

# **CDA 4213/CIS 6930 CMOS VLSI**

## **Fall 2019**

### **Final Project**

#### **Due date(s)**

**Partial Design Report:**

Week of 18<sup>th</sup> November

**Final Design Report:**

Monday, 9<sup>th</sup> December

Today's Date:	11/23/2019
Your Team Name:	Trenislav Wu
Team Members:	Trent Callahan, Denislav Tsonev, Boyang Wu
Work Distribution	<p>1) Trent Callahan - Created AND and MUX with Denislav. Debugged and tested 8x8 multiplier and shift registers. Drew gate-level designs for lab report and worked on the rest of the report.</p> <p>2) Denislav Tsonev - Created AND and MUX with Trent. Debugged and tested 8x8 multiplier and shift registers. Designed a logic design for the test case. Worked on report.</p> <p>3) Boyang Wu - Built 8x8 multipliers with existing gates and wired everything that was still not proper after the mosaic/copying. Debugged 8x8 multipliers.</p>
No. of Hours Spent:	50
Exercise Difficulty: (Easy, Average, Hard)	Average. The design was not hard with mosaics/copying but debugging wiring issues took a ton of time. Ended up scrapping an initial 8x8 design that worked.
Any Feedback:	If the images are too small try zooming in.

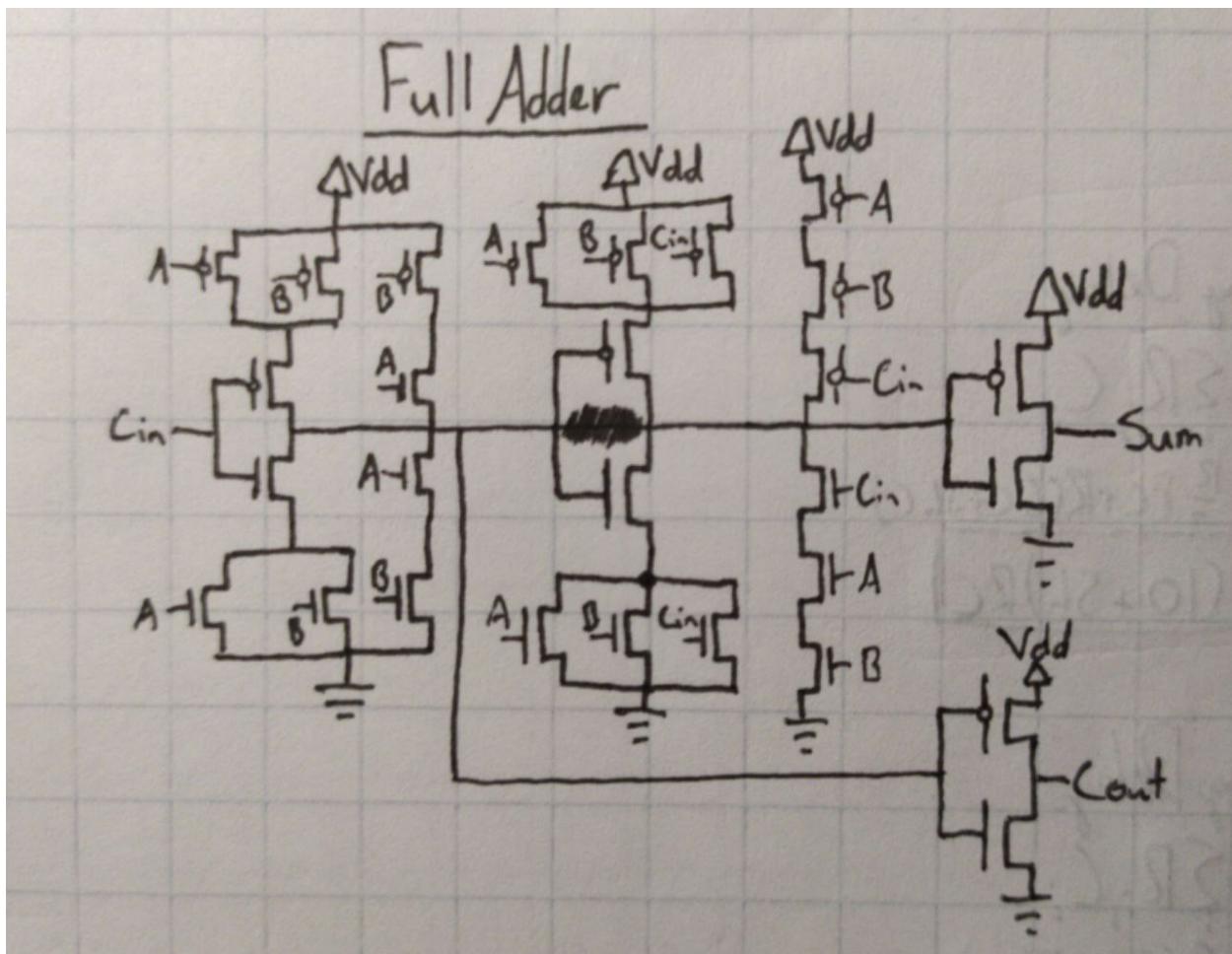
1) (10 pts) Proposed Design – Bit slice design

a) List all module bit-slices you have used for your design.

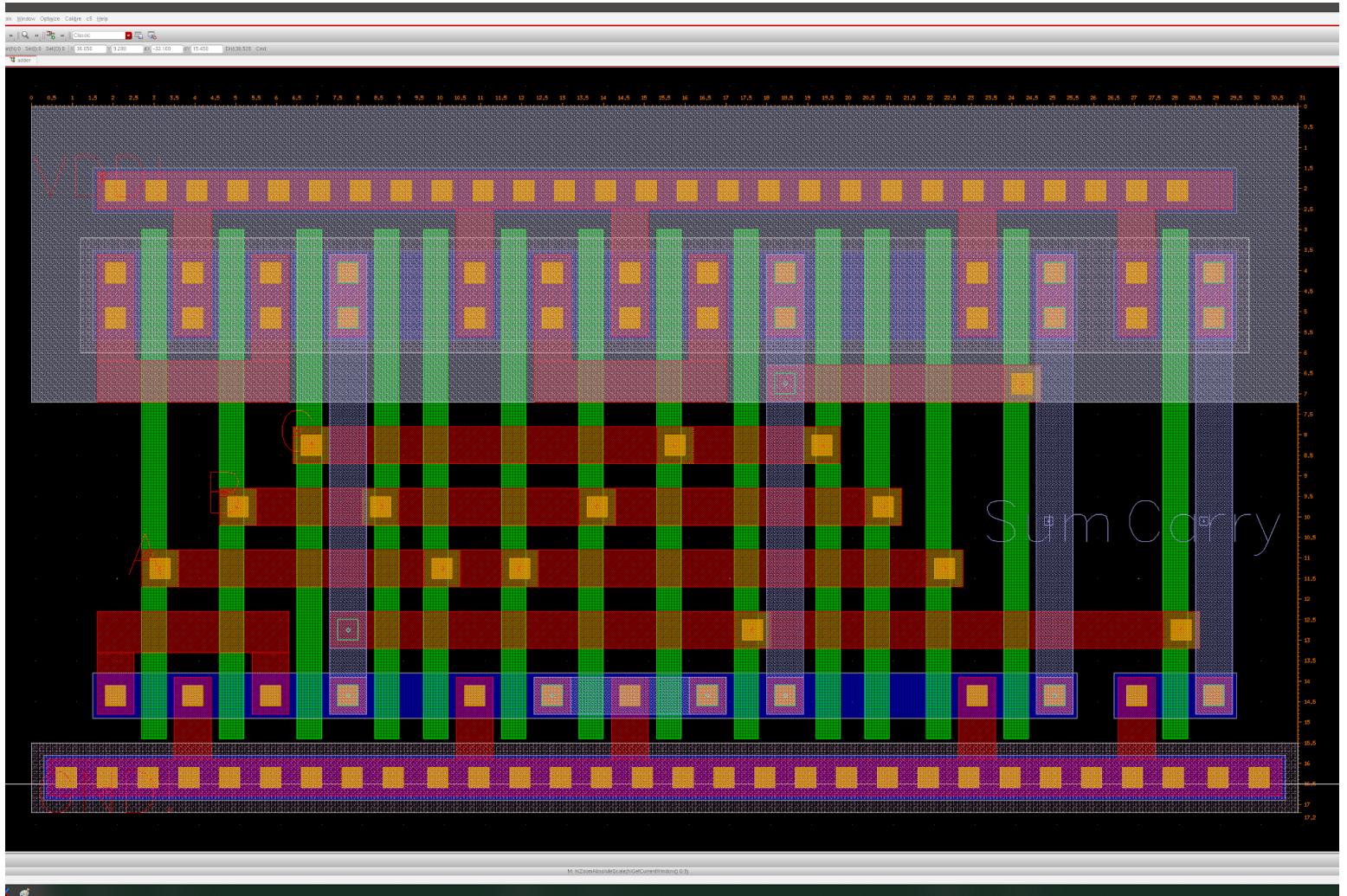
- i) Full Adder
- ii) AND-Adder
- iii) AND
- iv) 2-input MUX
- v) 8-bit Shift Register
- vi) 16-bit Shift Register
- vii) Ring Oscillator

b) (b) For each bit slice, show the gate-level design and layout design. For layout, include the snapshot from Cadence Virtuoso. If you have used any other blocks, include them as well.

