ECE 467 - Introduction to VLSI Design
Spring 2025
Project Report #1

Group members:
Michael Stanley
Dominik Kornak
Andrew Seong
Beyinah Alrashdan
George Anderson

Group member contributions

Michael Stanley: Contributed a little to cell library schematics. Created symbols for cell library gates. Created the FA, 8-bit RCA, and ran adder simulations. Created SRAM, ran some multiplier simulations.

Dominik Kornak: Contributed to cell library schematics. Ran simulations for cell library. Created flip flops and registers, ran FF simulations. Created decoders.

Andrew Seong: Contributed to cell library schematics. Created SRAM cell.

Beyinah Alrashdan: Calculations for inverter chain. Helped with multiplier.

George Anderson: Sized and created every inverter schematic and symbol, ran all inverter simulations and recorded all seen delays. Created multiplier and ran multiplier simulations.

Part 1: Inverter Chain Results

Introduction:

Inverter chains are important in electronic circuits to strengthen signals and control delays. They help ensure that signals stay clear and strong, especially when they need to travel long distances or drive larger loads. By carefully designing the number and size of inverters in the chain, engineers can balance power consumption, delay, and signal strength for efficient circuit operation.

Task: To create an inverter chain to drive a load of 1.28 pF with a minimum t_p delay.

Using simulation, we found the value of C_1 to be $0.175 \cdot 10^{-15}$ F. We find an approximation for N_{opt} as follows:

$$N = ln(^{C_L}/_{C_1}) = ln(^{1.28 \cdot 10^{-12}}/_{0.175 \cdot 10^{-15}}) \approx 8.9 \implies N_{opt} = 8 \tag{1}$$

$$u_{opt} = e^{ln(C_L/C_1)/N} \tag{2}$$

So, we will simulate inverter chains with 1-8 stages to empirically determine which N is optimal.

We created the inverter chains by first using a unit inverter with the PMOS having a width W of 180 nm and the NMOS having the W of 90 nm. We then sized each following inverter using equation (2). Below are the results of the simulated t_p delays for each inverter chain.

N	u	t_p [ns]
1	1	N/A
2	85.52	0.717
3	19.41	0.32995
4	9.25	0.2955
5	5.93	0.37645
6	4.41	0.5134
7	3.56	0.67585
8	3.04	0.872

Table 1: Inverter chain depth and associated delay

Below is a plot of the tabular data above:

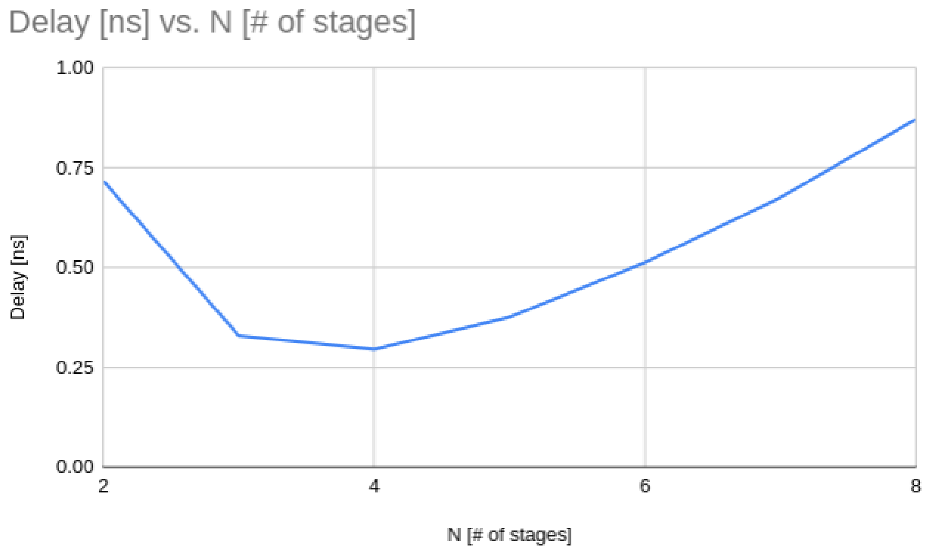


Figure 1: Inverter chain delay vs. N

As can be seen in table and figure 1 of the previous page, the inverter chain delay decreases from $N = 1$ until $N = 4$, and then increases towards $N = 8$. This tells us that $N = 4$ minimizes our delay. Thus, $N = 4$ is the optimum number of stages to drive a 1.28 pF load starting with our minimum sized inverter.
Below are plots of delay vs. temperature and supply voltage (V_{dd}) at the optimum number of stages:

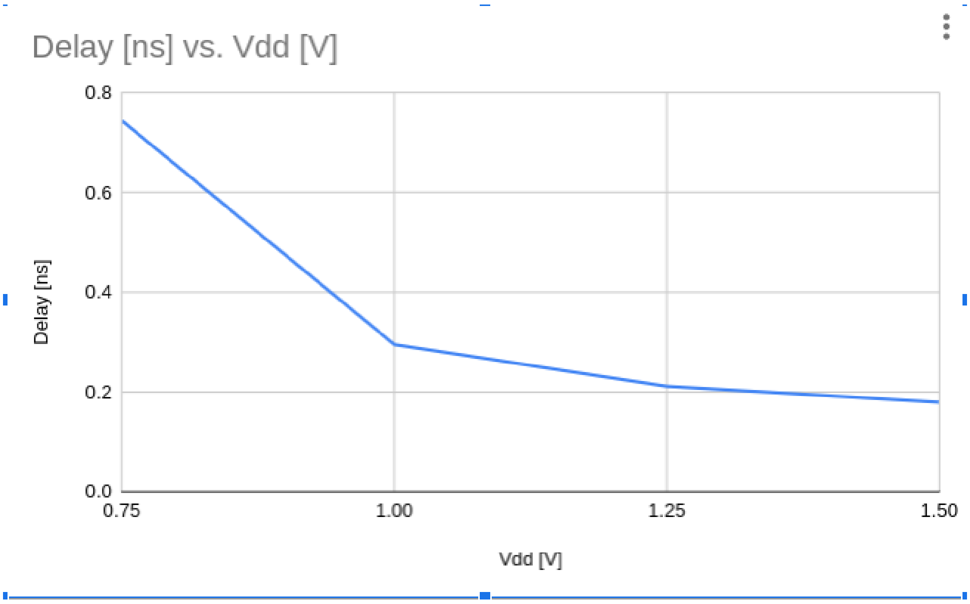


Figure 2: delay vs. V_{dd} @ N=4

In figure 2, we see that the delay decreases with increased supply voltage.

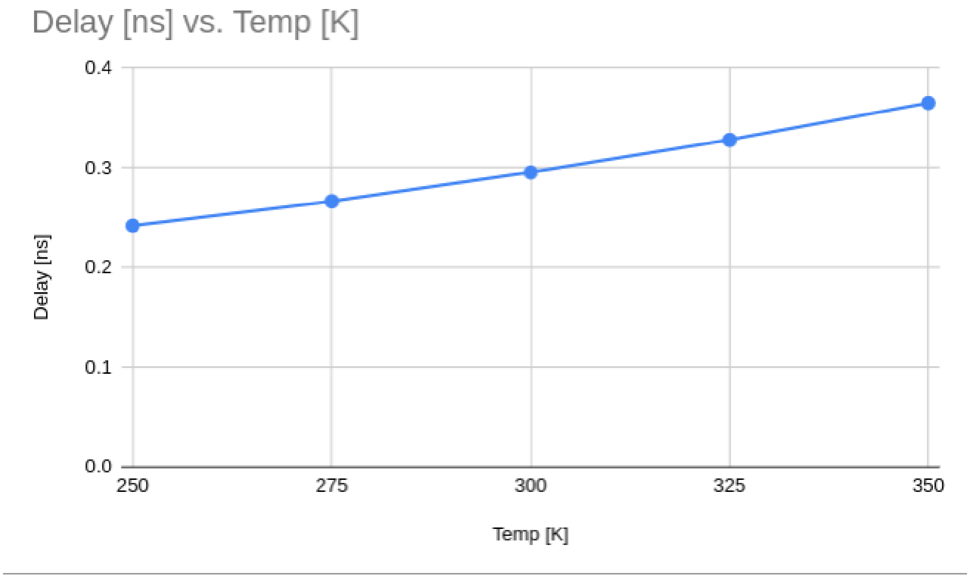


Figure 3: delay vs. temp [K] @ N=4

In figure 3, we see that the delay increases with increased temperature.

Below are our complete inverter chain schematic and symbols:

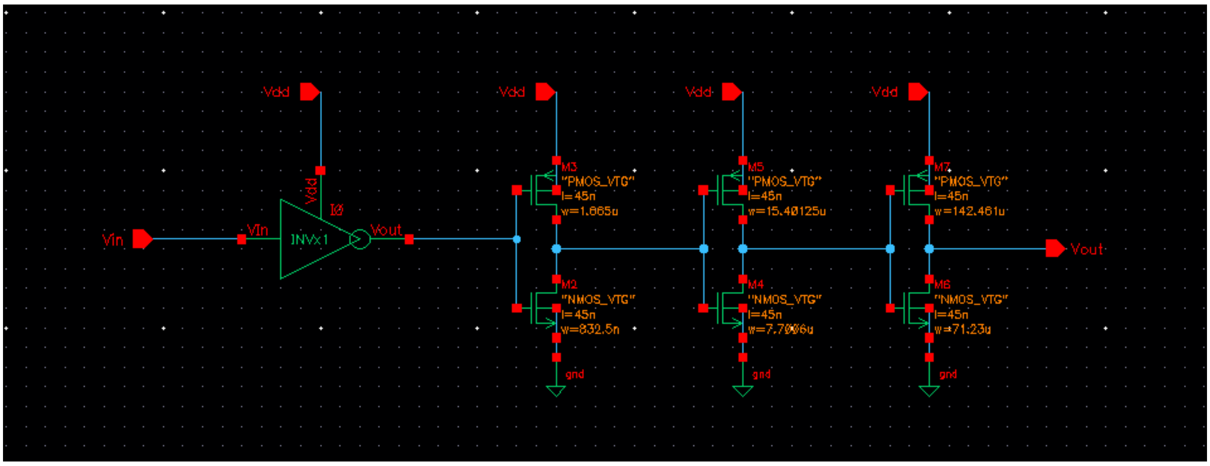


Figure 4: Inverter chain schematic (N=4)

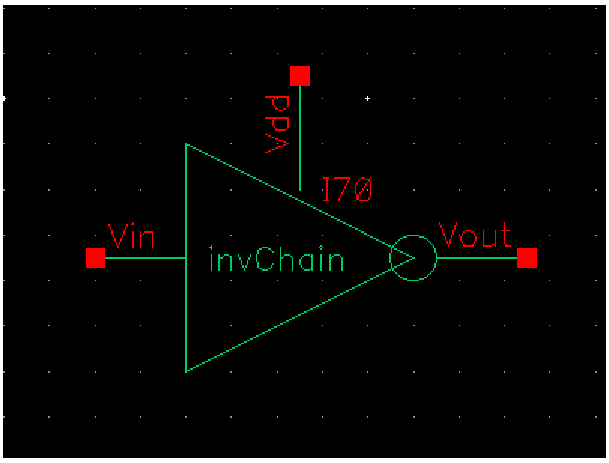


Figure 5: Inverter chain symbol (N=4)

Simulations:

