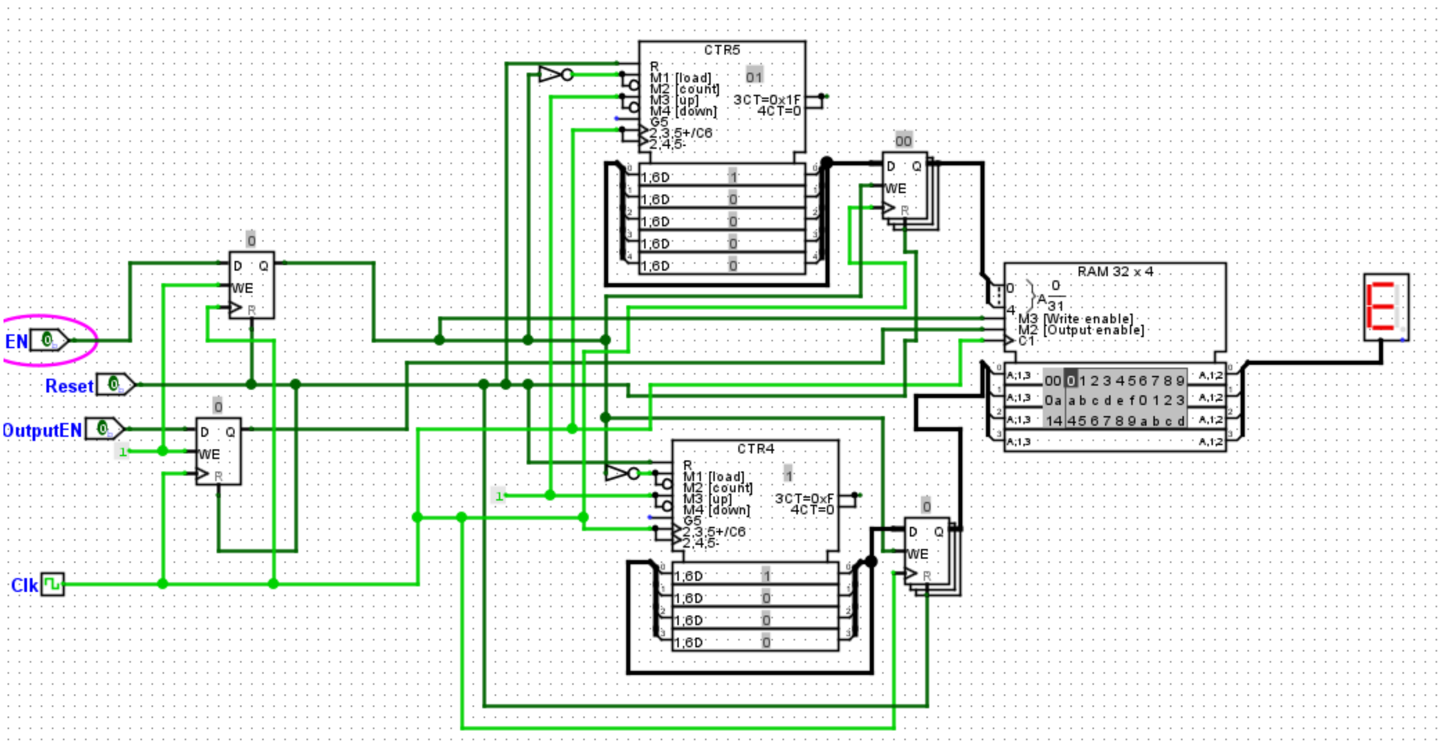
Tony Hong
1005109482
Nov 4, 2020

CSC258 Prelab (Lab 7)

Part 1: 32 x 4 RAM

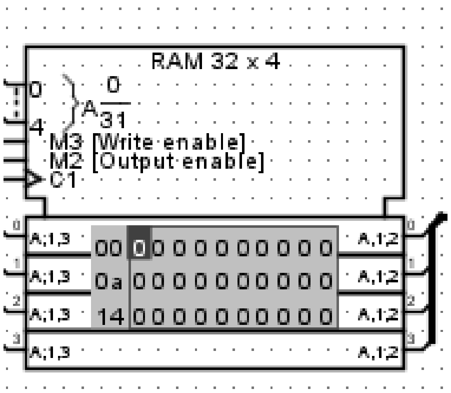
2. When both Write Enable and Output Enable are off when the clock goes high, no data is changed within the RAM while the output signal also stays the same. When both are on when the clock goes high, the register pointed to by the 5-bit address has its data overwritten to what is currently on DataIn and DataOut is also changed to match the change to that specific register.

