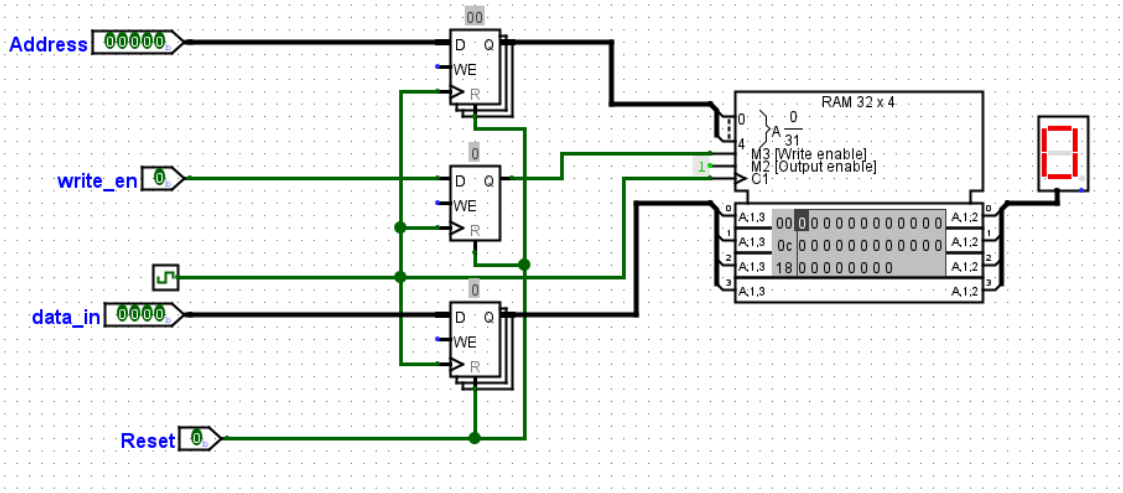
CSC258 Lab7 Report

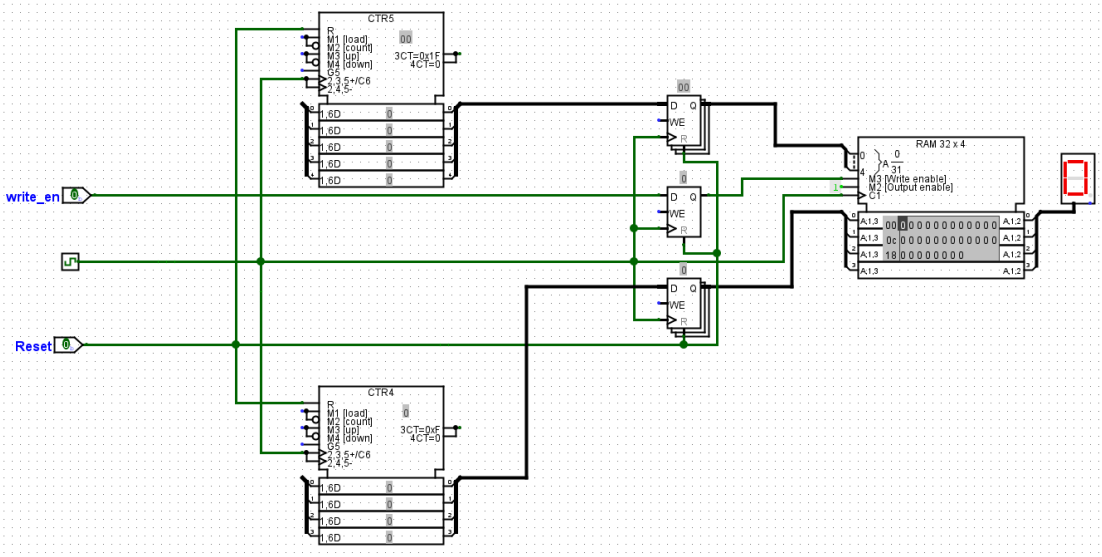
Tianle Wang

Part I

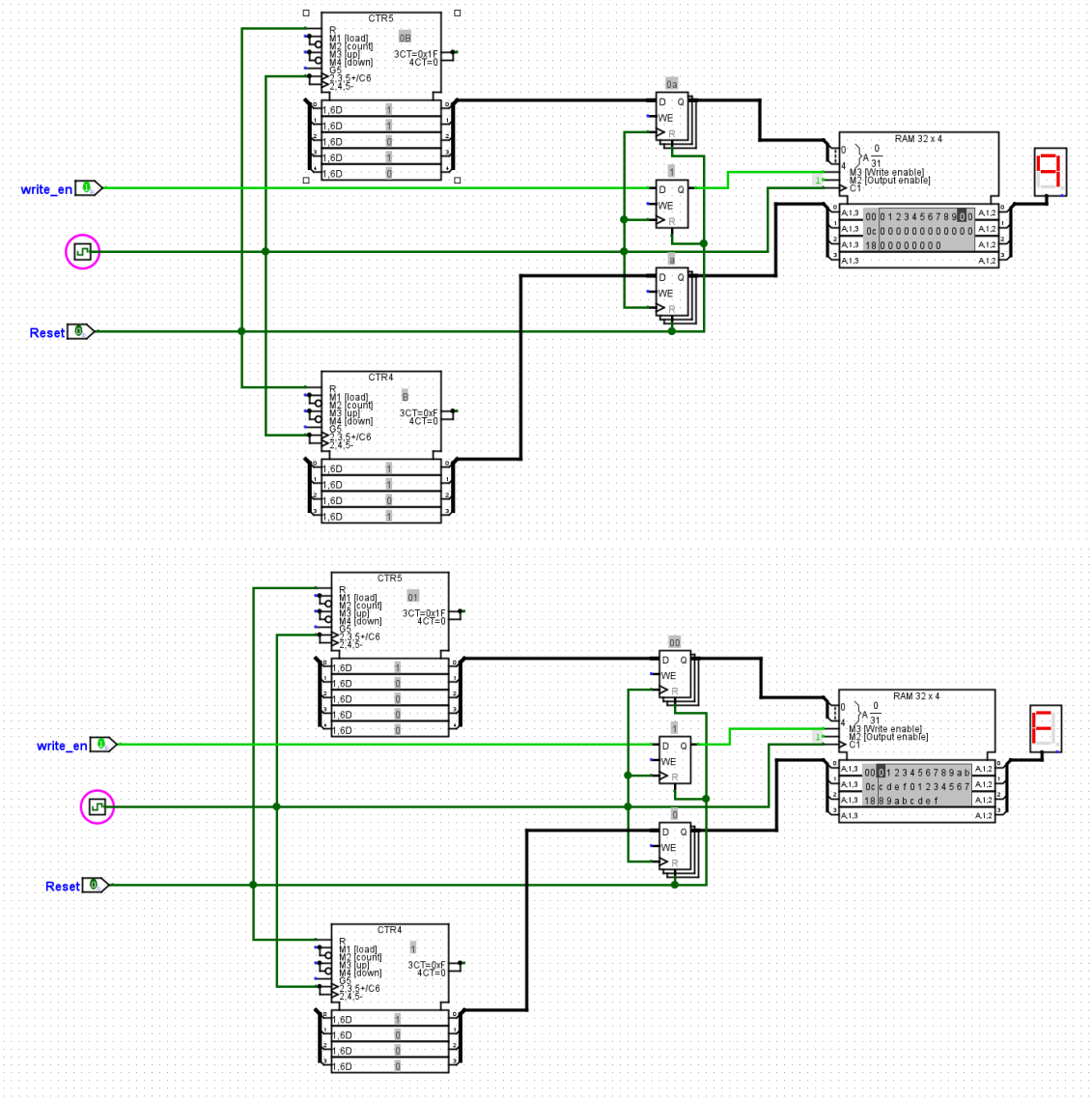
1. Create your `32 × 4 RAM` module in Logisim. You can find the component under *Memory* in components. Then you should adjust the address bit width and data bit width width in *Properties*.