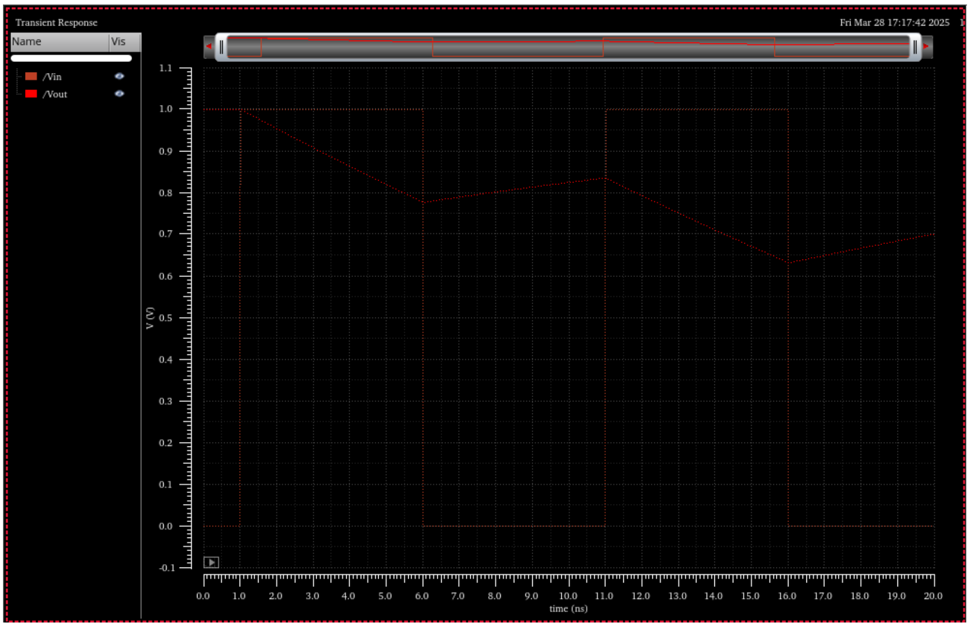


Figure 6: Inverter chain delay (N=1)

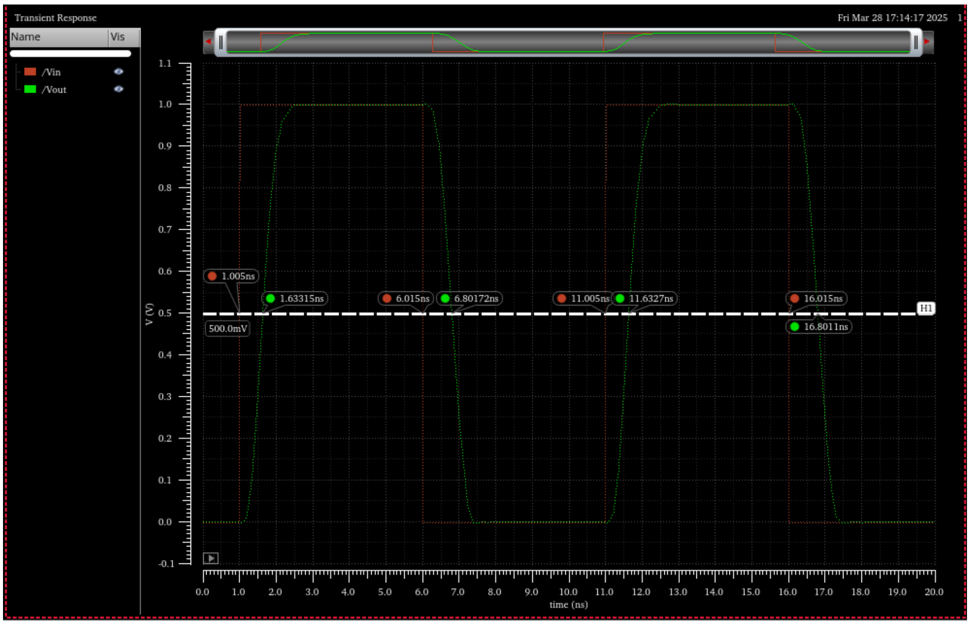


Figure 7: Inverter chain delay (N=2)

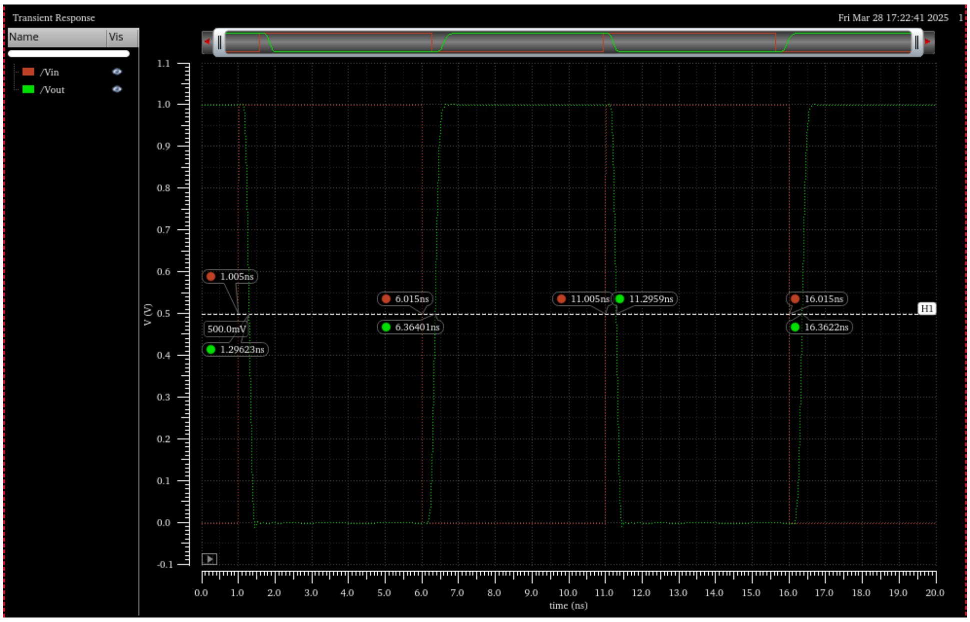


Figure 8: Inverter chain delay (N=3)

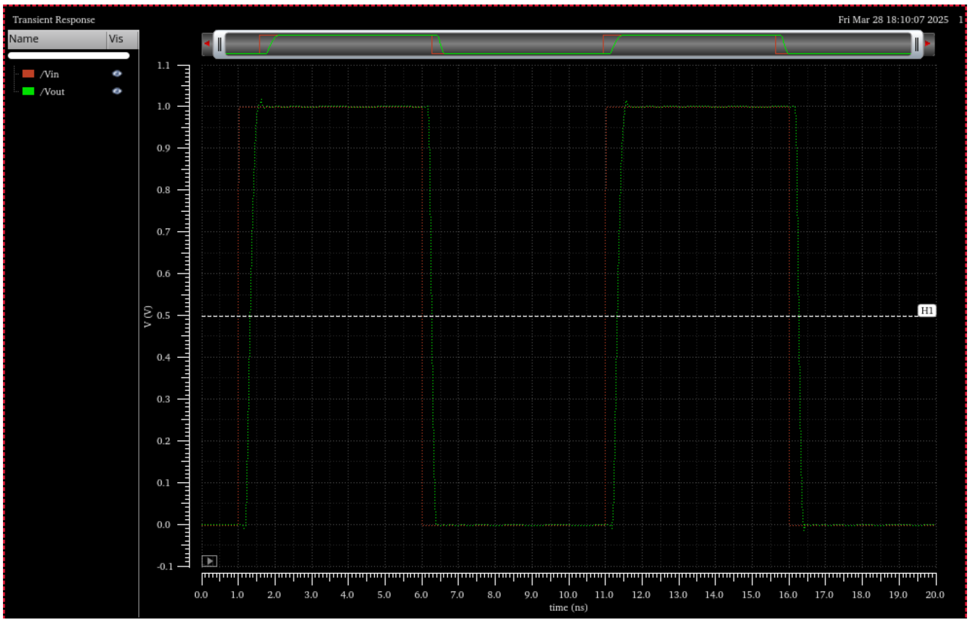


Figure 9: Inverter chain delay (N=4)

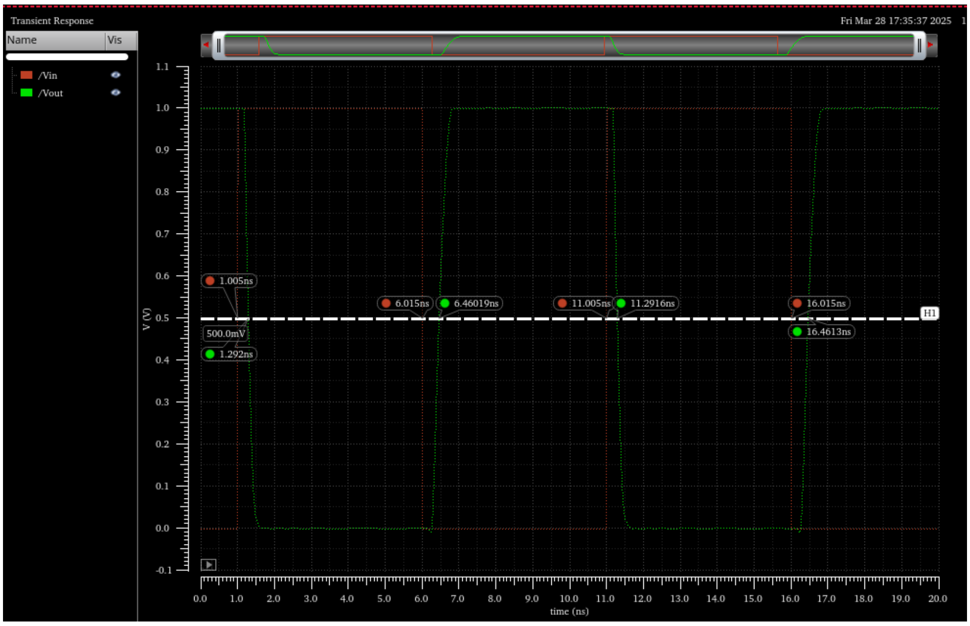


Figure 10: Inverter chain delay (N=5)

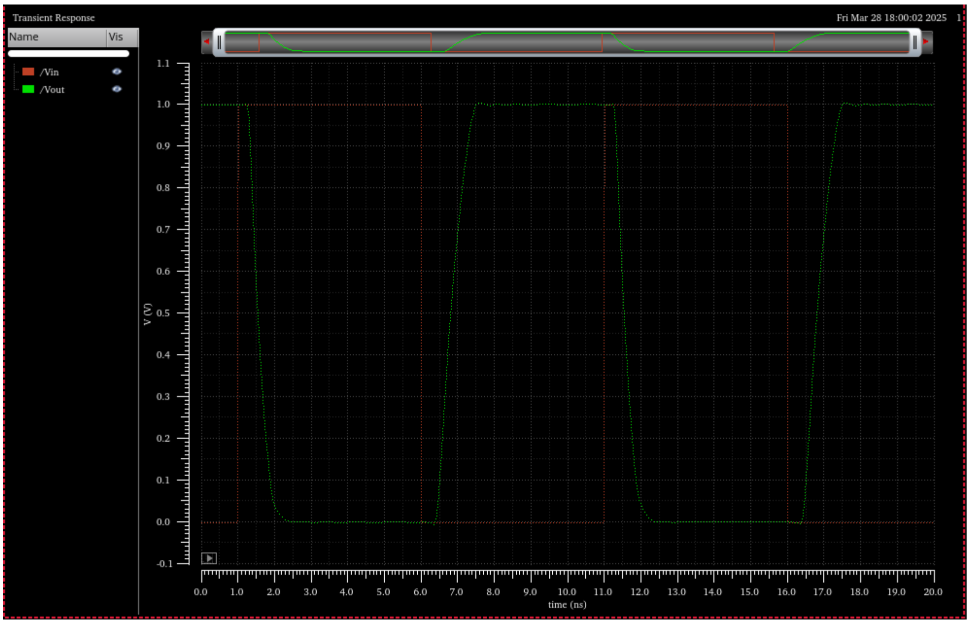


Figure 11: Inverter chain delay (N=6)