Below is the diagram of the circuit I designed in Logisim:



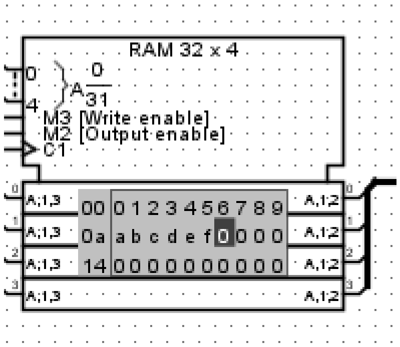
Below are some screenshots of the memory unit and its contents during testing:

1. At the start:



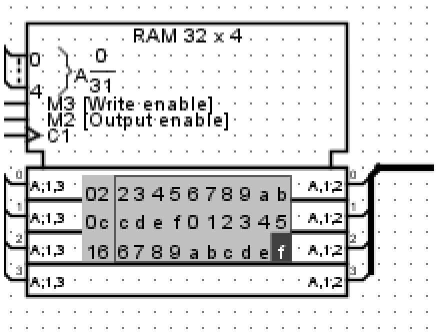
```
00 0 0 0 0 0 0 0 0 0 0
0a 0 0 0 0 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0
```

2. After 16 clock cycles:



```
00 0 1 2 3 4 5 6 7 8 9
0a a b c d e f 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0
```

3. After 32 clock cycles:

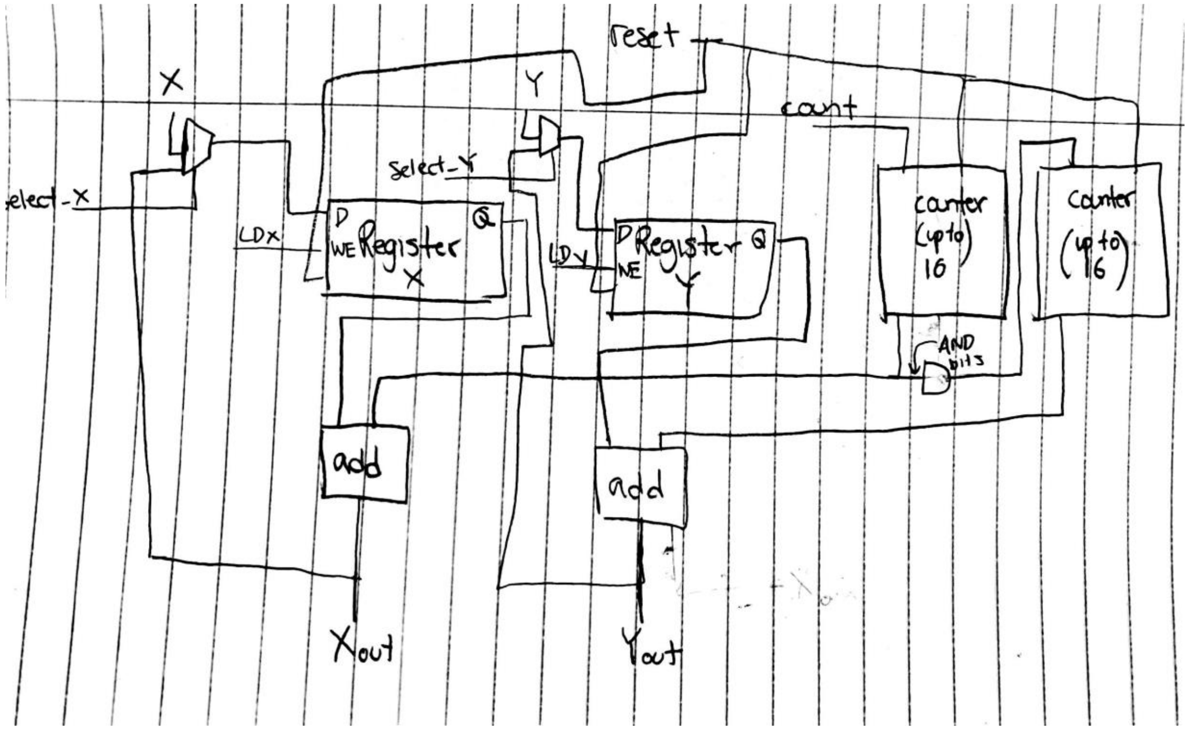


```
02 2 3 4 5 6 7 8 9 a b
0c c d e f 0 1 2 3 4 5
16 6 7 8 9 a b c d e f
```