2. Notice that the `RAM` module in Logisim has both Write Enable (for writing to memory) and Output Enable (for reading from memory). What happens if both signals are off when the clock goes high? What happens when both signals are on? Experiment with these signals to confirm your guess and write the behaviour in your prelab report.
    - If both signals are off when clock goes high, then only the pointer to address will change, no input or output for the address of memory
    - If both signals are on when clock goes high, then the pointer to address will change, change the address of memory as `data_in` and output for the address of memory
3. Create circuitry around this RAM unit that fills each location in memory with increasing values, starting with a zero value at memory location 00000, a value of 1 at memory location 00001, and so on. Once you get to memory location 10000, start again with a value of zero and keep writing increasing values until you write the value F at memory location 11111.



4. Connect the output of the memory unit to a seven-segment display.
5. Draw a schematic describing this circuit as part of your preparation.
6. Test your modules with Poke and the Logisim clock to verify its correctness. Include a few screenshots that shows the contents of the memory unit during your simulation.

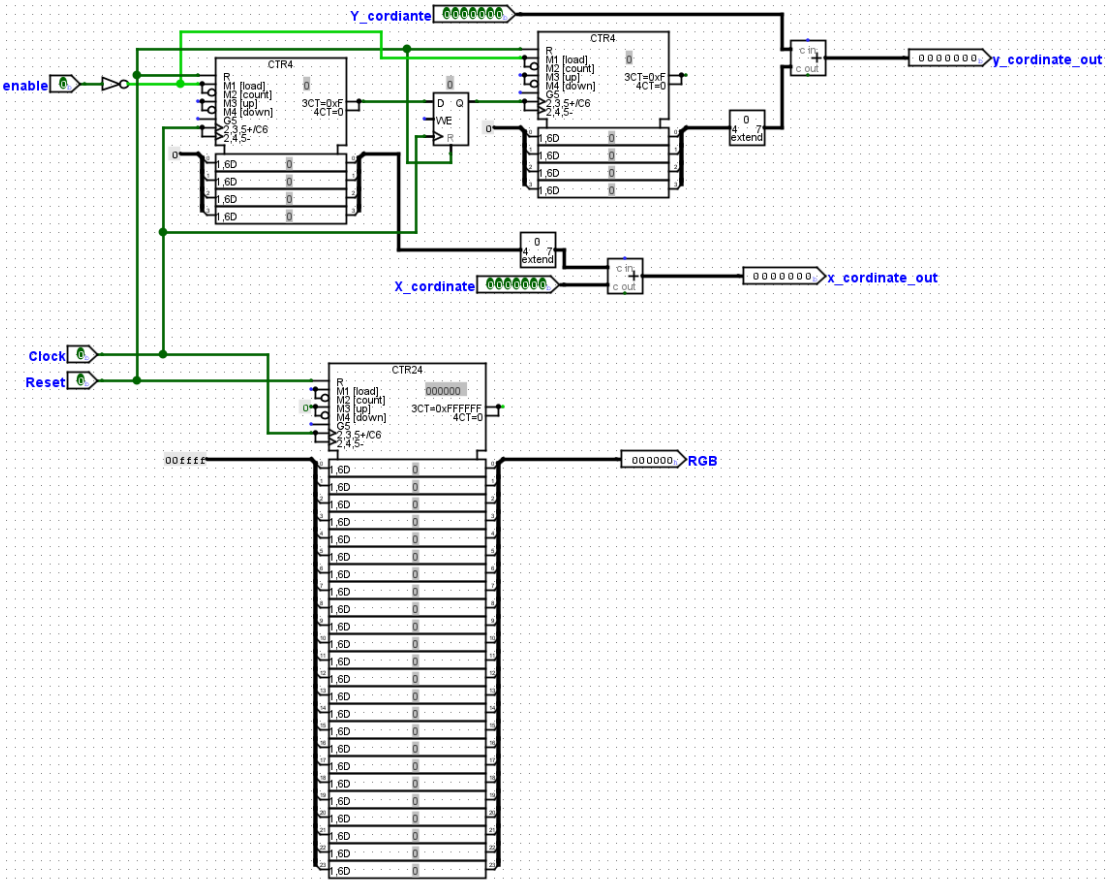


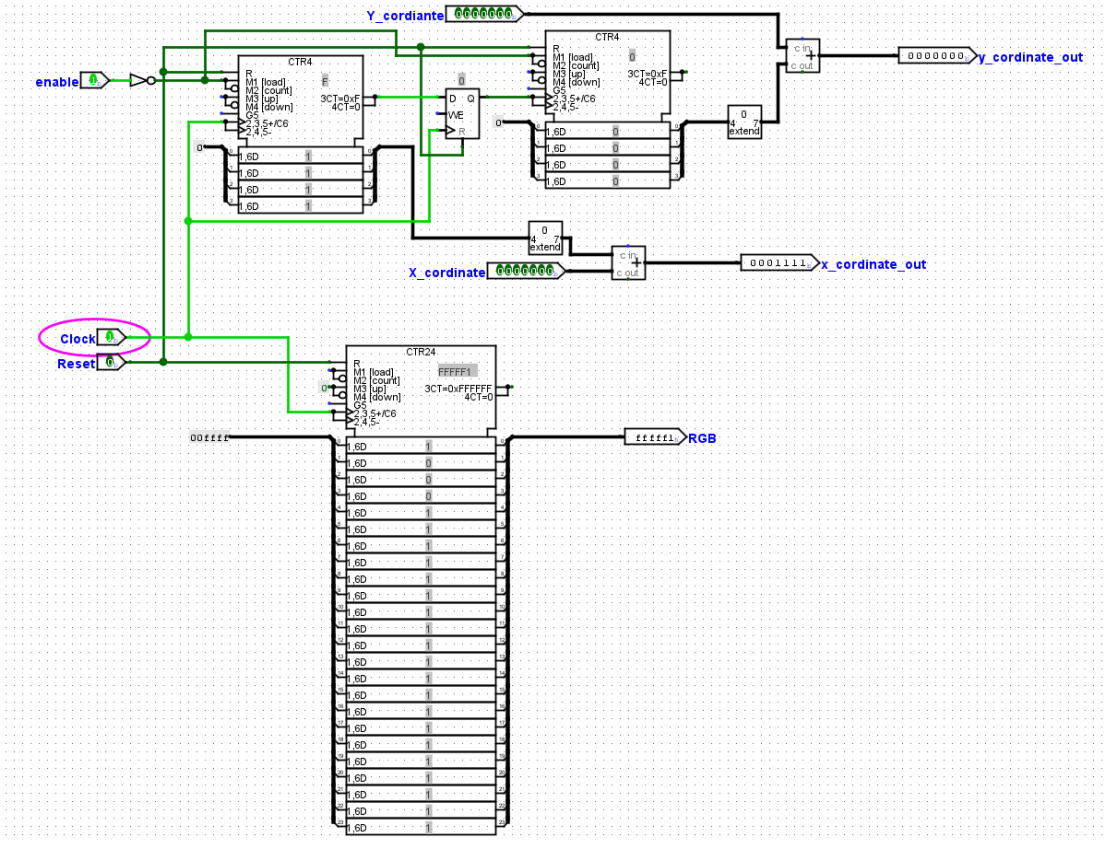
Part II

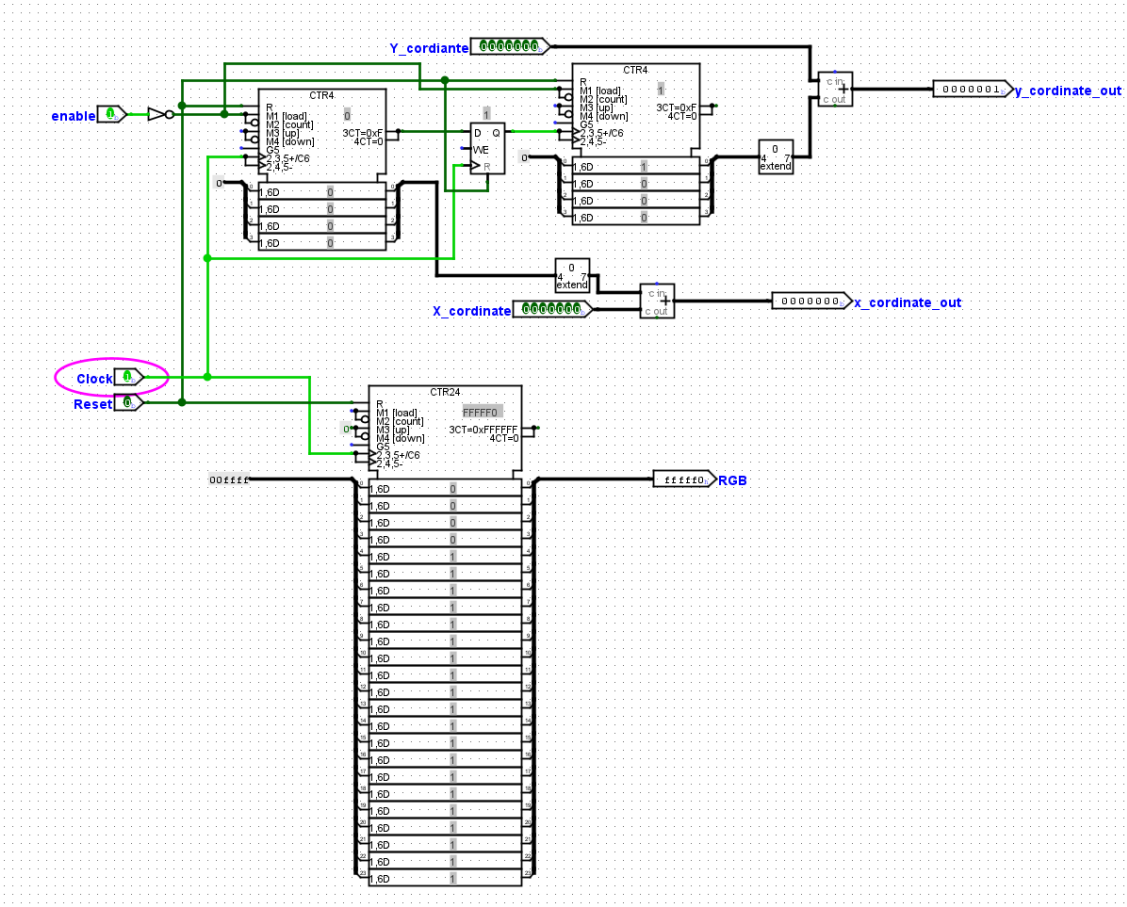
In your prelab report, answer the following questions:

1. What happens if you don't turn Enable off before updating X and Y?
    - Since `enable` is in charge of drawing, if it off, then updating X and Y won't draw anymore.
2. What happens if you turn Enable off before 256 clock cycles have passed?
    - It will stop drawing at the current location, 'till enable on.
3. What happens if you turn Reset on while Enable is on?
    - Since `reset` is asynchronous, then everything won't work.