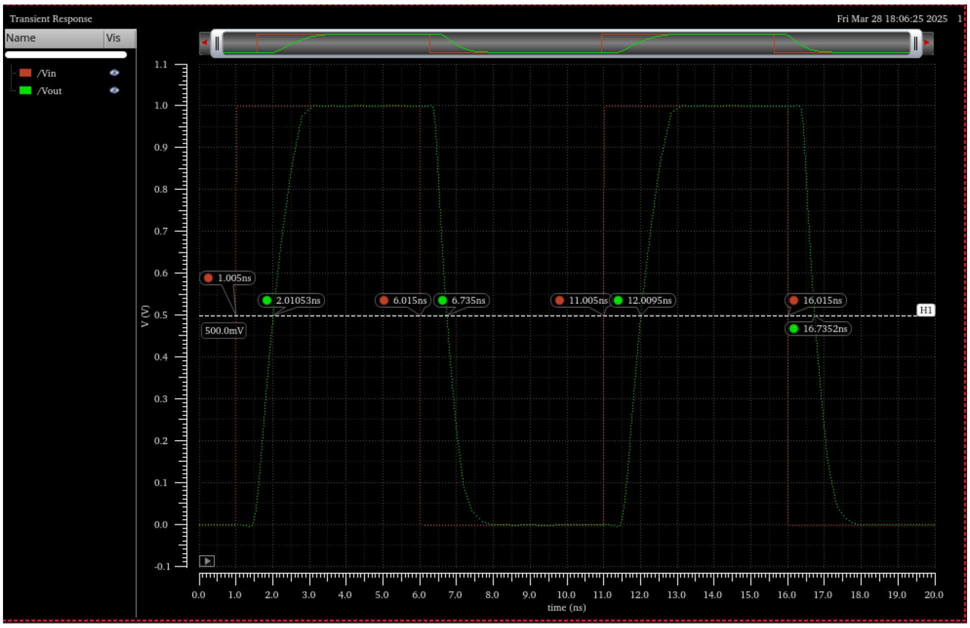


Figure 12: Inverter chain delay (N=8)

Conclusion

In part 1, we studied how the number of inverters in a chain affects signal delay and rise time. We found that the propagation delay, t_p, was smallest at N=4. This pattern makes sense because adding more inverters helps at first but eventually slows things down.

We also saw that increasing the supply voltage decreases the delay, but the effect became smaller at higher voltages. On the other hand, higher temperatures made delay worse. These results show why choosing the right number of inverters is important for building efficient circuits.

Part 2: Cell library Results

Below are the worst case delay results we obtained for each gate of the cell library at $V_{dd} = 1\ V$ and $temperature = 300\ K$ driving no output capacitance:

Gate	Sizing	T_plh [ns]	T_phl [ns]	t_p [ns]
NAND2	x1	0.0071	0.01014	0.00862
	x2	0.0069	0.0103	0.0086
	x3	0.0069	0.0103	0.0086
NAND3	x1	0.0073	0.0151	0.0112
	x2	0.0072	0.0151	0.01115
	x3	0.0072	0.0151	0.01115
NOR2	x1	0.0225	0.0085	0.0155
	x2	0.0225	0.0083	0.0154
	x3	0.0225	0.0082	0.01535
NOR3	x1	0.0475	0.0118	0.02965
	x2	0.0475	0.0113	0.0294
	x3	0.0475	0.0112	0.02935
XOR2	x1	0.0305	0.0698	0.05015
	x2	0.0305	0.0782	0.05435
	x3	0.0308	0.0896	0.0602
XOR3	x1	0.152	0.182	0.167
	x2	0.2125	0.197	0.20475
	x3	0.266	0.155	0.2105
XNOR2	x1	0.0388	0.0167	0.02775
	x2	0.047	0.0166	0.0318
	x3	0.0555	0.0165	0.036
XNOR3	x1	0.1566	0.1314	0.144
	x2	0.176	0.1927	0.18435
	x3	0.1355	0.249	0.19225

Table 2: Worst case delays of gates from the cell library

Below are the symbols we created for our gates from the cell library:

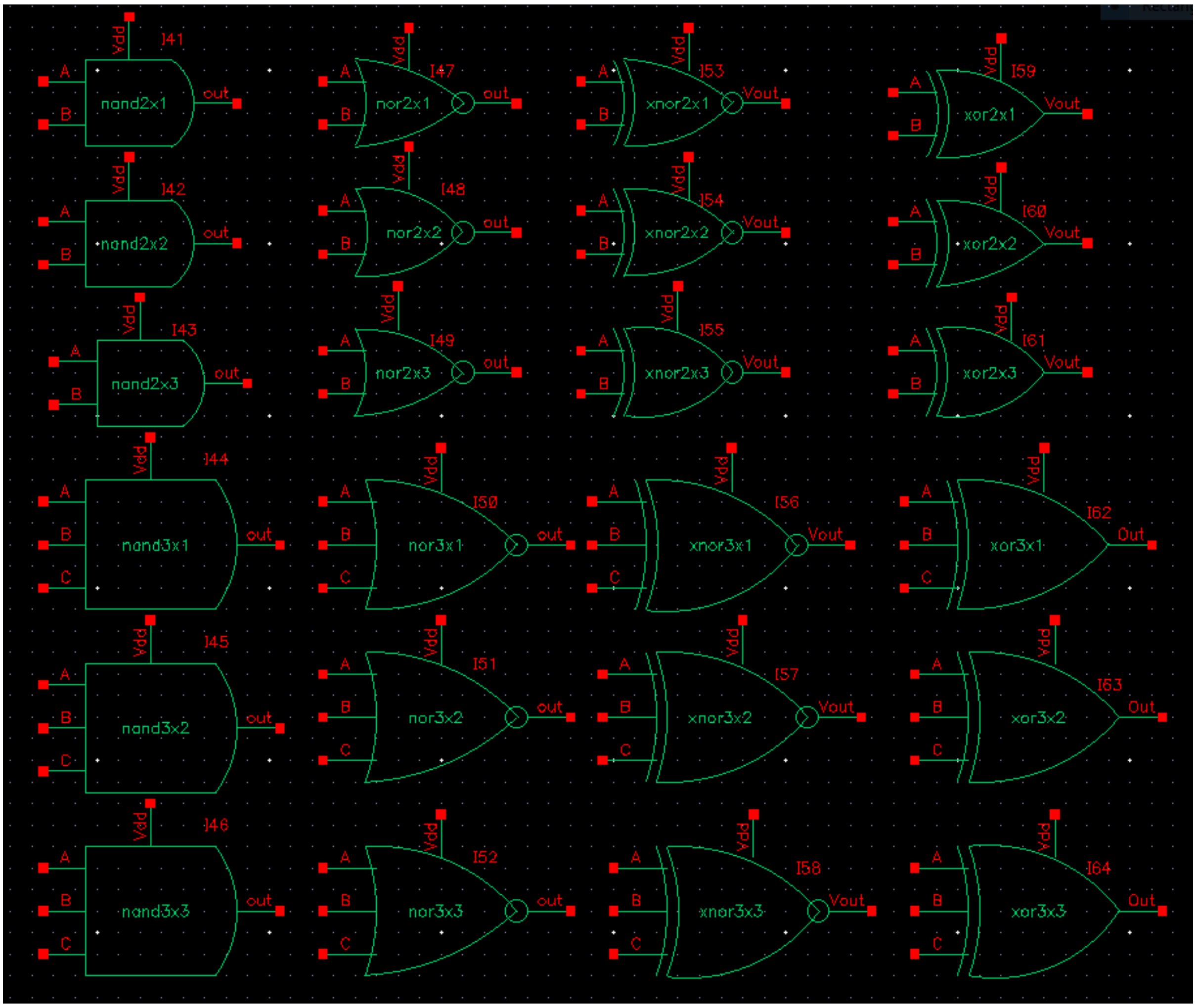


Figure 13: Cell library symbols

Part 3: 8-bit RCA

The microarchitecture of our 8-bit adder:

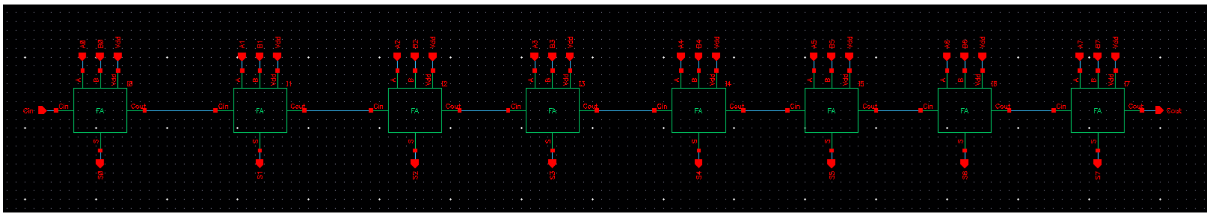


Figure 14: 8-bit RCA microarchitecture

The symbol for our 8-bit adder:

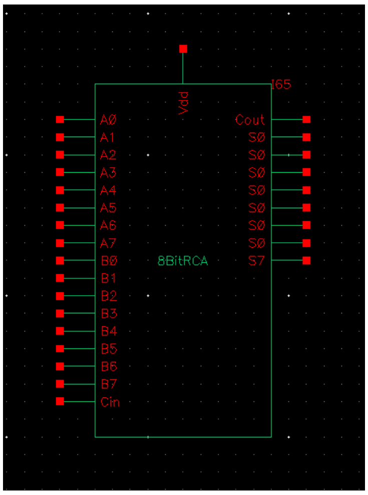


Figure 15: 8-bit RCA symbol

The circuit level schematic of our 1-bit full adder (FA):

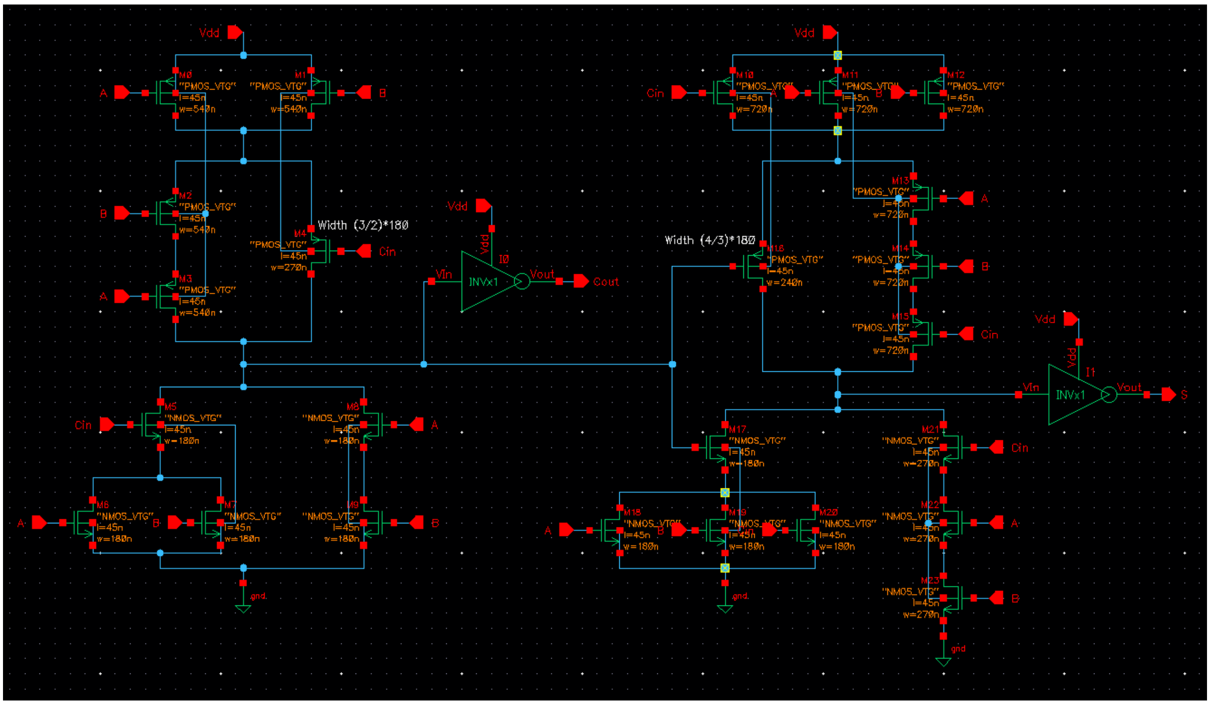


Figure 16: 1-bit FA circuit level schematic

Functionality of FA, demonstrated on every valid input combination (A, B, C_{in}):

