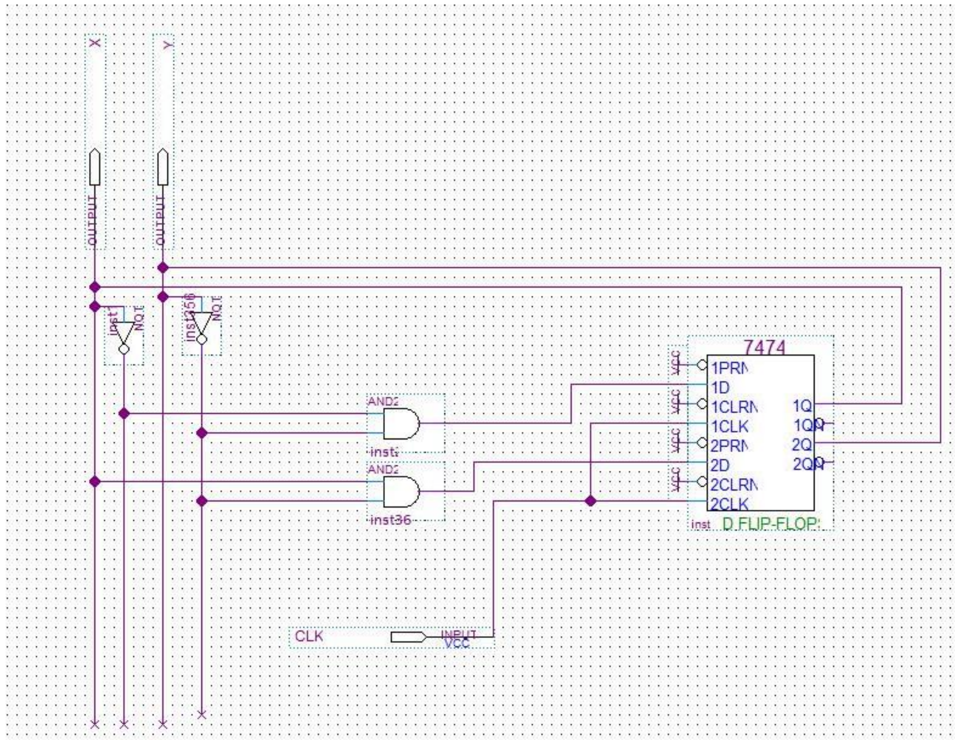
$$X = /Q1 */Q0$$

$$Y = Q1 */Q0$$

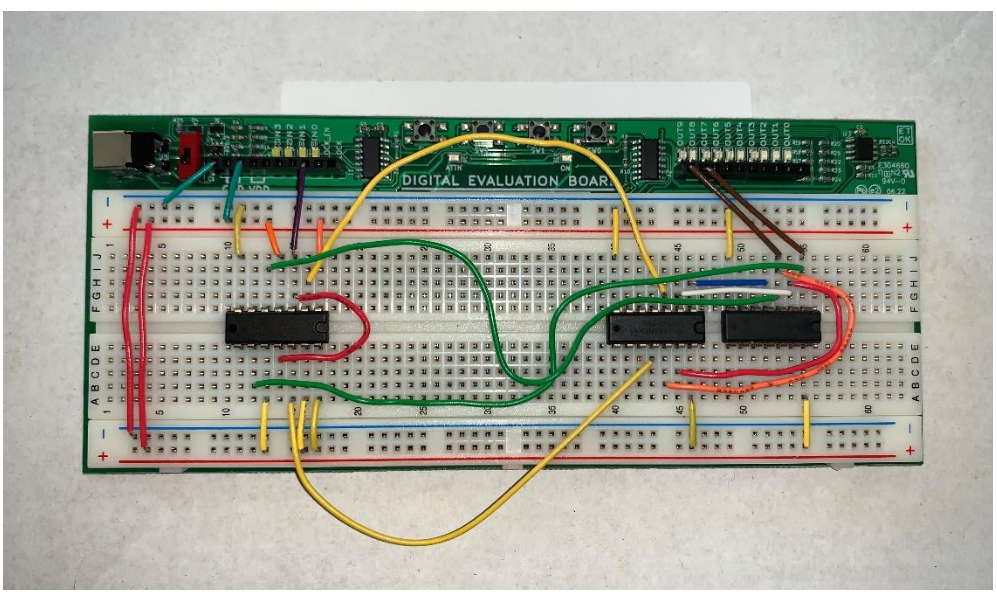
d. Design the combinational logic portion of your state machine based on your logic equations from part c.




e. Incorporate flip-flop(s) to your combinational logic to complete your motor drive state machine design. Remember to include an input clock signal for your flip-flop(s) and to connect preset and clear signals to VCC to deactivate them.




f. Build your state machine on your DEB using appropriate switches for your inputs. For outputs, use OUT9 for X and OUT8 for Y. Check to yourself that your state machine correctly outputs the desired sequence.




g. After designing your motor controller, consider the possibility of X=1 and Y=1 occurring simultaneously. Why won't this combination occur? Or, if it occurs in your design, what modification would you make in your design to prevent it from happening? Explain in words, no need for a prototype.

If designed correctly, the finite state machine never reaches this combination because it just follows the sequence and then repeats so no other combination is possible in the sequence. However, if it does occur in the design, the reset signal of the flip-flops is designed to reset a sequence to its initial state, therefore, this signal can be used to prevent the X=1 Y=1 combination from happening.