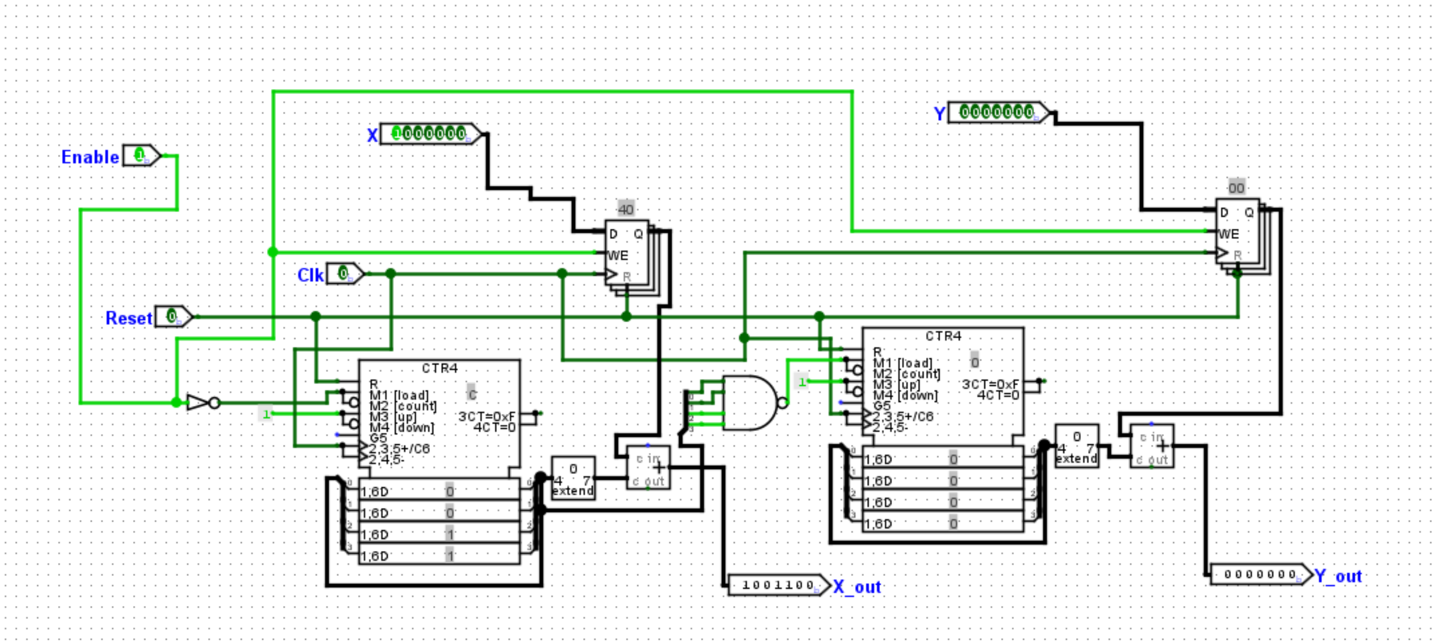
Part 2: RGB Display

1. If the enable signal is not turned off before updating X and Y, then it is possible that pixels not meant to be drawn are drawn on the RGB display. This is because the RGB display is always updating the screen due to the constant clock ticks. Therefore, pixels may be drawn during the process of changing X and Y.
2. If Enable is turned off before 256 clock cycles have passed, the 16 by 16 pixel square will not be complete and less than 256 pixels will have been drawn.
3. If reset is turned on when enable is on, the screen will be turned all black. When the reset is turned back off, then the square will be redrawn.

Below is a schematic for the Datapath that implements the functionality using 2 counters with one acting as a rate divider of the other.