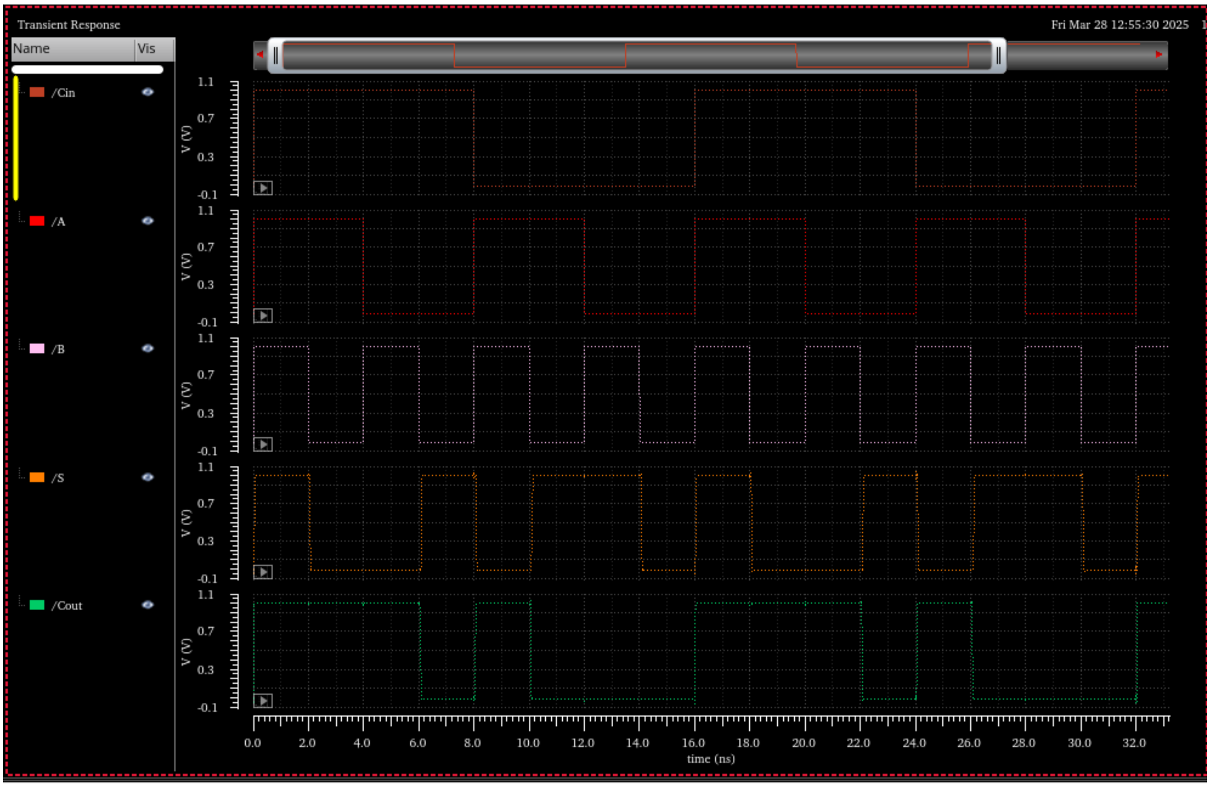


Figure 17: 1-bit FA functionality

Looking at the schematic of the FA in figure 16, we can determine the worst case input combination. Immediately, we see that the worst case delay will occur when the sum bit its pulled low/high by a change in the left-hand side network. The combination resulting in this delay is when $C_{in} = 1,\ A = 0$ and B transitions between 1/0.

When we simulated this, we got the following trace:

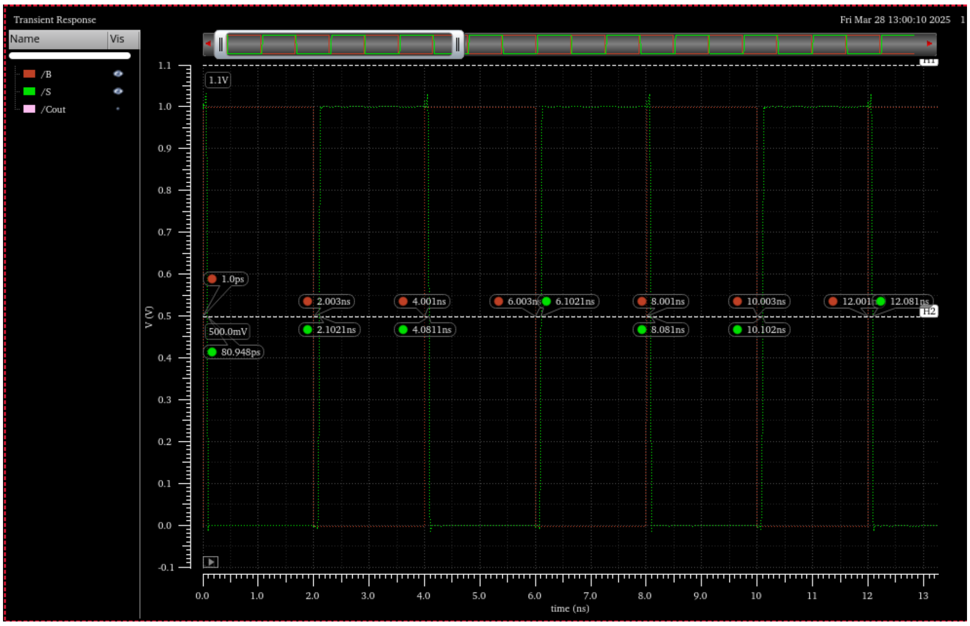
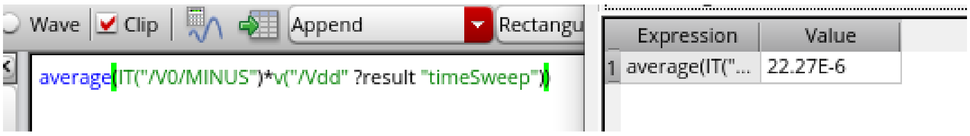


Figure 18: Worst case delay FA

Giving us a worst case delay of $t_p = 0.0919\ ns$ for the FA.

The average power consumed by the FA was found using the following calculation:



```
average(IT("/V0/MINUS")*v("/Vdd" ?result "timeSweep"))
```

	Expression	Value
1	average(IT("...	22.27E-6

Figure 19: Power consumption of FA

Which is 22.27 μW.

Below are plots of the worst case delay vs. V_{dd} and temperature.

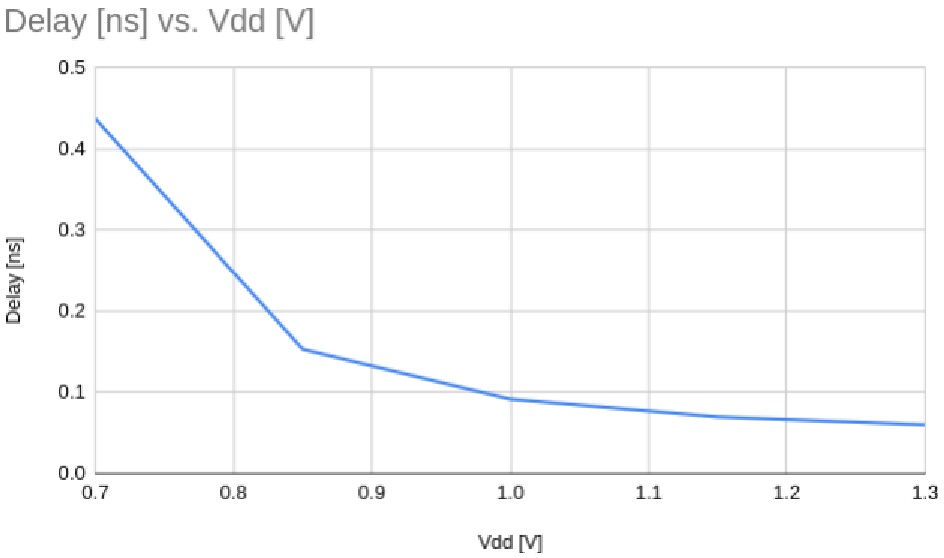


Figure 20: Worst case delay of FA vs. V_{dd} [V]

As we see above, delay decreases with increased supply voltage.

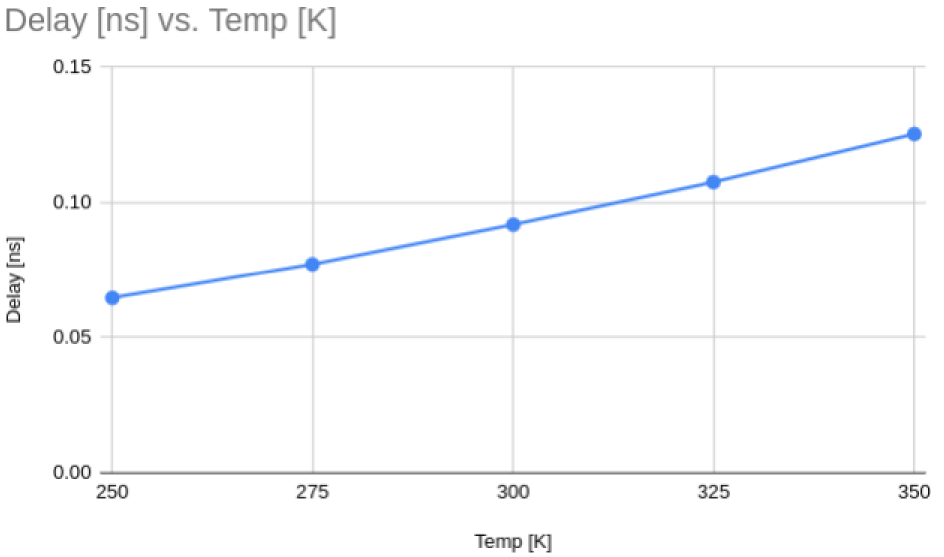


Figure 21: Worst case delay of FA vs. temperature [K]

We see an increase in delay as temperature rises.