#### I. Full Adder:

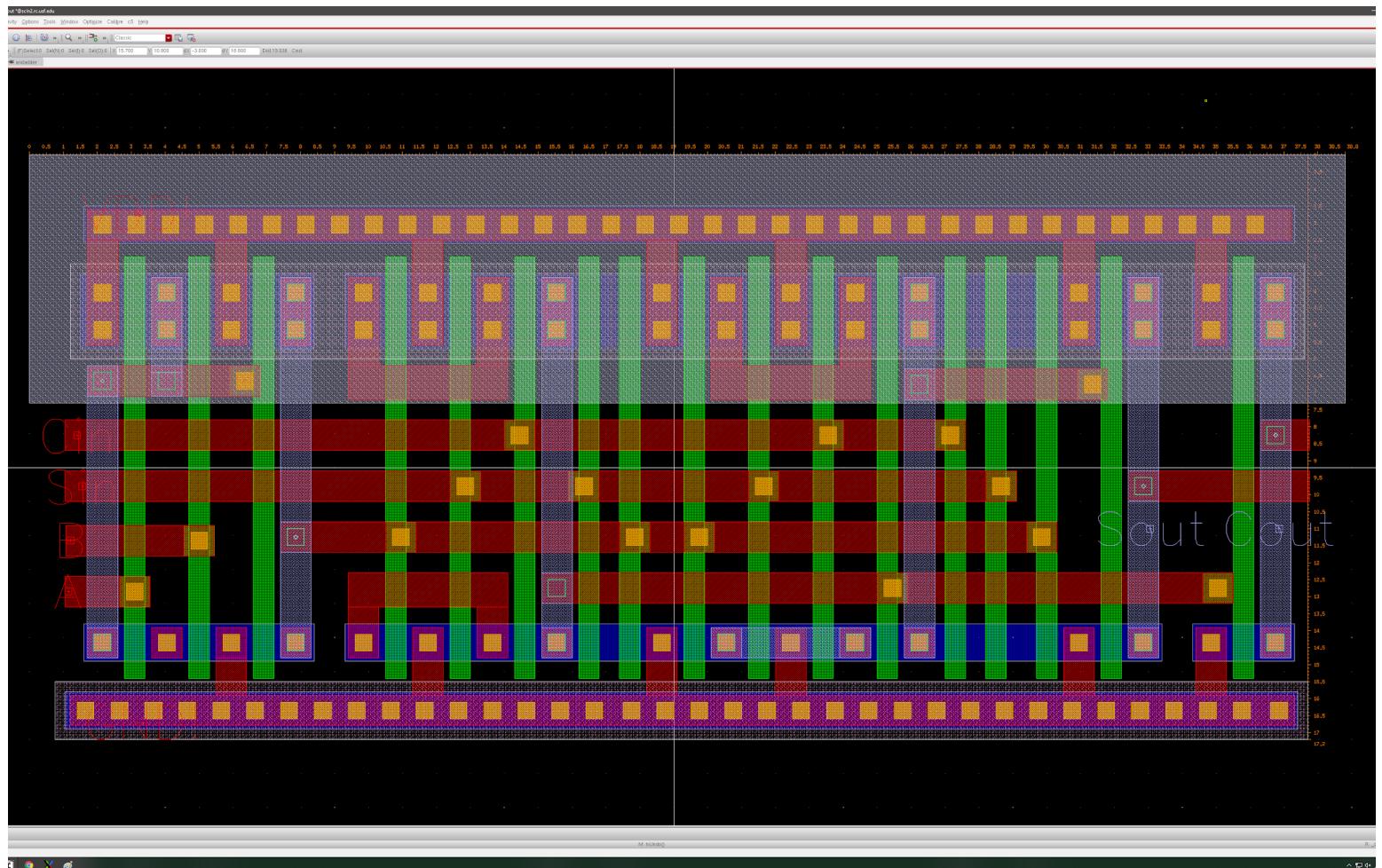
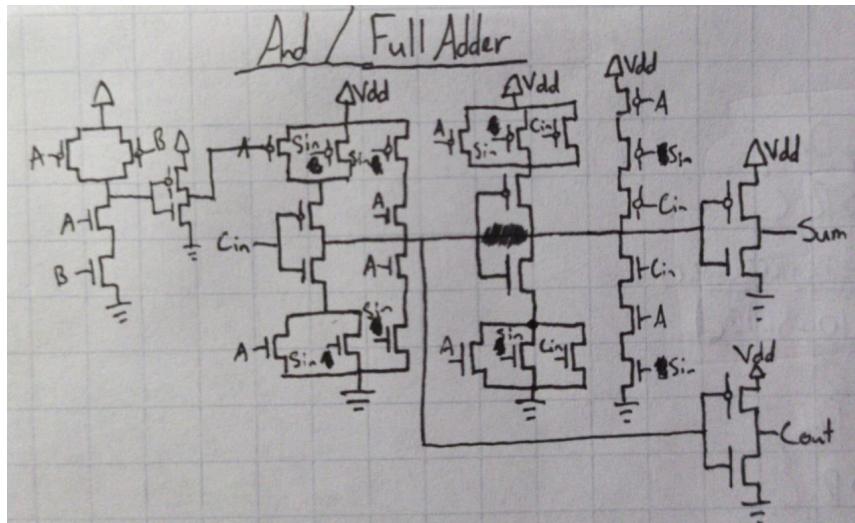


## Full Adder:



$$17.2 \times 31 = 533.2$$

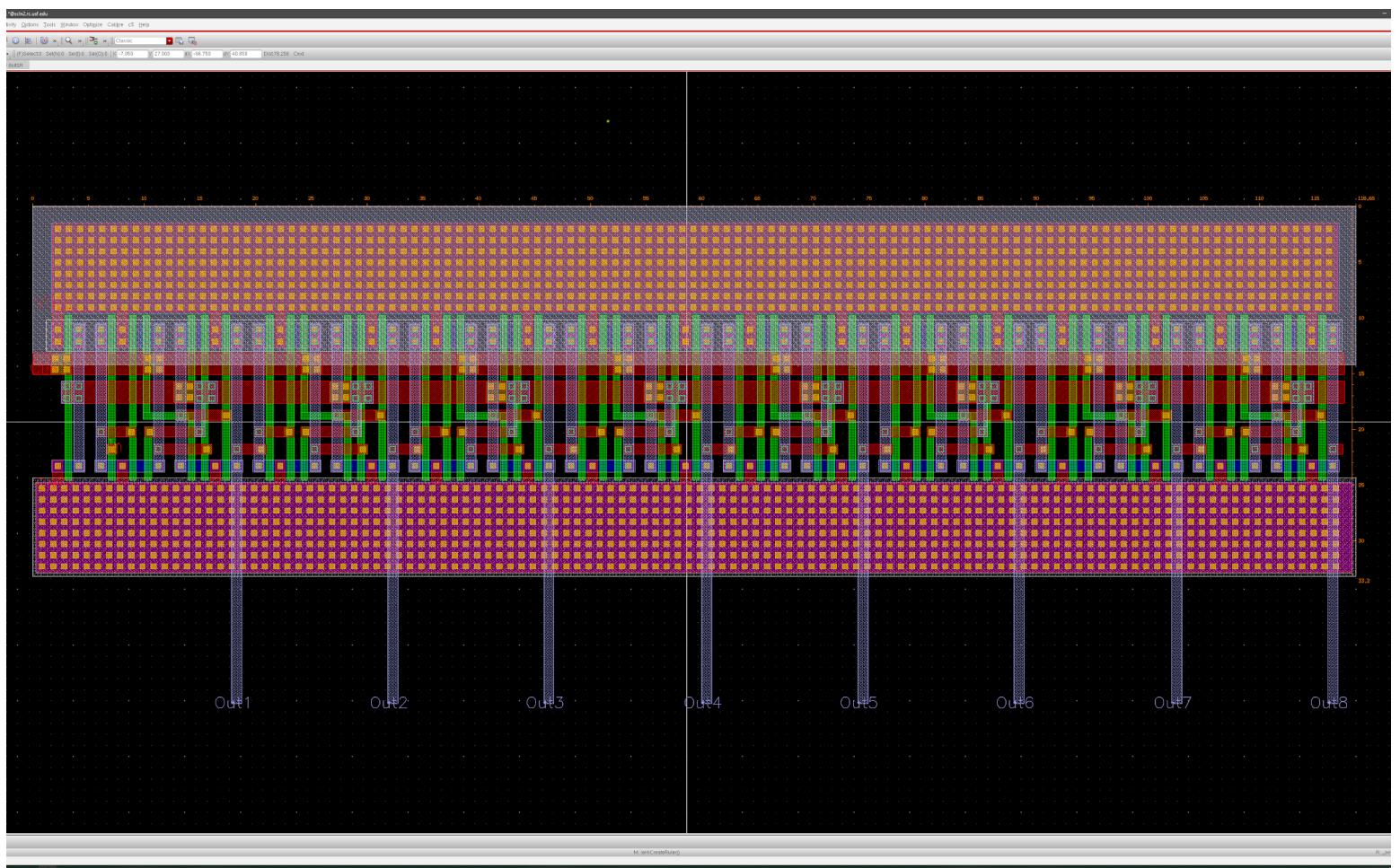
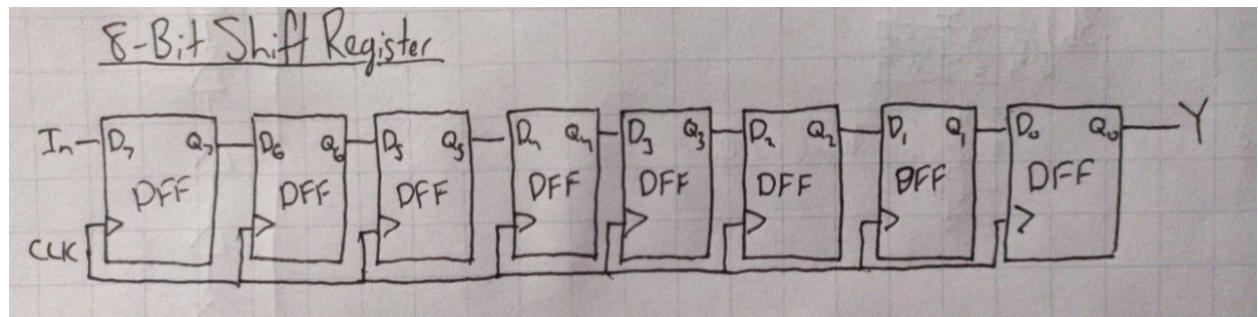
## II. Full Adder with AND gate:



$$17.2 \times 38.8 = 667.36$$

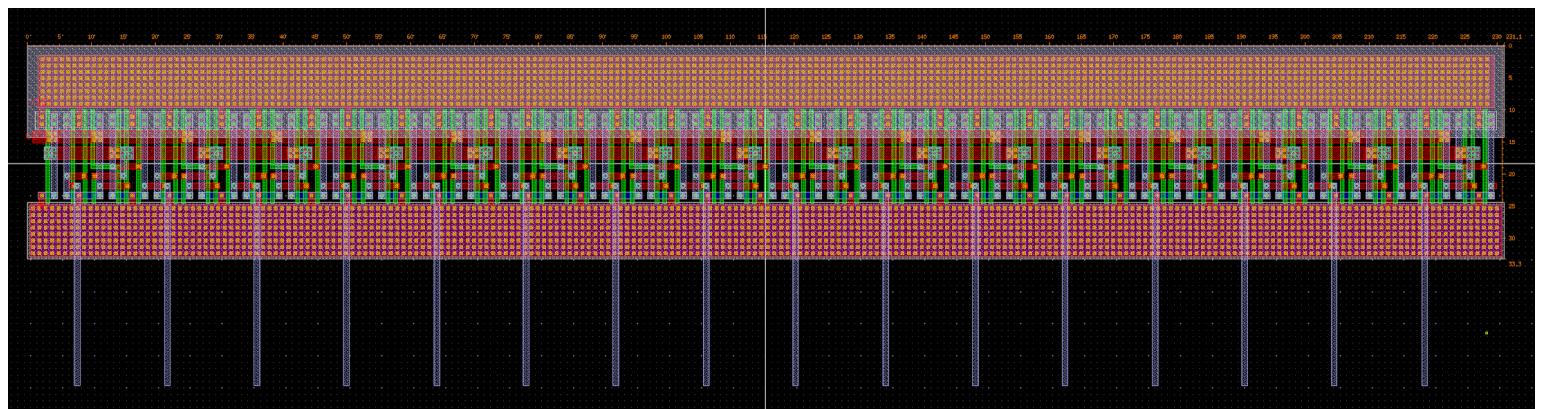
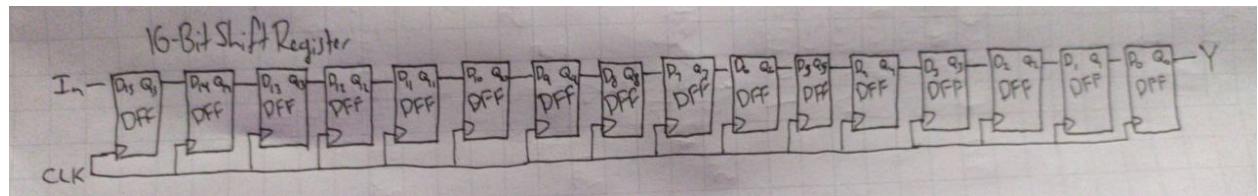
### III. Registers (Inputs and Output):

#### 8-bit shift register (with wires for parallel out and serial out)



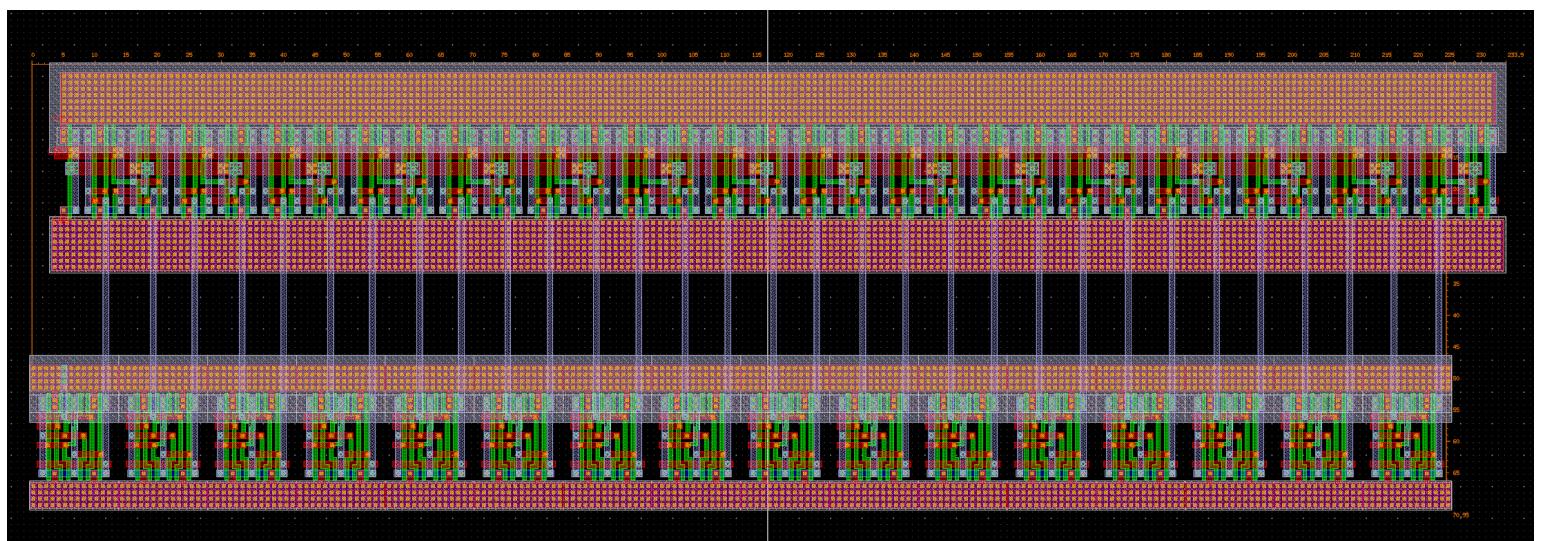
$$118.65 \times 33.2 = 3939.18$$

### 16-bit shift register (with wires for parallel out and serial out)



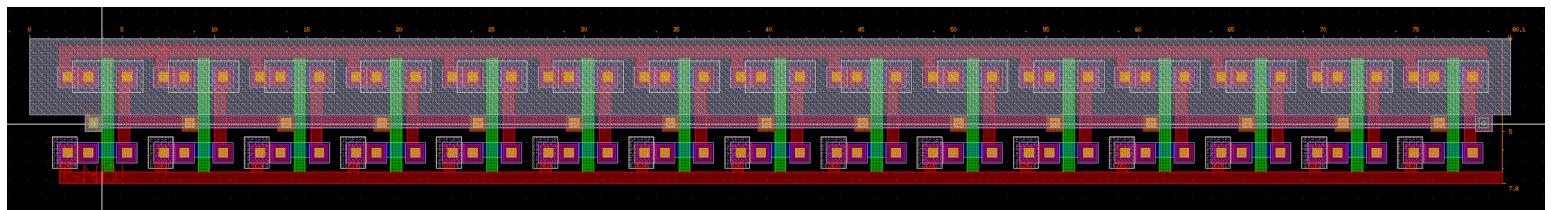
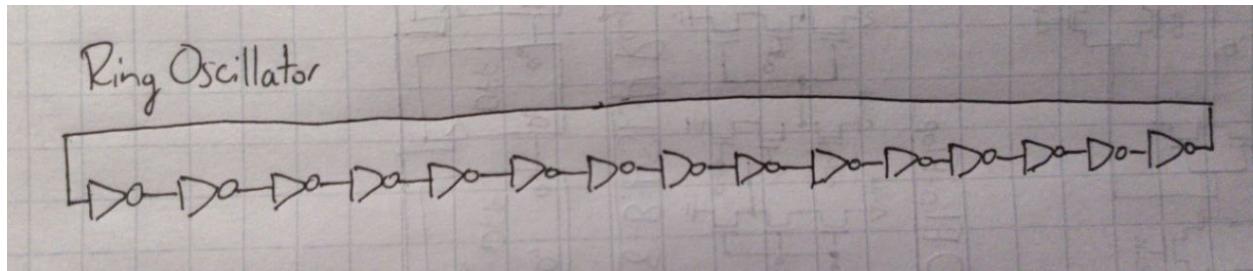
$$231.1 \times 33.3 = 7695.63$$