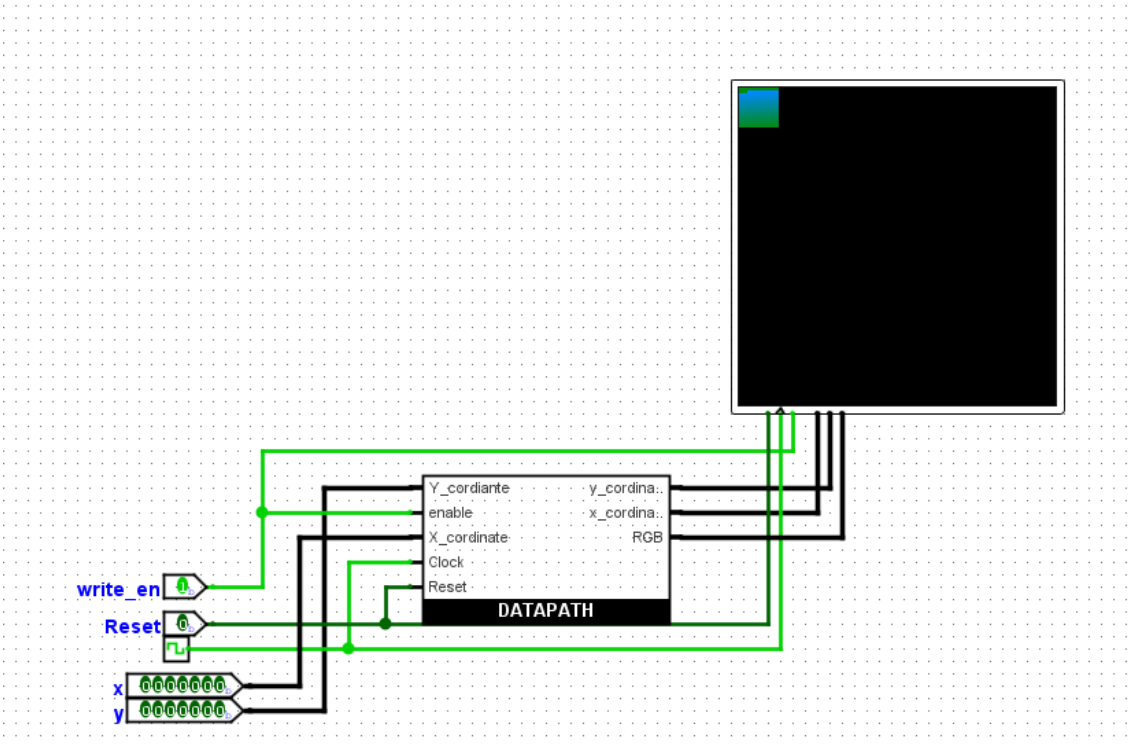
1. Draw the schematic of a `datapath` that implements the required functionality and build it in Logisim. Simulate the `datapath` to confirm that it sends the correct output values, given the `datapath` control signals that you provide. Include schematics, Logisim file, and screenshots of simulation output in your prelab.