Part 4: Multiplier

Microarchitecture of our 4-bit multiplier:

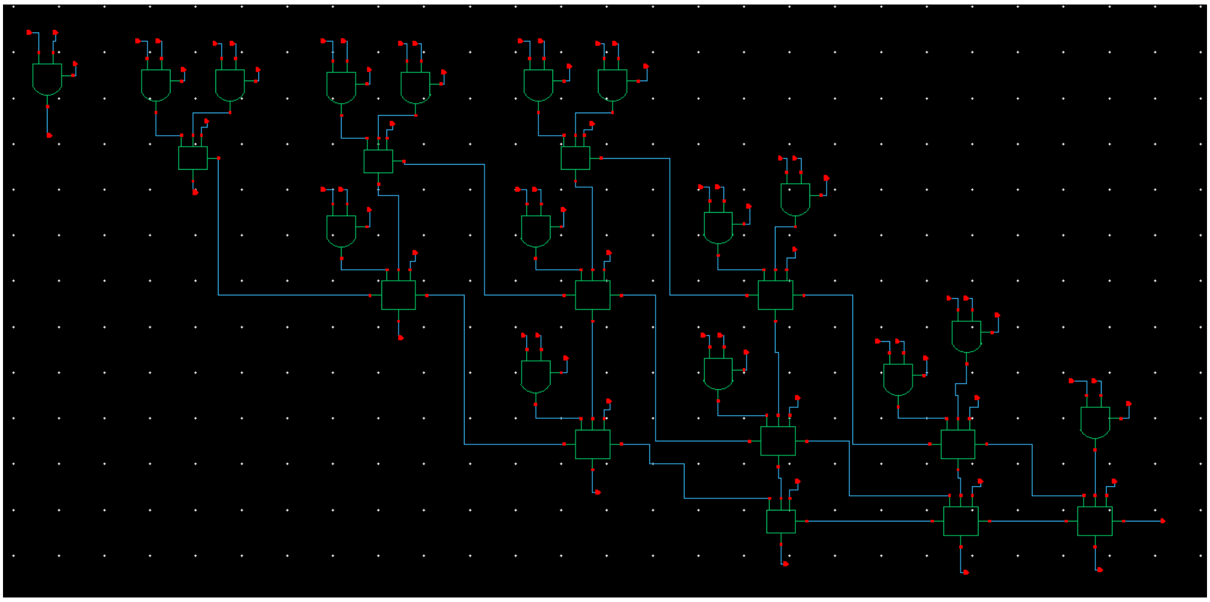


Figure 22: 4-bit Mult Microarchitecture

Below is a waveform confirming the functionality of our multiplier. We start with A=0011, B=0001. So the initial product is P=00000011. A remains constant for this test. Next, B=0011. P=00001001. Then, B=0111, P=00010101. Finally, B=1111, P=00101101.

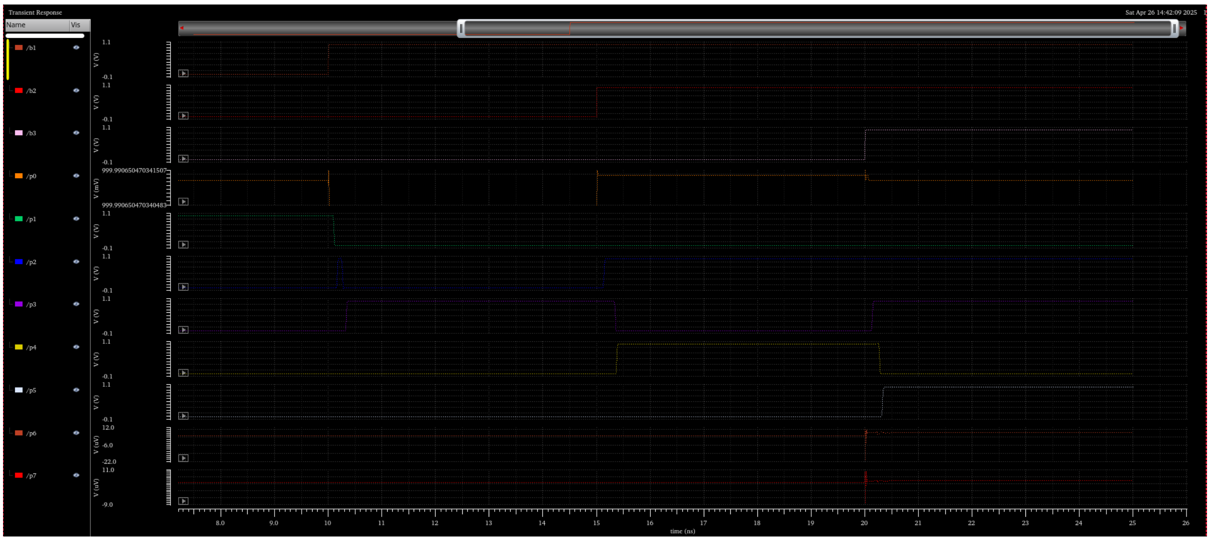


Figure 23: 4-bit Mult functionality

Maximum delay occurs when A0 and B1 both flip to 1 at the output P6 with a delay of .53083ns at 999.99 mV

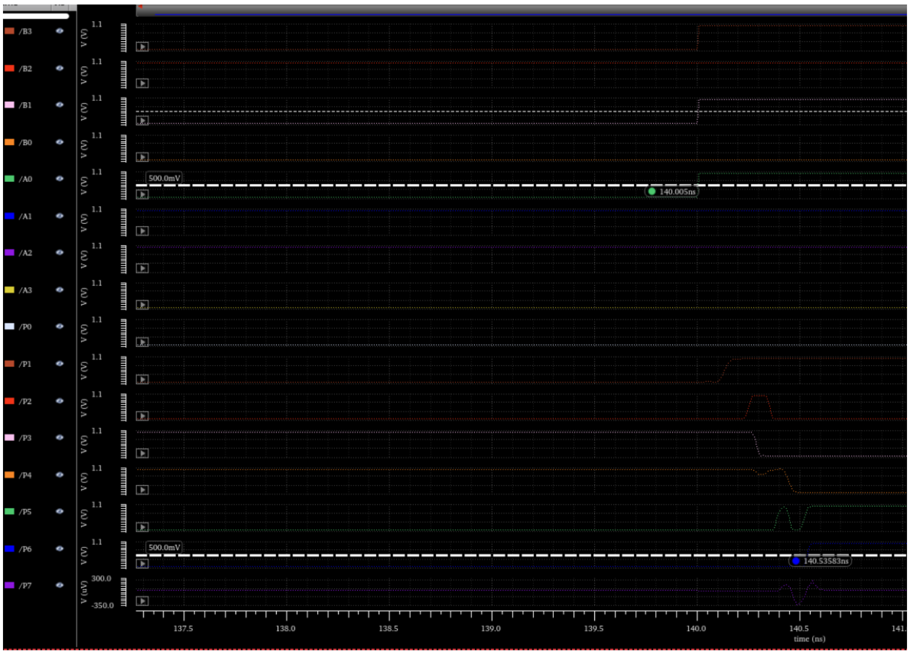


Figure 24: 4-bit Mult max delay

Below are plots of the worst case delay vs. V_{dd} and temperature.

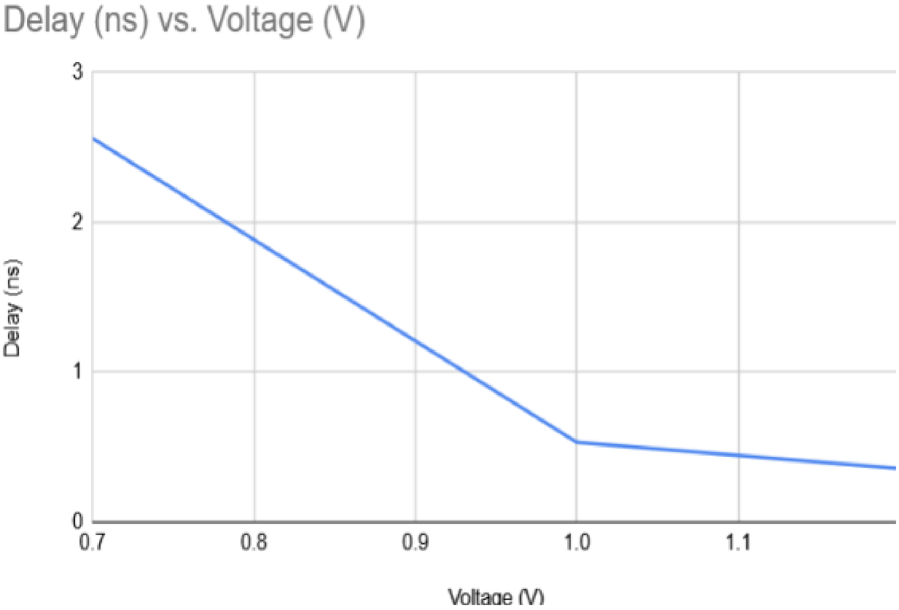


Figure 25: Worst case delay of 4-bit mult vs. V_{dd} [V]

As we see above, delay decreases with increased supply voltage.

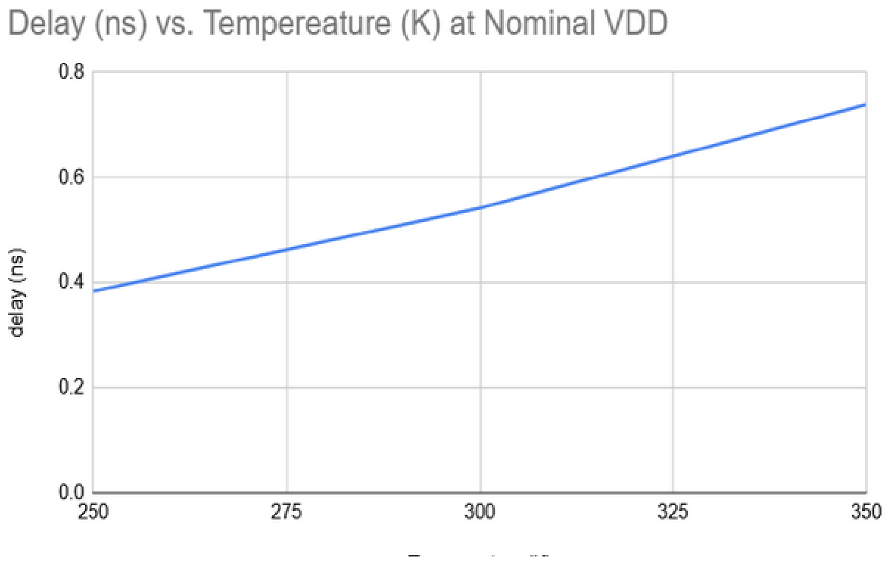


Figure 26: Worst case delay of 4-bit mult vs. temperature [K]

We see an increase in delay as temperature rises.