### 16-bit SR with 16 2 input multipliers (partial wiring for D1 in and Q out)



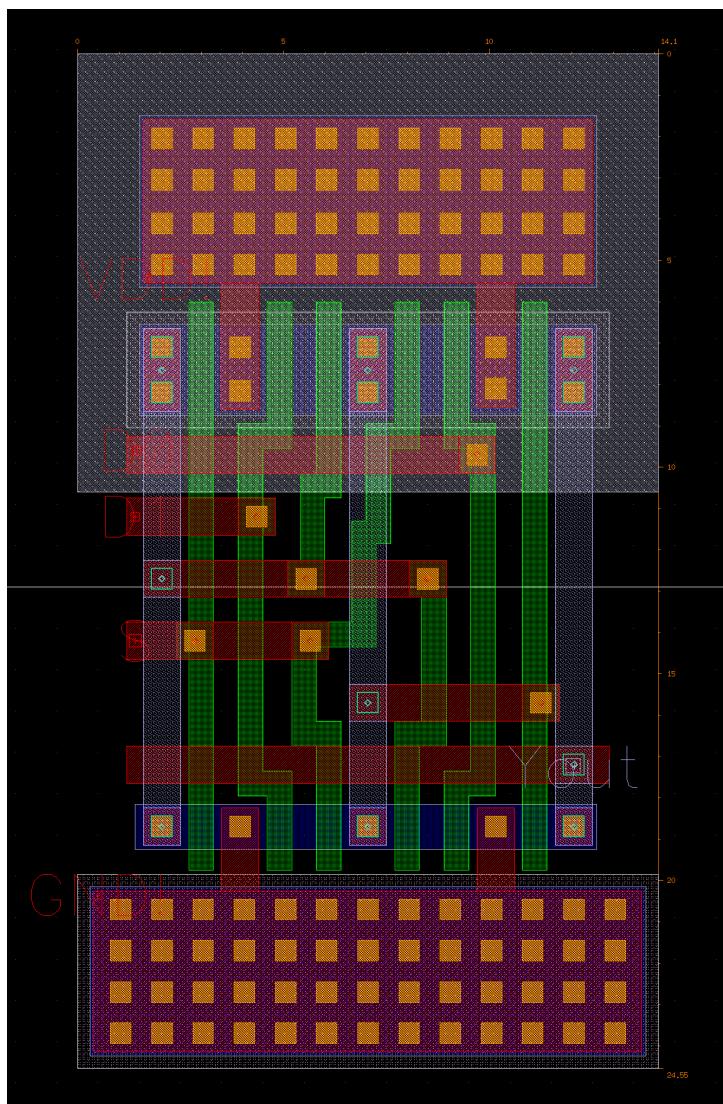
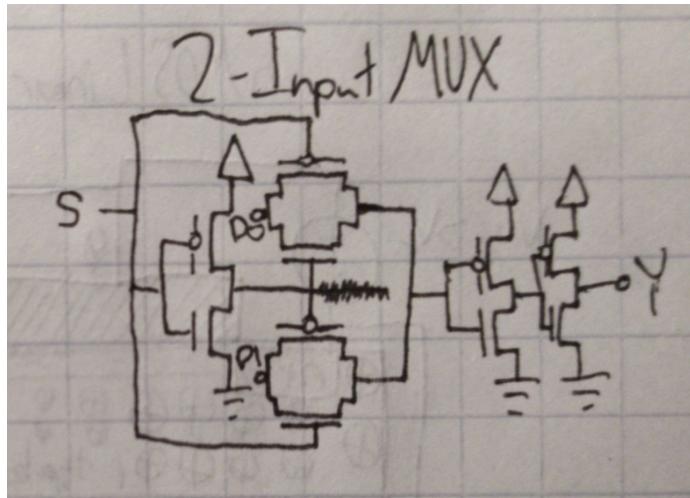
$$233.9 \times 70.95 = 16595.205$$

IV. Ring Oscillator (If used):

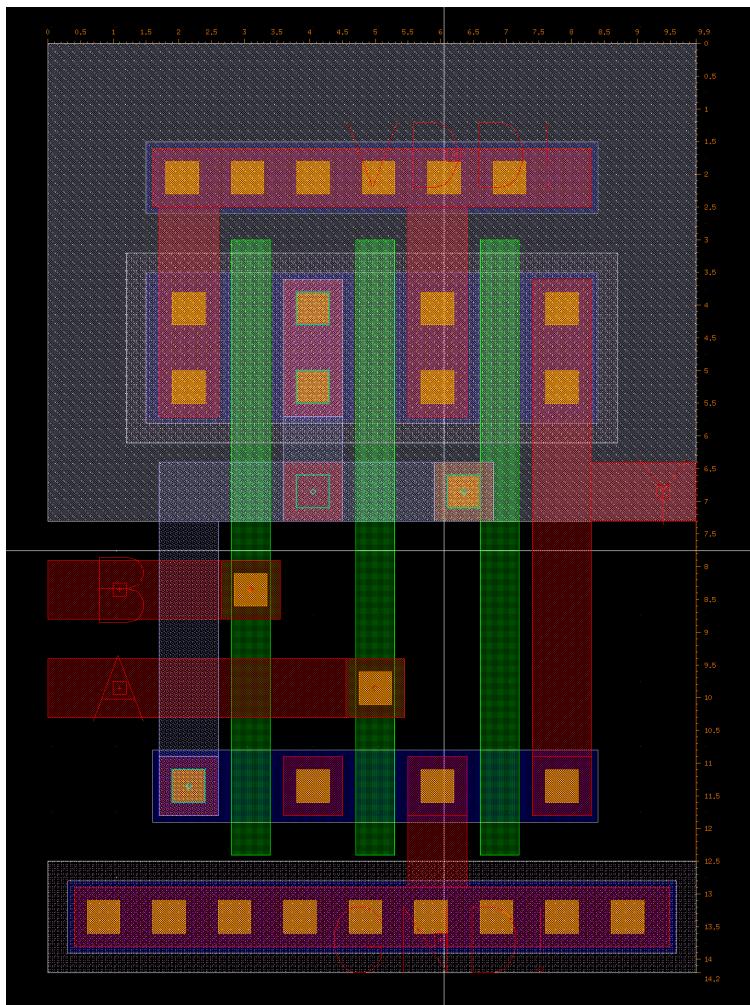
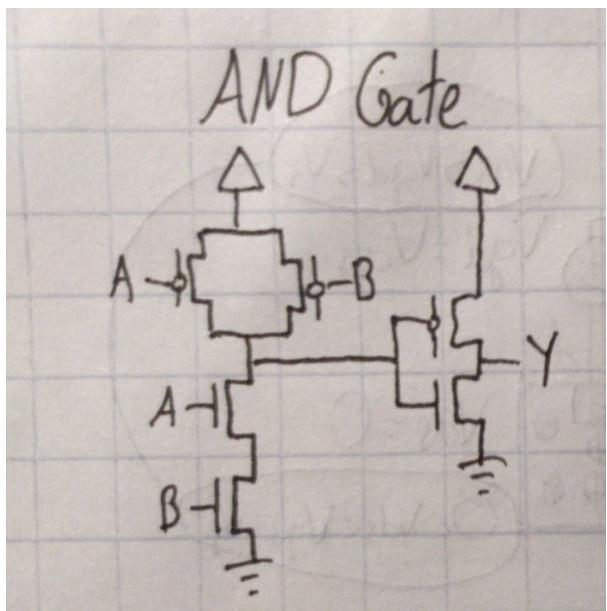


Currently works for a 7ns wavelength/delay. May need to be extended to around 15ns by doubling the number of inverters, but longer delay can be simulated by waiting 2 clock cycles instead of 1. Will try to use in the final design for extra credit if time allows.