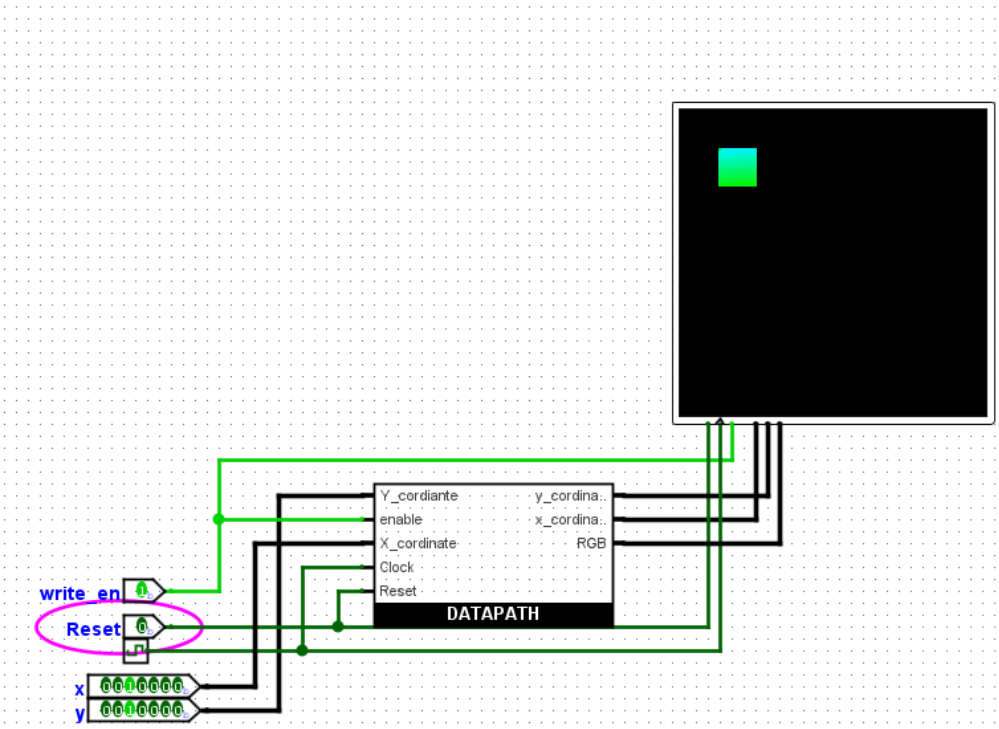

Y_cordiante
enable
CTR4
CTR4
0000000
y_cordinate_out
extend
X_cordinate
0000000
x_cordinate_out
Clock
Reset
CTR24
000000
3CT=0xFFFFFF
4CT=0
00ffff
000000
RGB

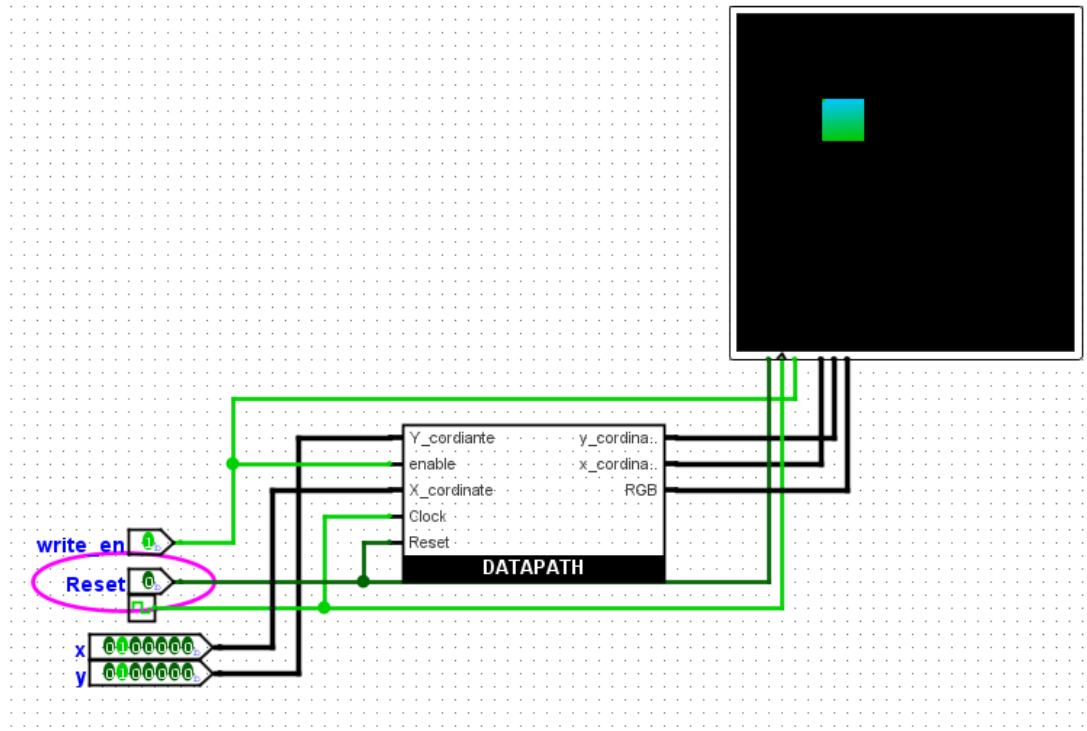

Y_cordiante
enable
CTR4
CTR4
0000000
y_cordinate_out
extend
X_cordinate
0001111
x_cordinate_out
Clock
Reset
CTR24
FFFFF1
3CT=0xFFFFFF
4CT=0
00ffff
ffffff
RGB