Part 4: F/F and Registers

Schematic of the FF architecture we choose to use:

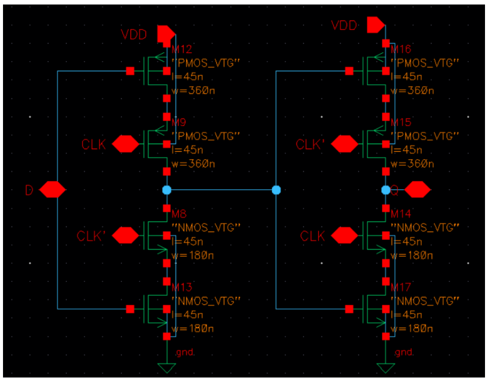


Figure 27: FF schematic

Setup and T_{c-q} delay:

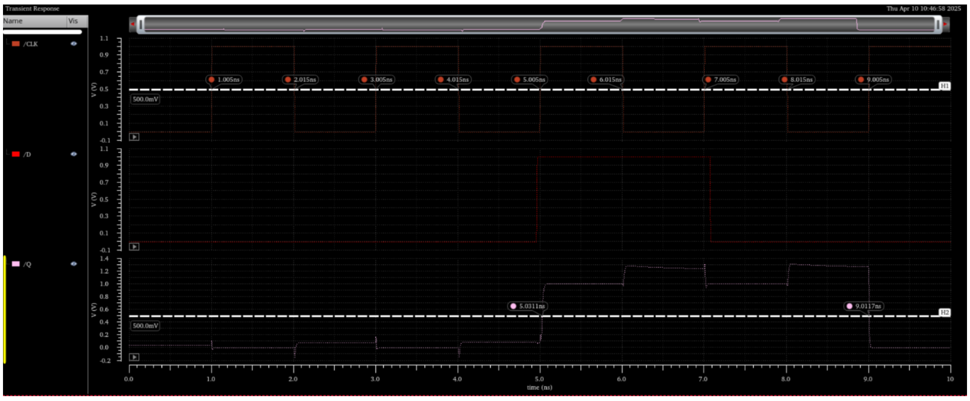


Figure 28: FF delays

$$T_{setup} = 0.4\ ns,\ 4\%\ \text{of CLK period}$$
$$T_{c-q} = 0.0261\ ns,\ 2.61\%\ \text{of CLK period}$$

T_{c-q} delay vs V_{dd}:

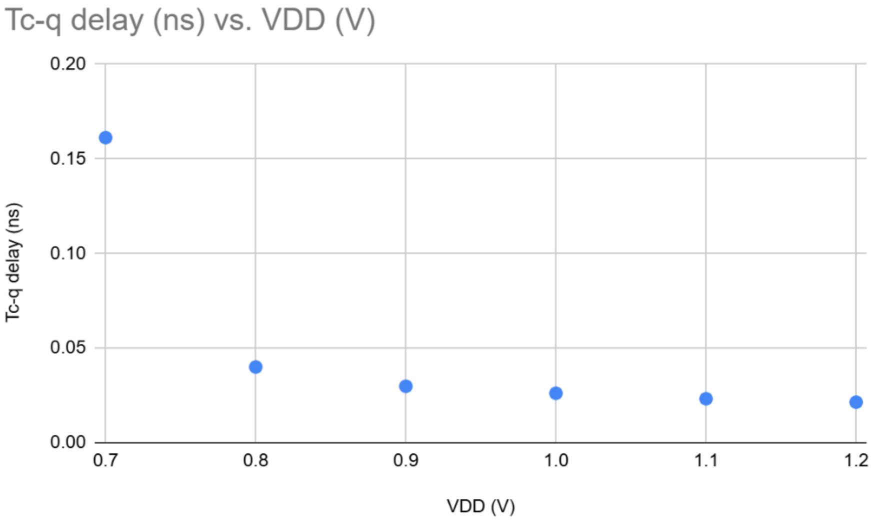


Figure 29: T_{c-q} delay vs V_{dd}

As seen in figure 29, delay decreases with increased V_{dd}.
T_{c-q} delay vs temperature:

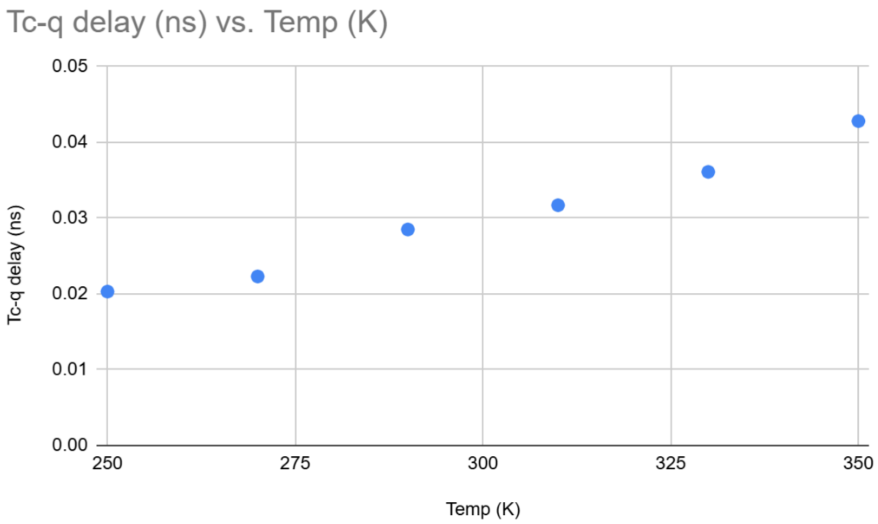


Figure 30: T_{c-q} delay vs temperature

We see that T_{c-q} delay increases with temperature.
FF power consumption vs V_{dd}:

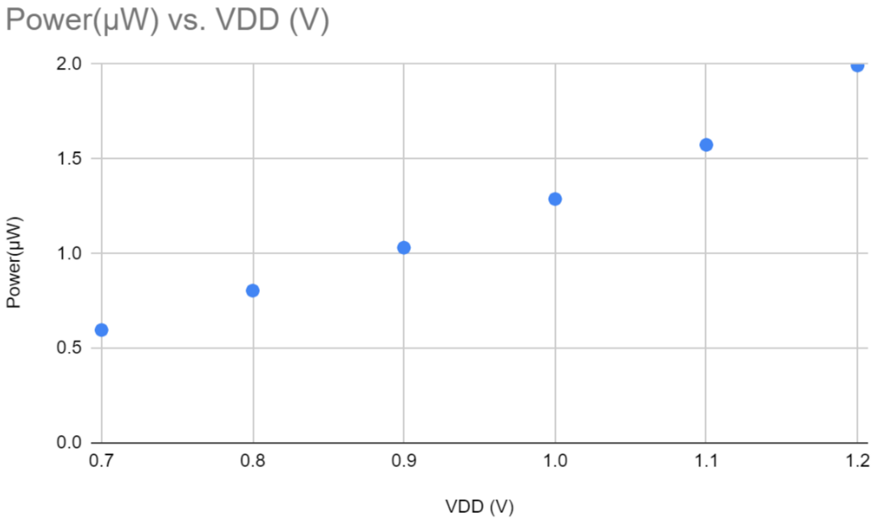


Figure 31: FF power consumption vs V_{dd}

As V_{dd} increases, so does power consumption as is expected.

FF power consumption vs temperature:

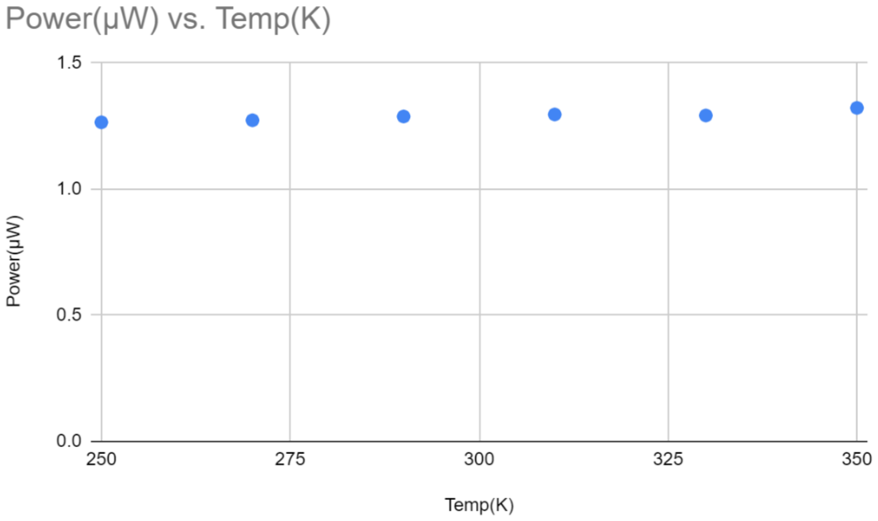


Figure 32: FF power consumption vs temperature

4 and 8 bit register schematics and symbols:

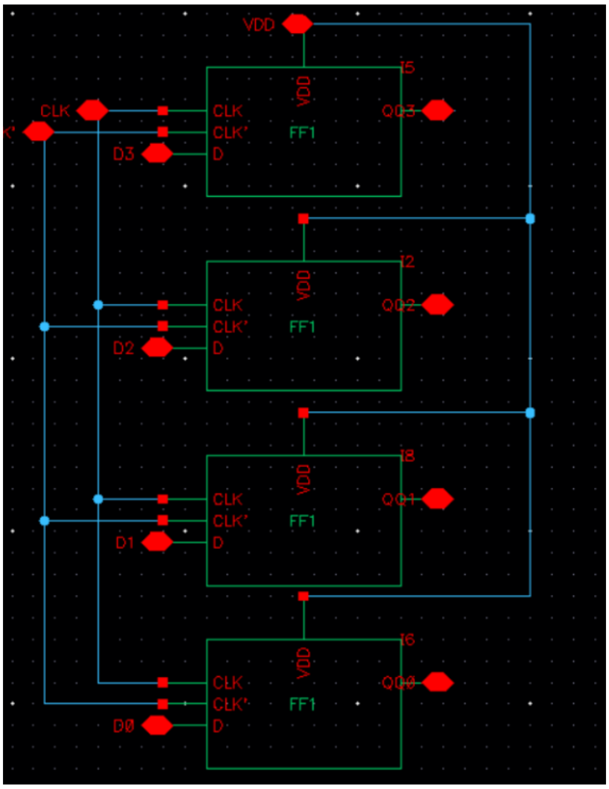


Figure 33: 4-bit register schematic

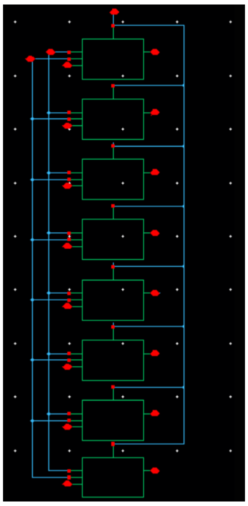


Figure 34: 8-bit register schematic

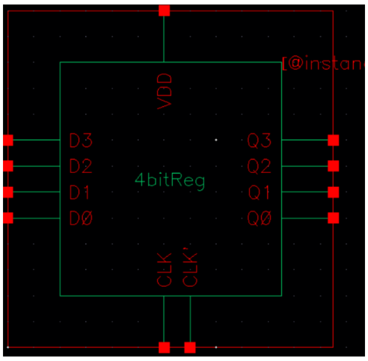


Figure 35: 4-bit register symbol

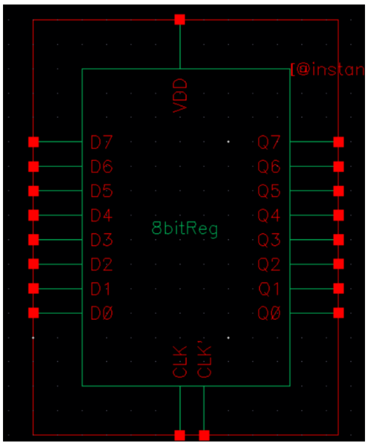


Figure 36: 8-bit register symbol

Part 5: SRAM cell

6T SRAM cell schematic:

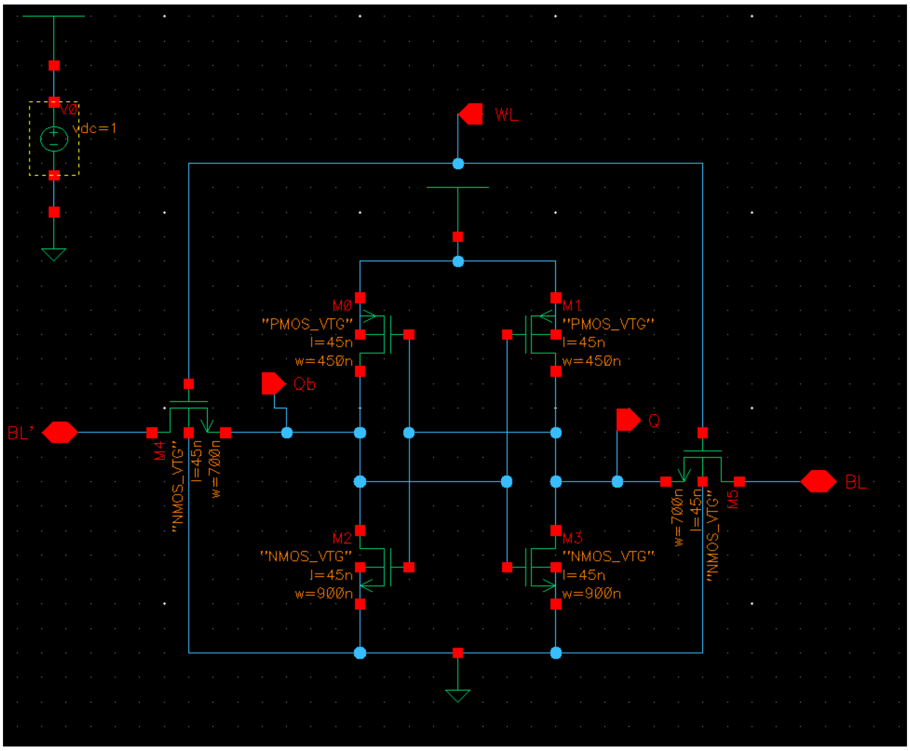


Figure 37: SRAM cell schematic

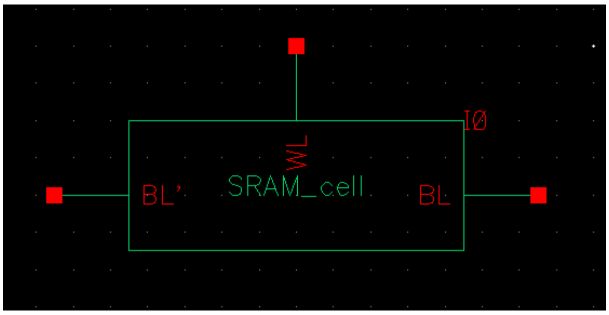


Figure 38: SRAM cell symbol

We sized our transistors in the following way:

Pull up network: 450nm
Pull down network: 900nm
Access: 700nm

Our noise margins were .31 V for the read and .332 V for the write. We found these values by plotting Vtrip and Vread and taking the difference. When reading a 1, Vtrip was our Q and Vread was our Q'. Below are our graphs of noise margin for read/write vs V_{dd} and temperature:

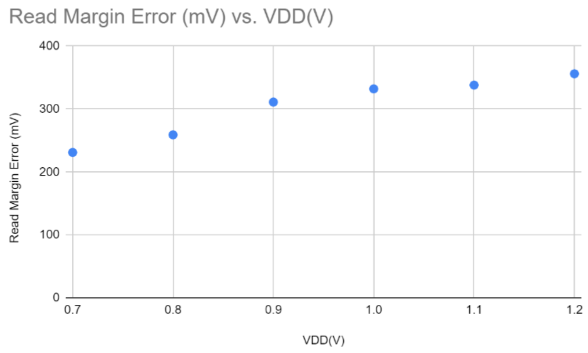


Figure 39: SRAM cell read noise margin vs V_{dd}

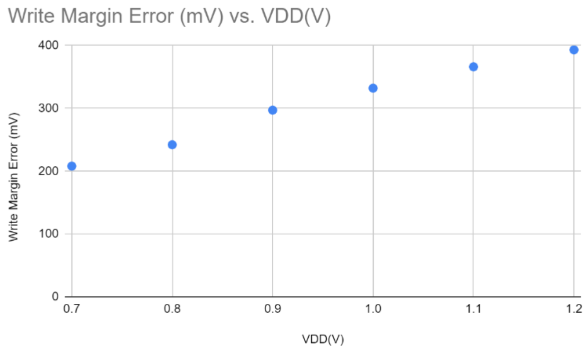


Figure 40: SRAM cell write noise margin vs V_{dd}

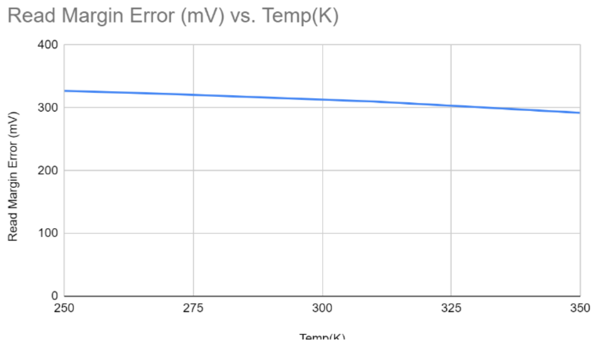


Figure 41: SRAM cell read noise margin vs temperature

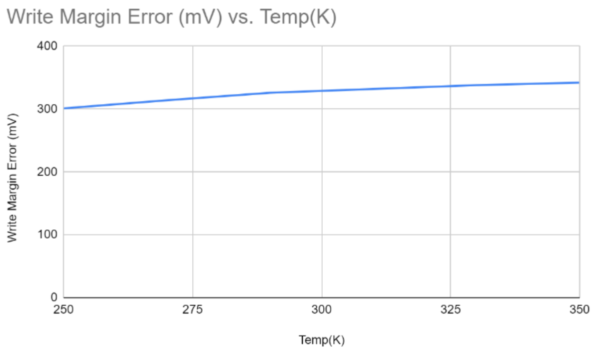


Figure 42: SRAM cell write noise margin vs temperature

Part 6: SRAM Array

We implemented our 32x32 SRAM using the "organized by bits" architecture. The SRAM is made out of 4 sub blocks. Each sub block consists of 32 rows and 8 columns. Each block has 1 3-8 decoder attached to its columns. The column address is applied to each of the 4 blocks, and the decoders select the corresponding bit in each block. The particular word is selected by inputting the 5 row address bits into a 5-32 decoder, selecting the associated word line.

Each column is precharged when clock is low using the following circuit:

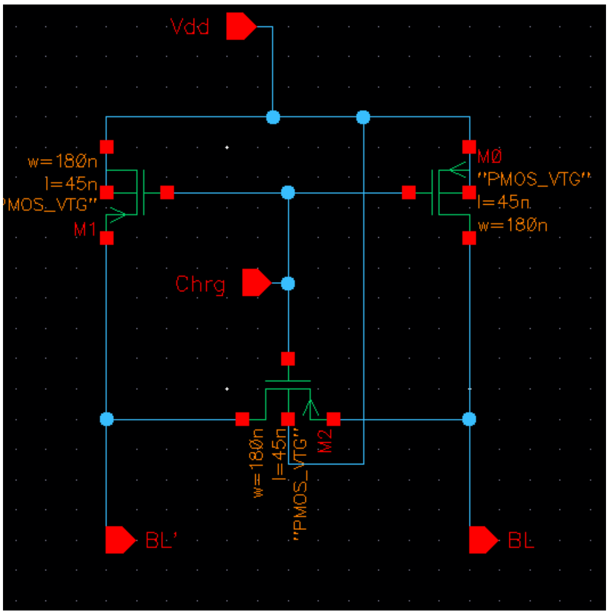


Figure 43: Precharge schematic

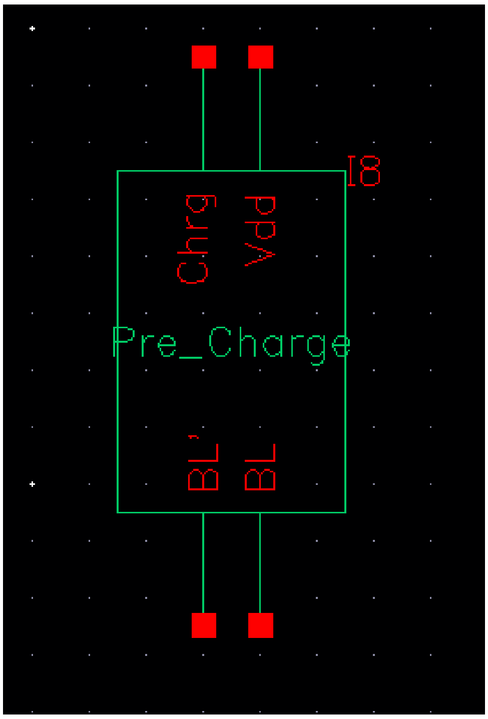


Figure 44: Precharge symbol

Each column also has a read/write circuit attached to the bottom of the column. When WE is high, the incoming data can be written to a cell if the S bit is also high. Data can be read when WE is low and S is high. The S bit is controlled by a column decoder.

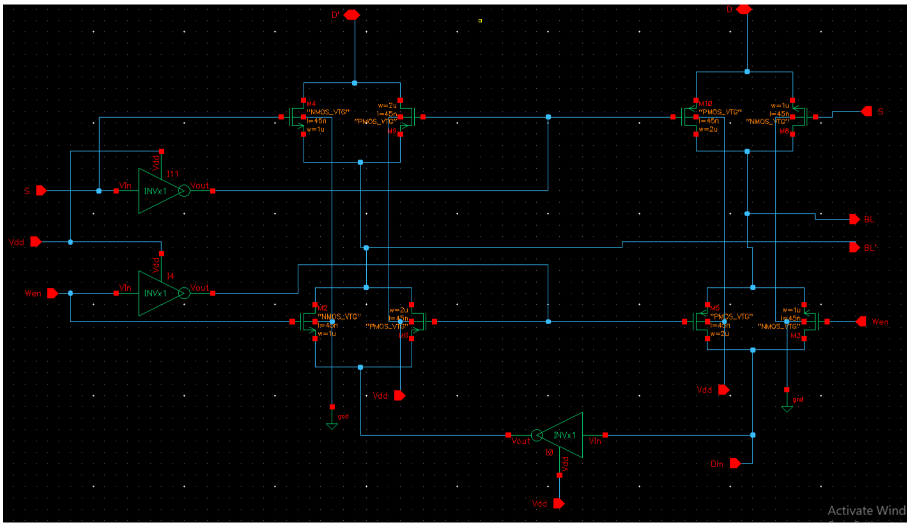


Figure 45: Read/write control schematic

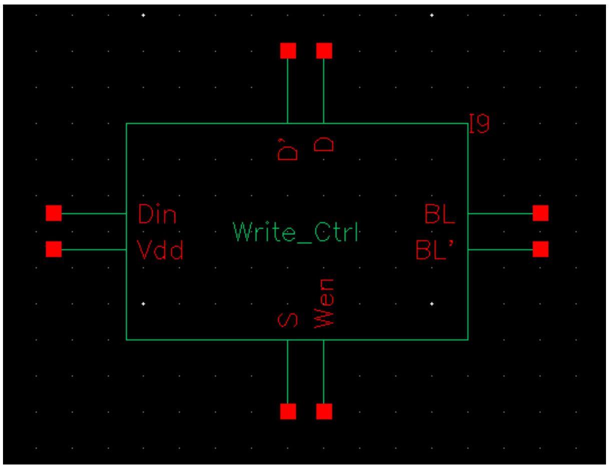


Figure 46: Read/write control symbol

Every block has a differential amplifier connected to it. When clock is low, if SAE is high, the differential amplifier will read the voltage produced on the BL and BL’ lines selected by the column decoder.