## V. 2-input MUX



## VI. AND

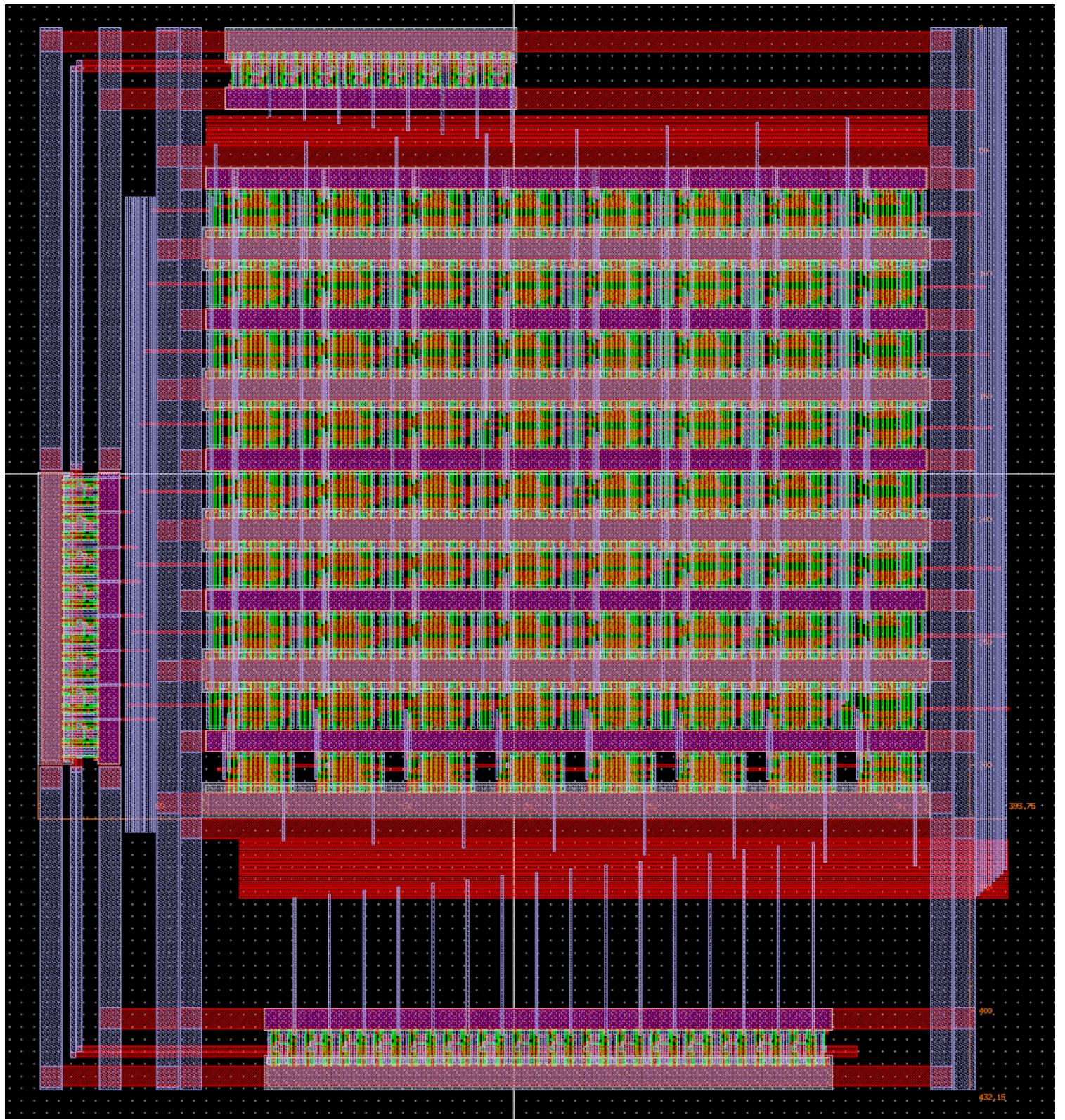


**(2)(15 pts.)** Show the layout of your multiplier with the registers (outside the padframe). Explain the design and functionality of your multiplier.

The design of our 8x8 multiplier is implemented through a ripple carry adder which connects full adders in series and has a carry bit that ripples and carries through the design. That way when you add multiple bits together we can keep track of the carry and if needed will carry over to the next ripple carry adder implemented. The 8-bit registers, which are our input, wait for 8 clock cycles until they are both full and represent an 8-bit number, which then is propagated into the combinational logic of the full adders connected in the middle. Those ripple-carry adders calculate the multiplication of both those numbers and then parallel out the number into the 16-bit shift register which is our output in the end. Finally, the parallel out is controlled by 16 multiplexers, which allow a selector bit to choose when we want to parallel in or serial out the results we have received.

Layout is on the next page. 8-bit shift registers and 16-bit shift registers are in the layout but are currently not connected. They have wires going to (but are currently not connected to) buses in M1/M2 which are currently test inputs for A7-A0 and B7-B0 in place of the two 8-bit SRs and C15-C0 in place of the 16-bit SR.

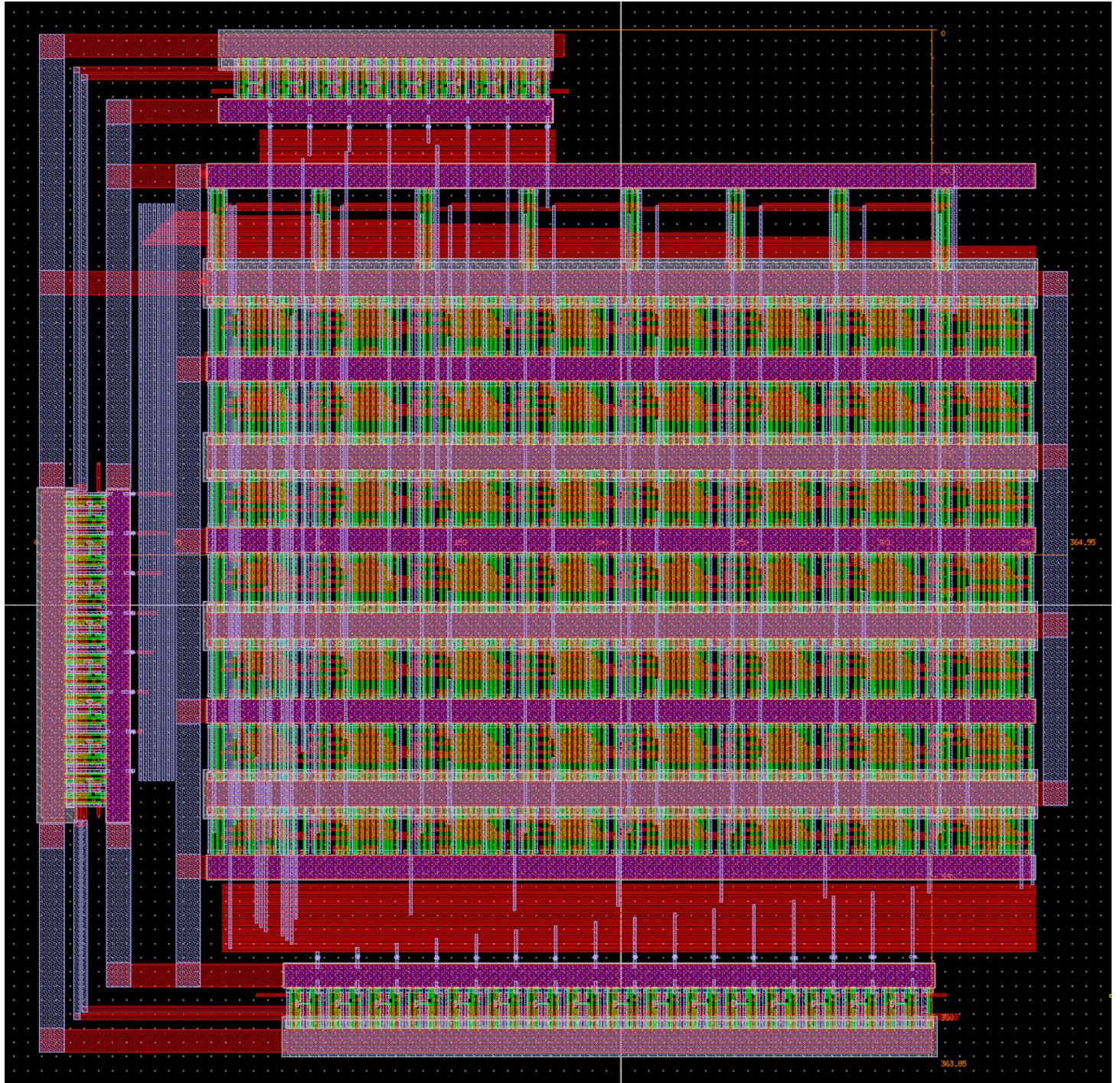
New working design (8 rows of AND+Adders + 1 row of Adders going from right to left)