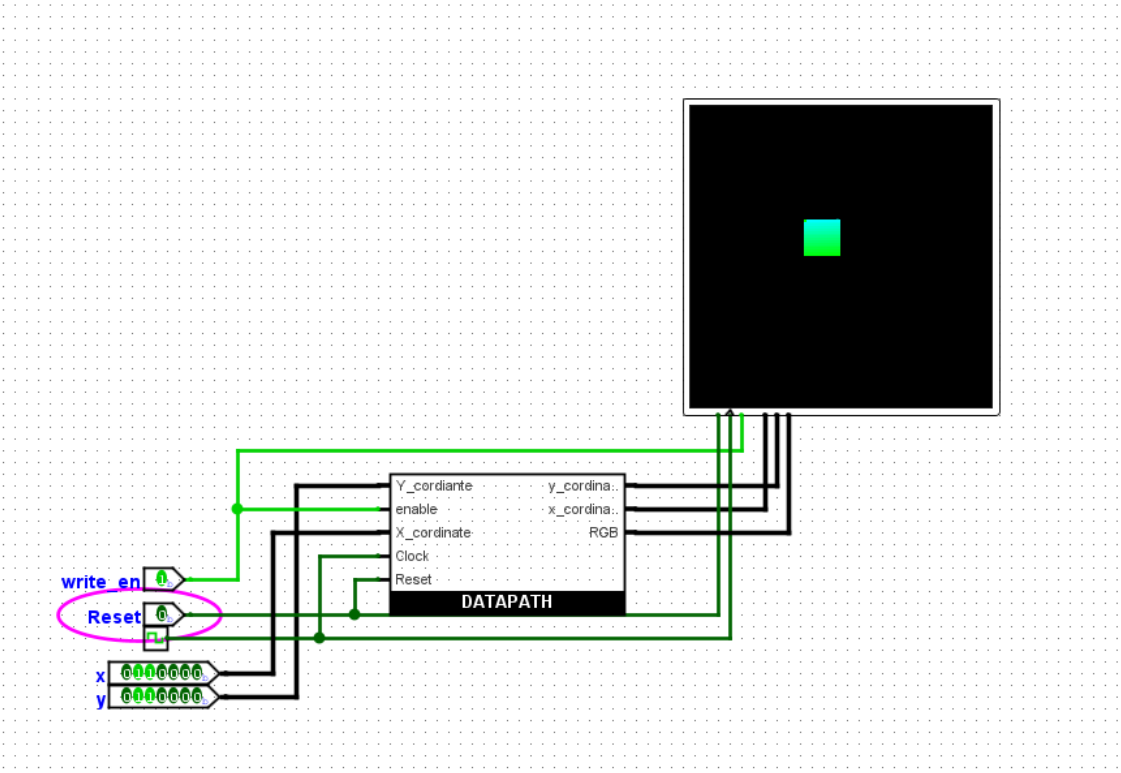


2. Design the FSM that sends the necessary signals to the `datapath` and then implement that FSM in Logisim. Your simulation for this part should only include the FSM.
    - Since the answer from Piazza says that "Anything is fine as long as it works", I do not setup FSM, and also worked.
3. Connect your FSM, `datapath` and the RGB Video together. You should simulate the combination of them using Poke( ) before connecting your circuit to the clock. Include screenshots of your simulation.




Y_cordiante
enable
X_cordinate
Clock
Reset
y_cordina:
x_cordina:
RGB
DATAPATH
write_en
Reset
x
y


Y_cordiante
enable
X_cordinate
Clock
Reset
y_cordina:
x_cordina:
RGB
DATAPATH
write_en
Reset
x
y



4. Demonstrate your design to the TA.