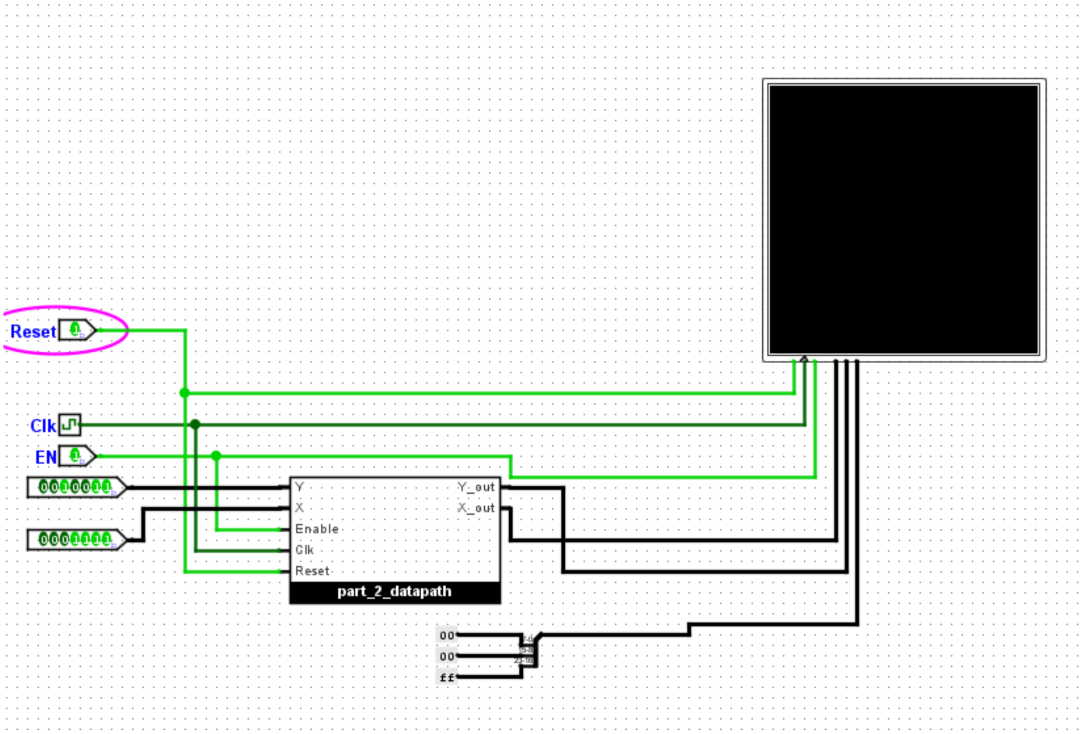


Below is a Logisim implementation of the Datapath above.

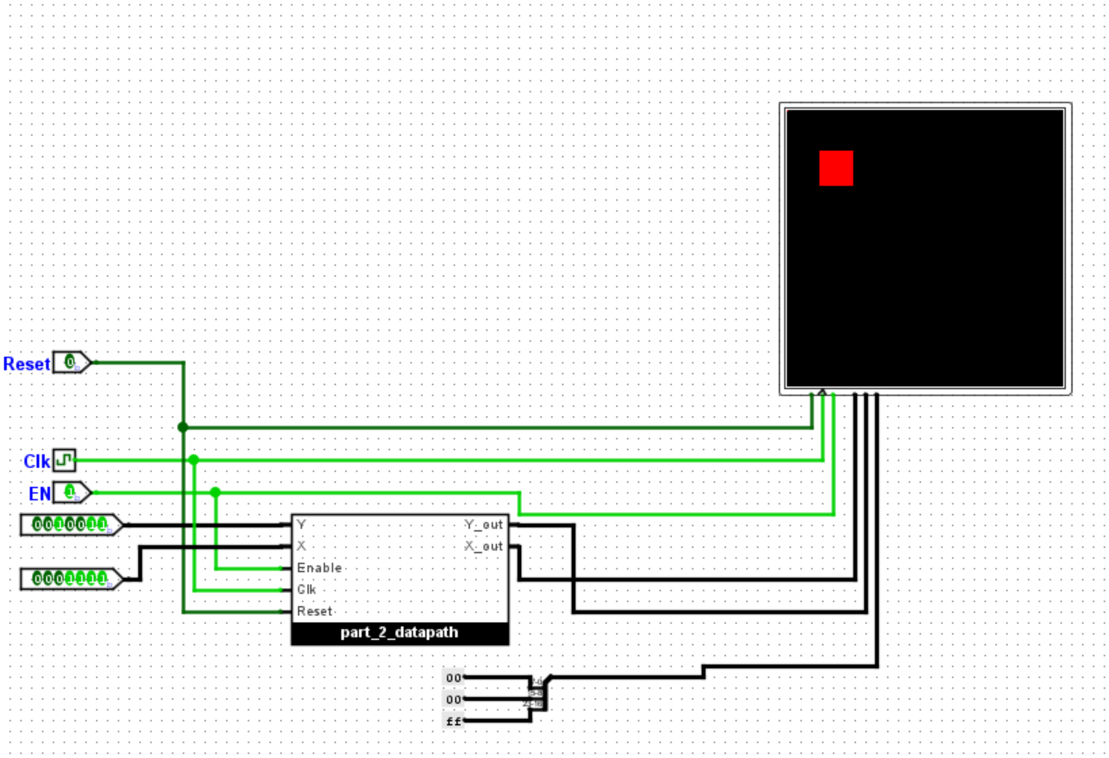


I did not use an FSM to design my solution. This is because the two counters in my Datapath already keeps track of the current location to draw based on X and Y and will automatically locate the next pixel by incrementing appropriately.