### Old design (1 row of AND gates, 7 rows of AND+Adders)

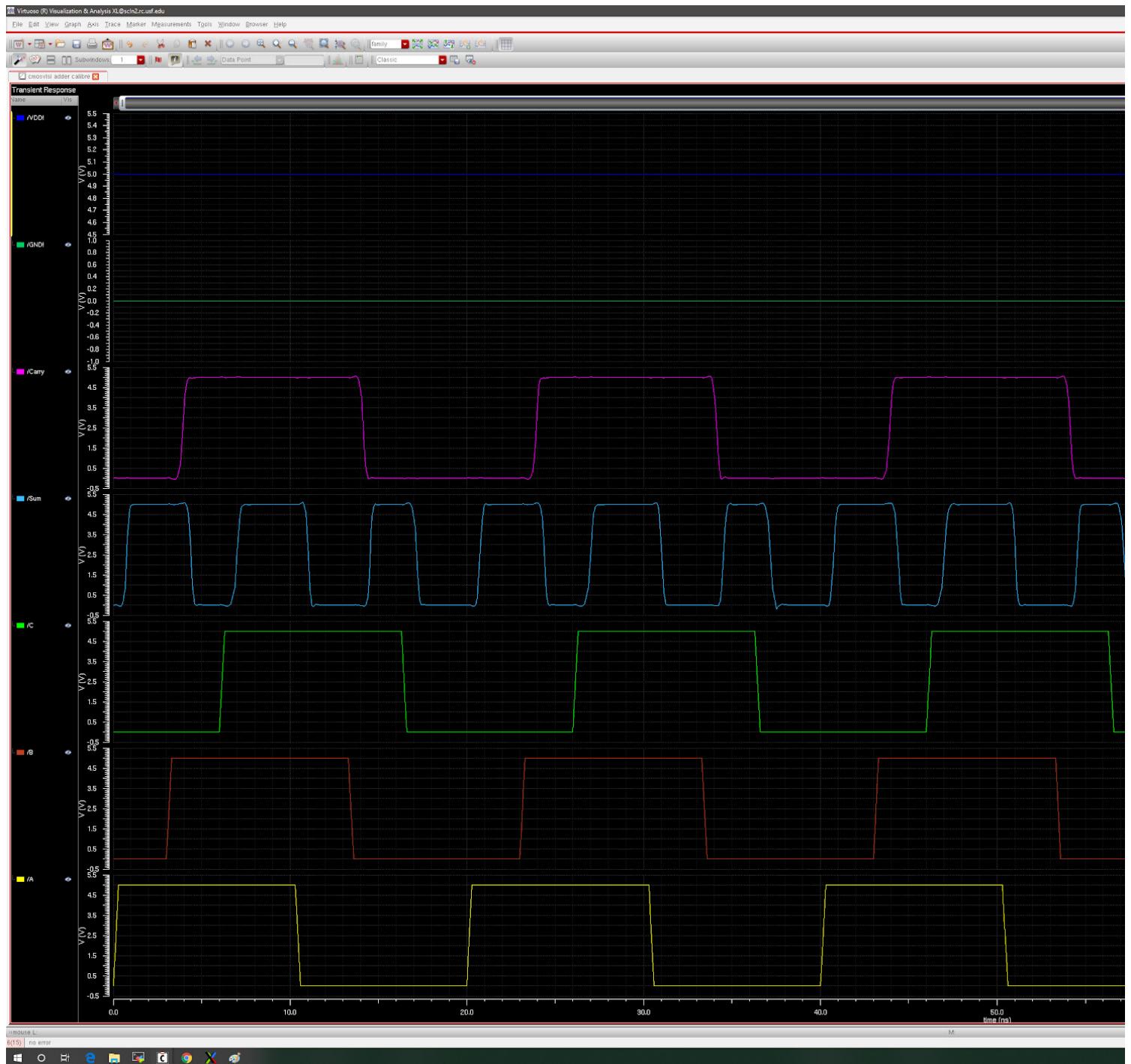
- Ended up not working at first since the 8-bit SRs were wired up to VDD and GND, but also the test lines. 8-bitSRs output 0 when the test lines (buses in M1/M2) input the needed values, which ended up causing issues for the final waveforms. Was mostly fixed so we have two working 8x8 multiplier designs

(3)(25pts) Simulation Results (without padframe):

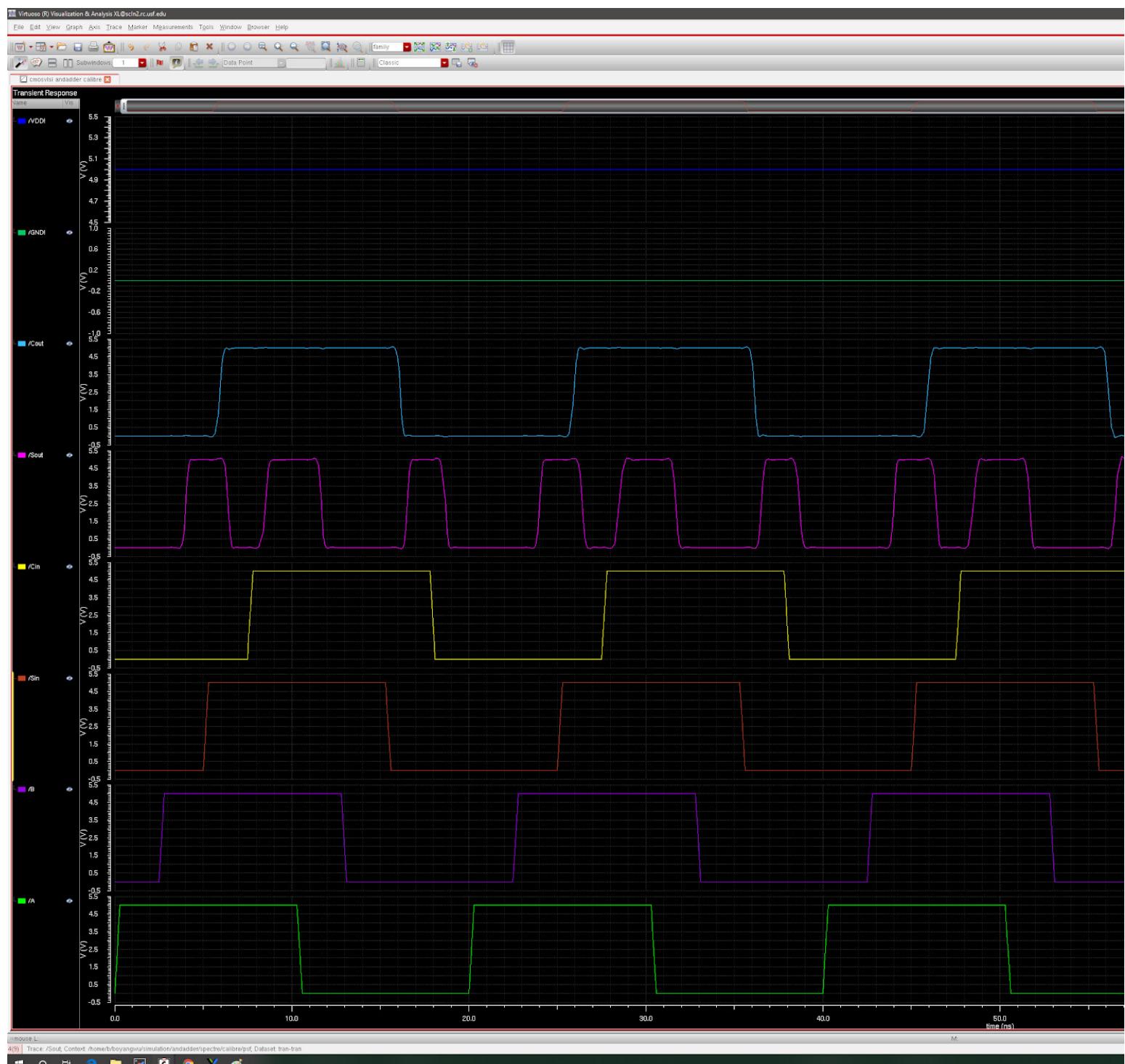


(a) (5 pts total) Individual cells:

### I. Full Adder:



## II. Full Adder with AND gate:

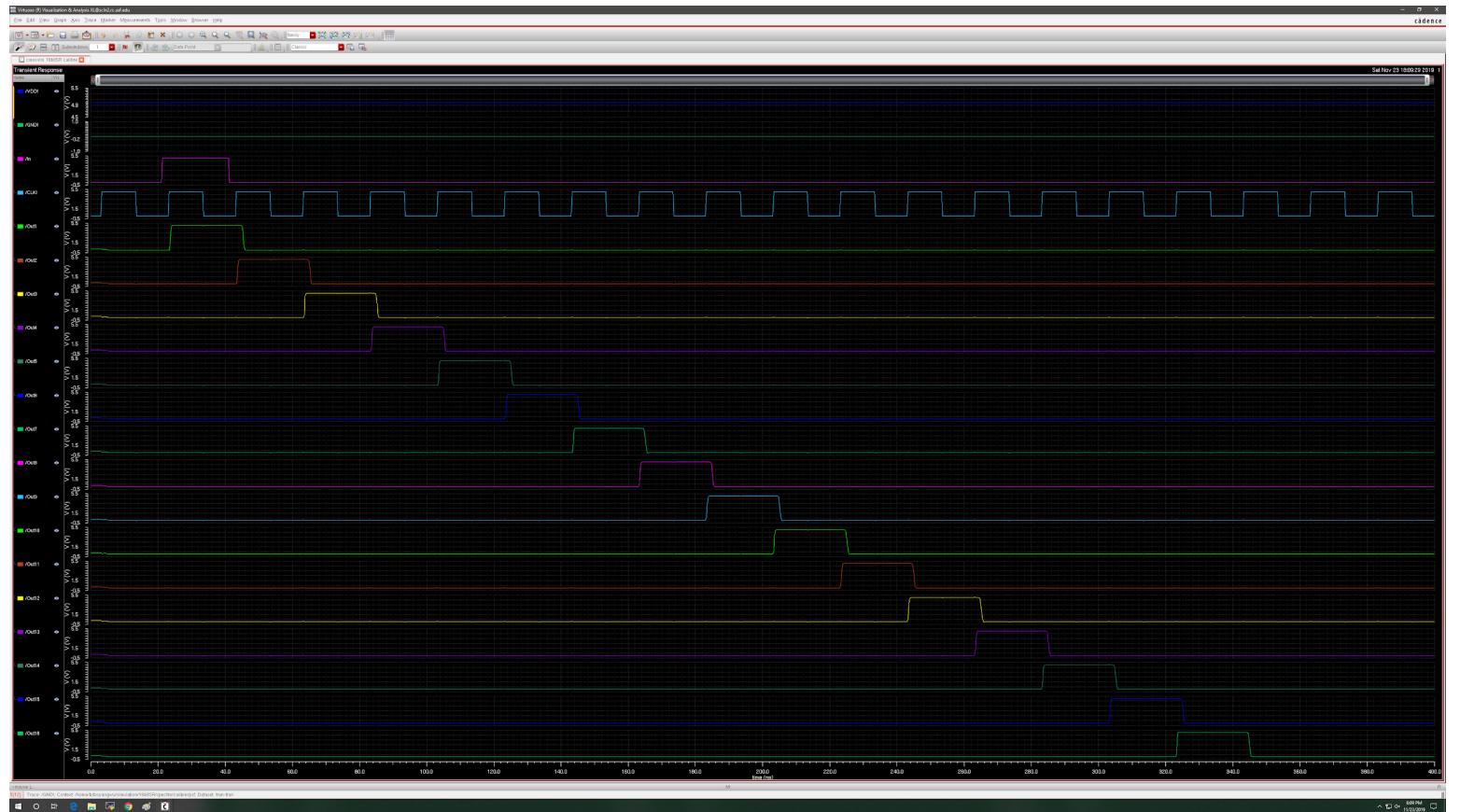


### III. Registers (Test Mode):

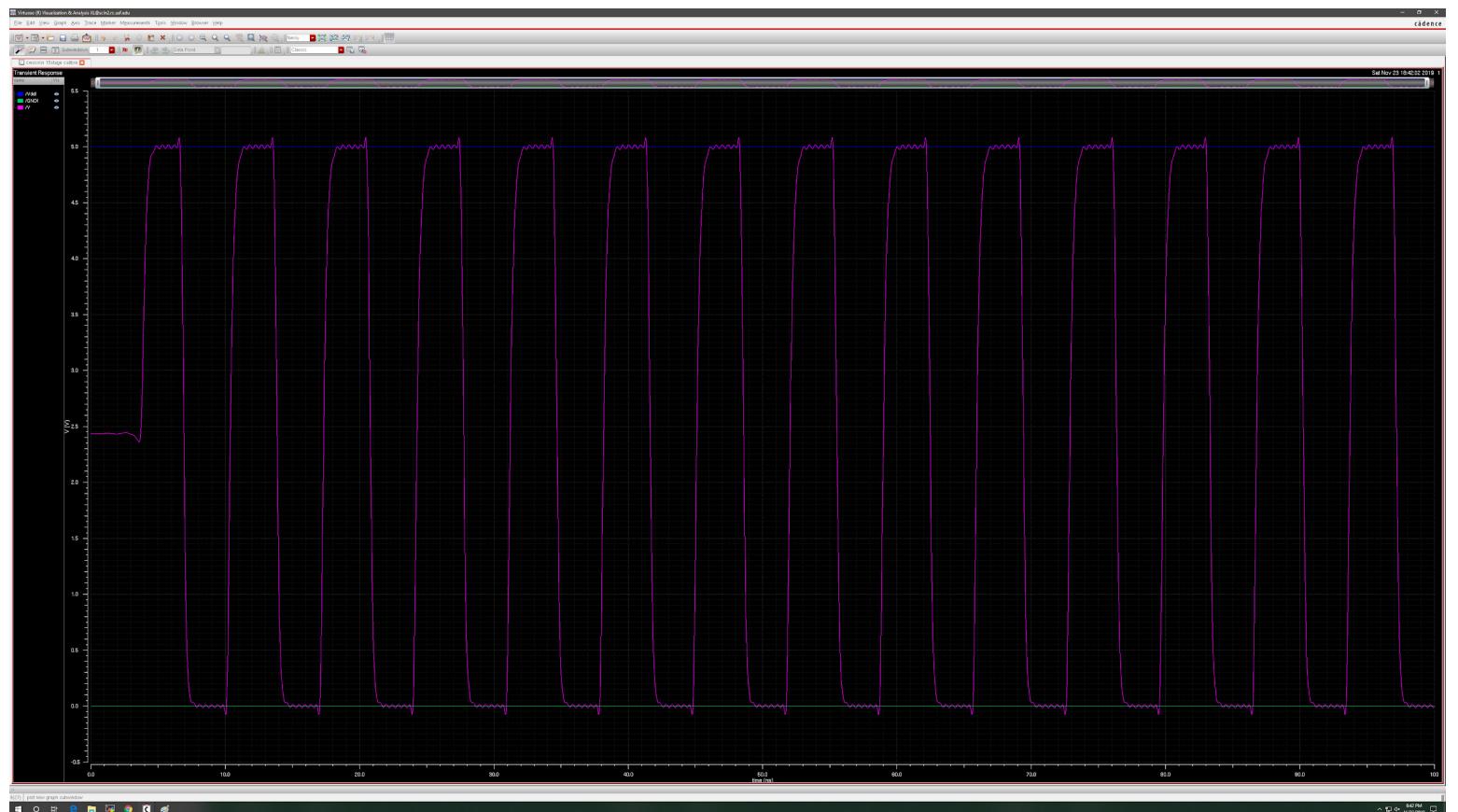
8-bit shift register



## 16-bit shift register



#### IV. Ring Oscillator (If used):



7ns delay

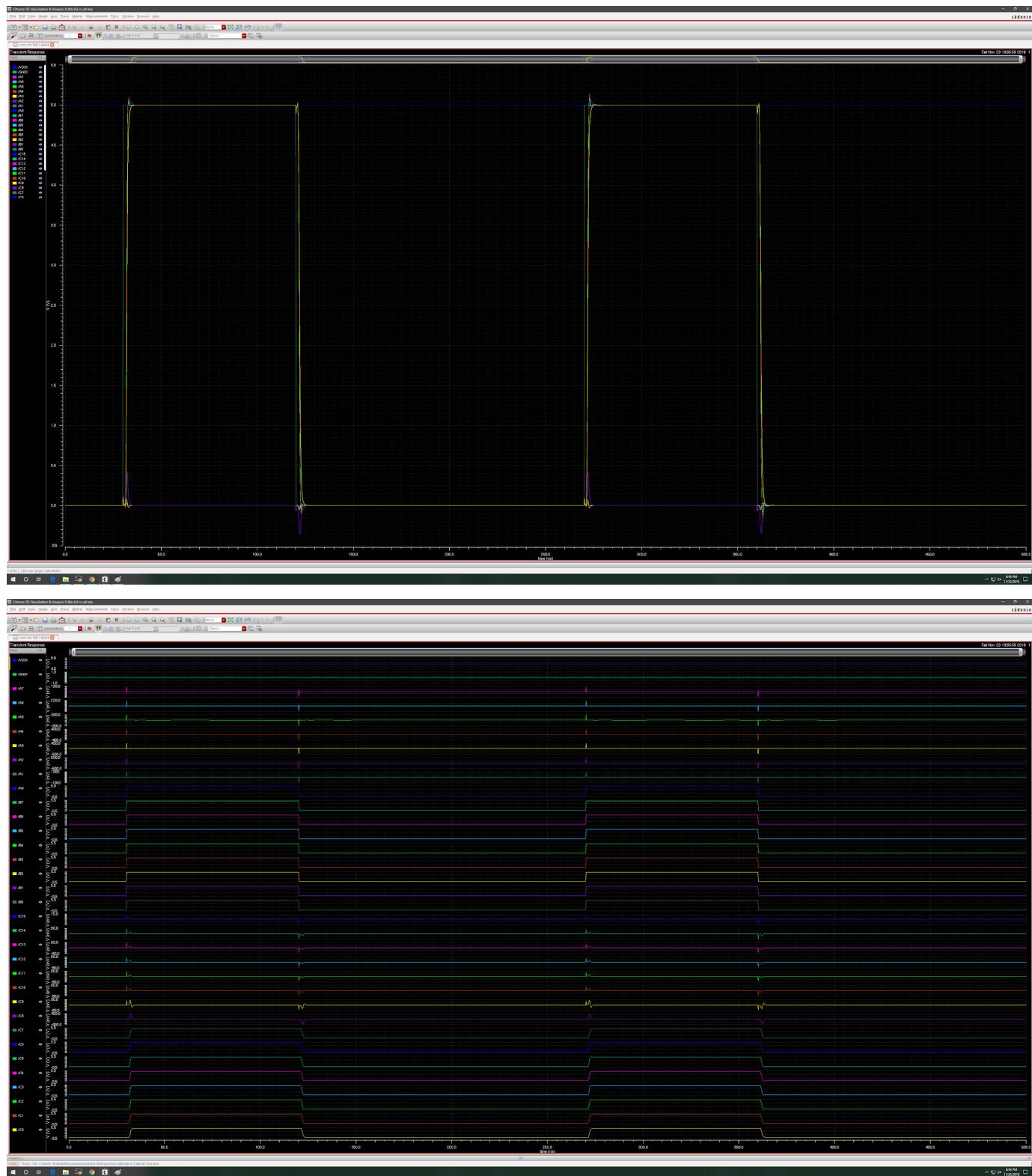
(b) (20 pts) The final multiplier:

- Parallel in/out are not currently working so look at the waveforms between 40ns to 115ns for correct input.
- There is incorrect input of around 10ns or so (critical path delay) right when inputs switch from 0 to 1 (around 30ns) which will be accounted for in the final design.
- The separated waveforms will also look bad since the Y-axis scales are usually in the mV or less, so we will show waveforms of the combined (to show strong 0's (0V) and 1's (5V)) as well as the separated (to see each individual input/output line). Apologies in advance since it may be harder to read. Zoom in since the original screenshots should be fairly detailed but images imported to document size would be smaller.
- There are also some weird things happening to zeroed input values but they are all in the aV range, so everything is still strong 0's and 1's.

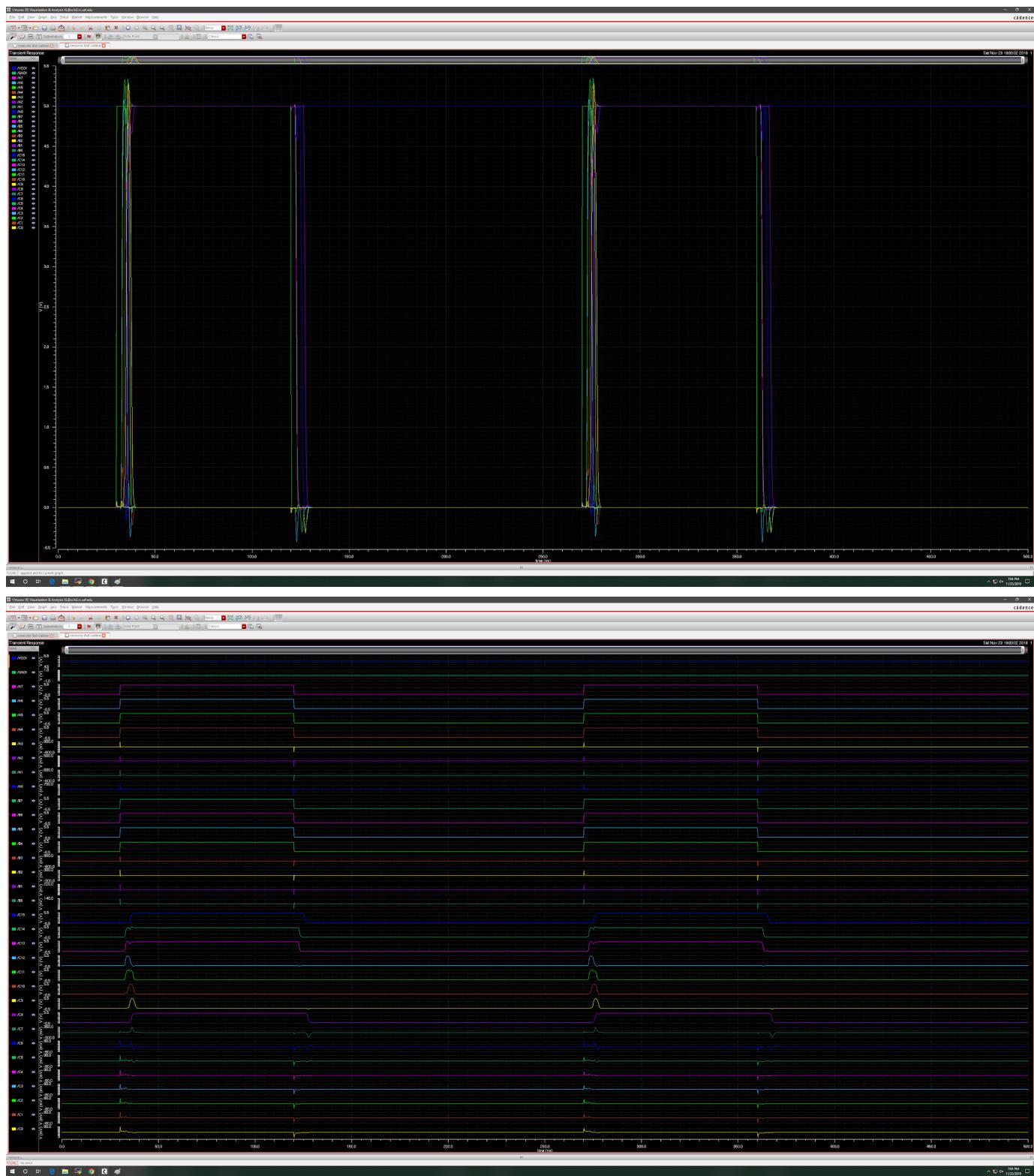
$$00000000 \times 00000000 = 0000000000000000 (0 \times 0 = 0)$$



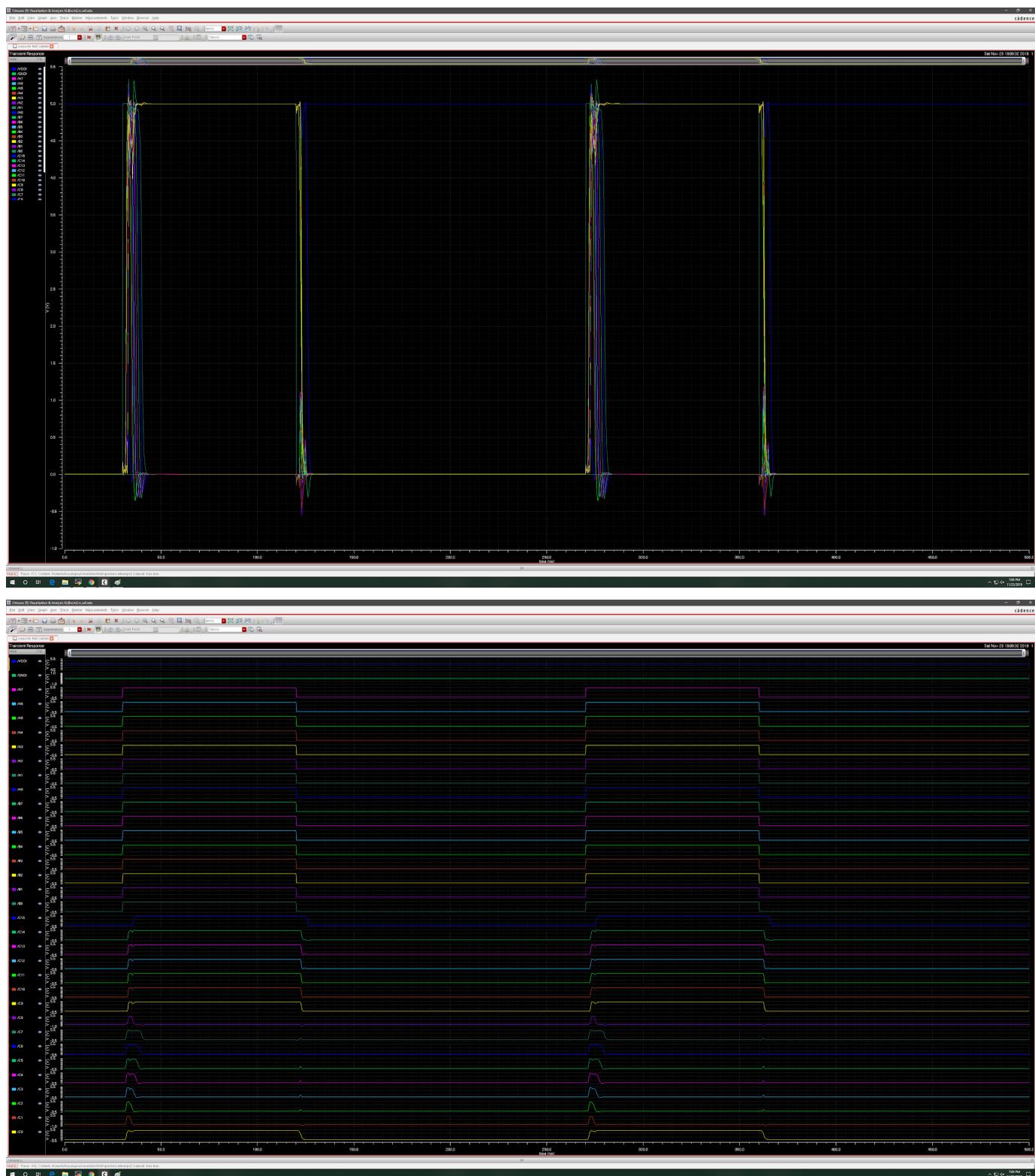
$00000001 \times 11111111 = 0000000011111111$  ( $1 \times 255 = 255$ )



$$11110000 \times 11110000 = 1110000100000000 \quad (240 \times 240 = 57600)$$



$$11111111 \times 11111111 = 1111111000000001 \quad (255 * 255 = 65025)$$



**(4)(10 pts.)** Layout of the final design (with padframe):

**N/A. Not required for partial design on 11/24.**

**(5)(20 pts)** Simulation waveforms for the final design (with padframe):

**N/A. Not required for partial design on 11/24.**