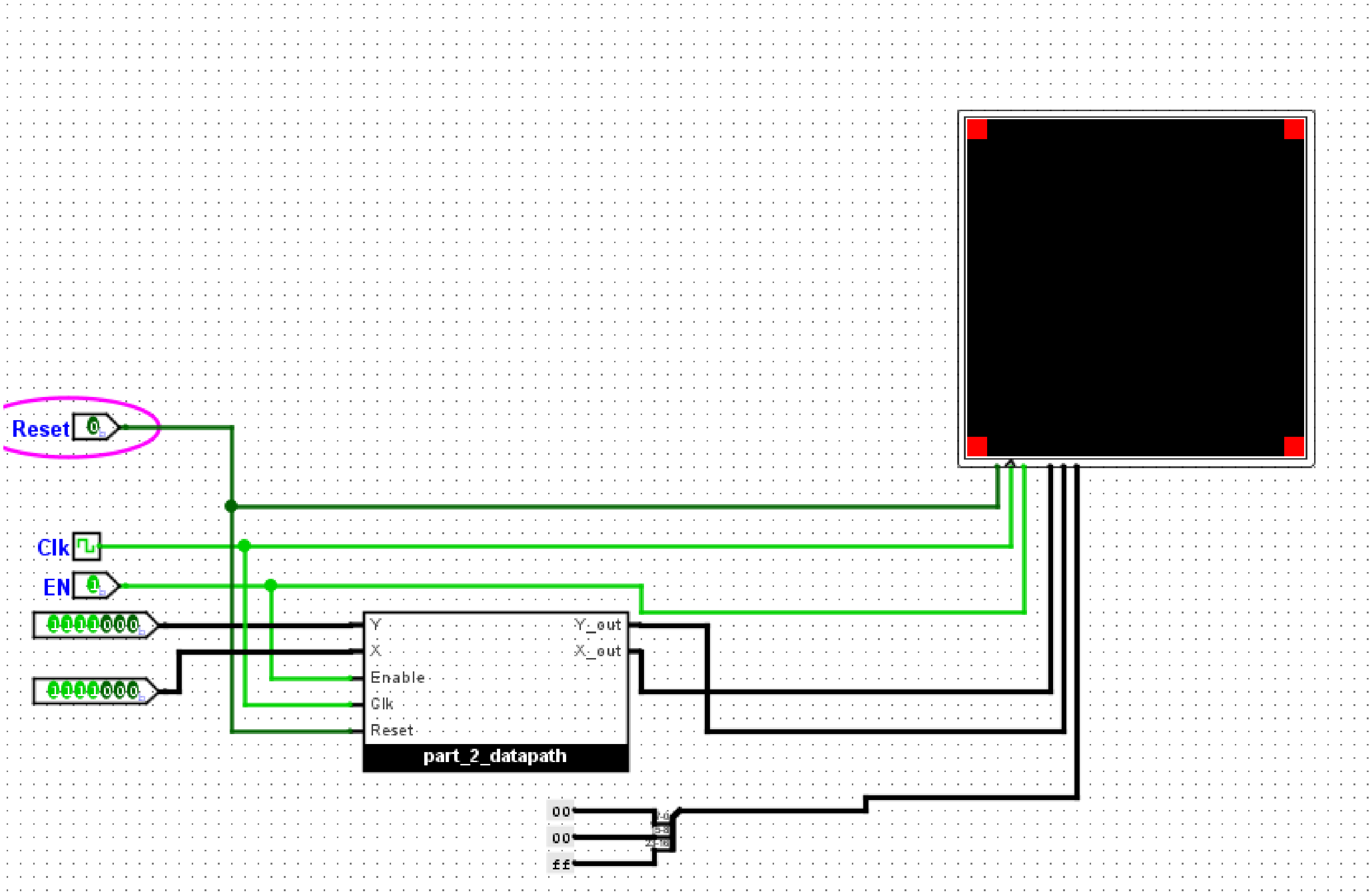
Below is my Logisim design for the high-level circuit that draws the square:



Below I test drawing a square with X=15 and Y=19



The Other test I conducted was drawing a square on the bottom right where there will be overflow to the top-right, top-left and bottom-left. Here X and Y are both 120.


Reset
0
Clk
EN
1
1111000
1111000
Y
X
Enable
Clk
Reset
Y_out
X_out
part_2_datapath
00
00
ff
7-0
15-8
23-16