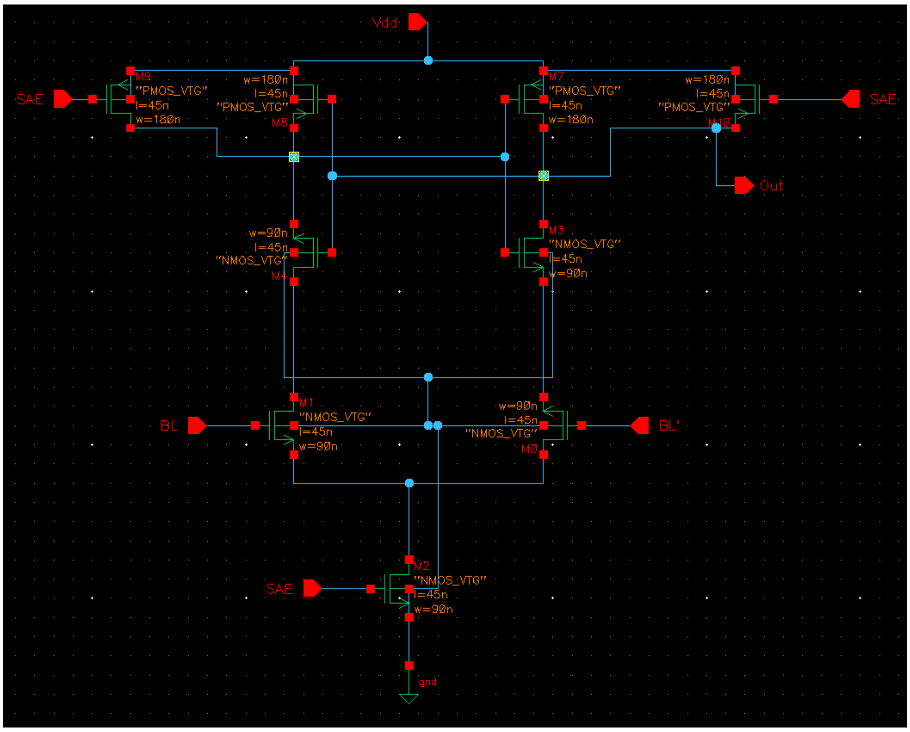


Figure 47: Differential amp schematic

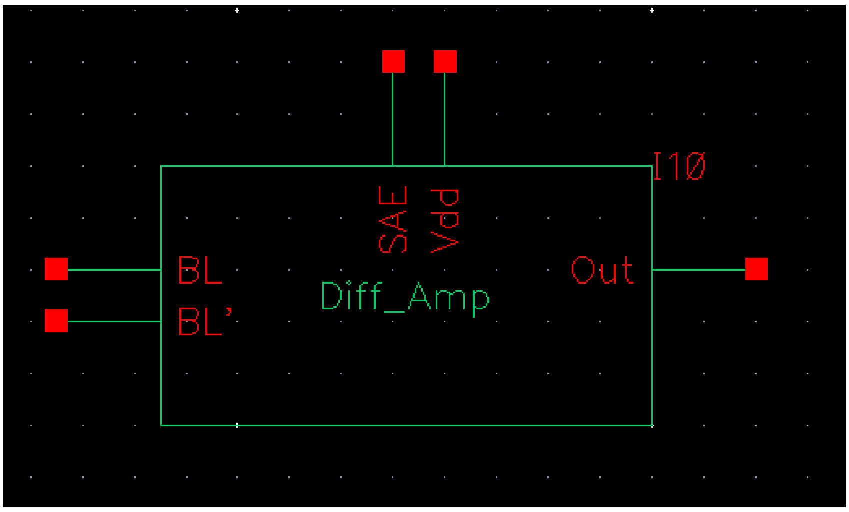


Figure 48: Differential amp symbol

Below shows how one SRAM column is constructed. Each column has one precharge and write/read circuit associated with it.

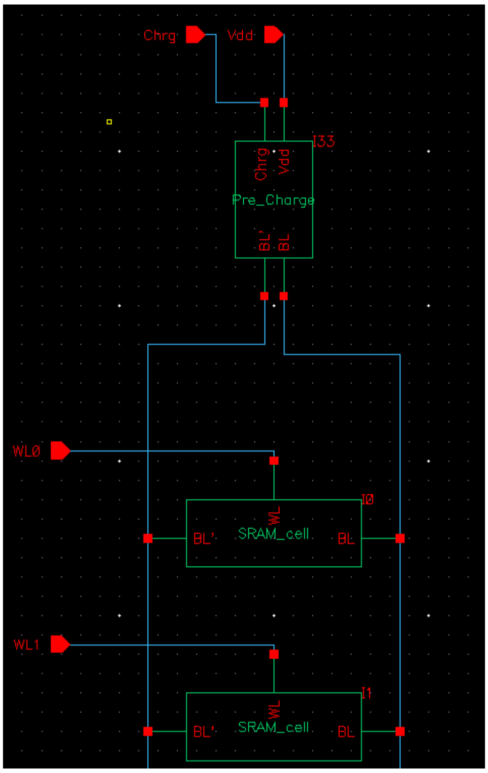


Figure 49: Column precharge

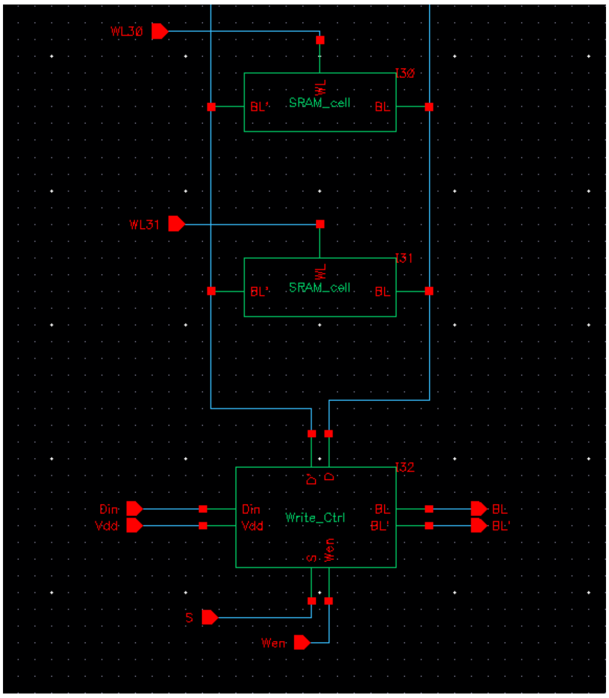


Figure 50: Column write/read control

Below is the schematic for 1 SRAM block. In the bottom you can see the 3-8 decoder associated with the block, and on the right you see the differential amplifier.

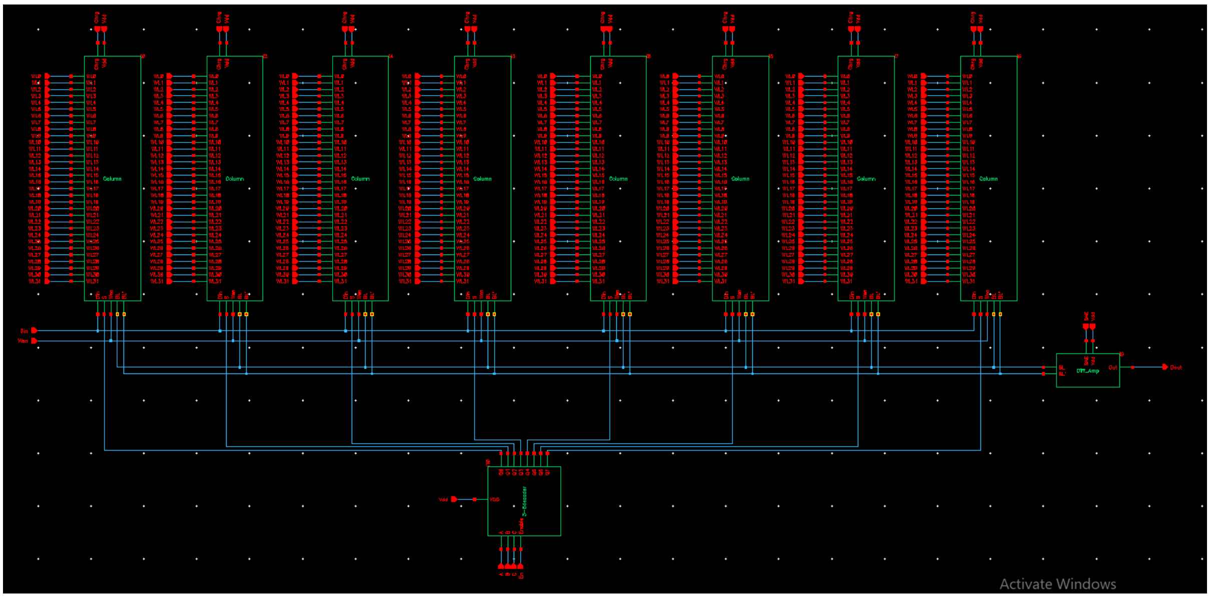


Figure 51: SRAM block schematic

Finally, we have the 32x32 SRAM schematic, consisting of 4 SRAM blocks, and the 5-32 bit decoder for WL selection:

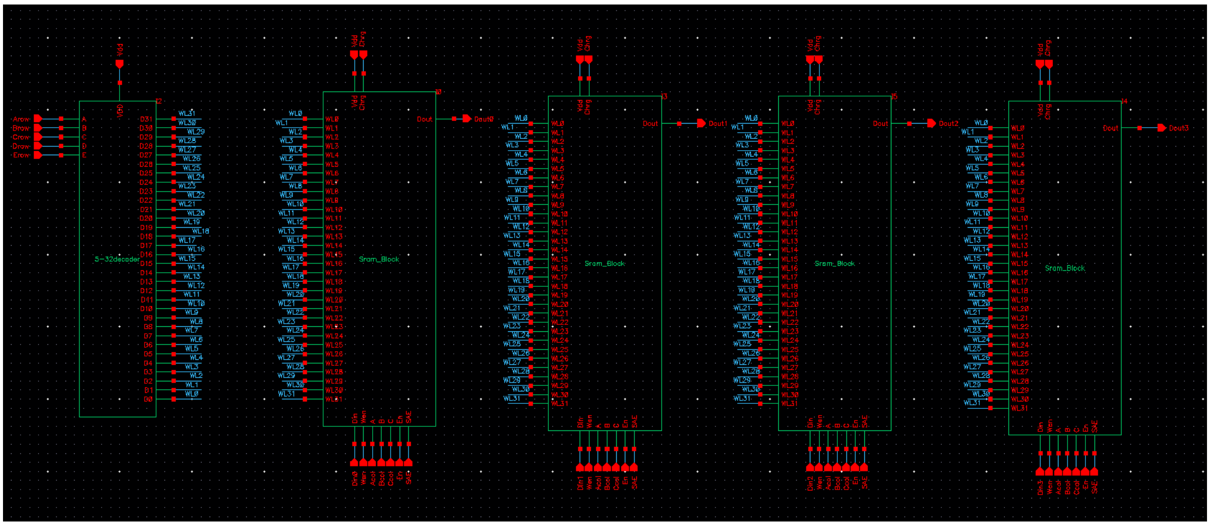


Figure 52: 32x32 SRAM schematic

Next, we show a waveform for a read and write operation in our SRAM cell:

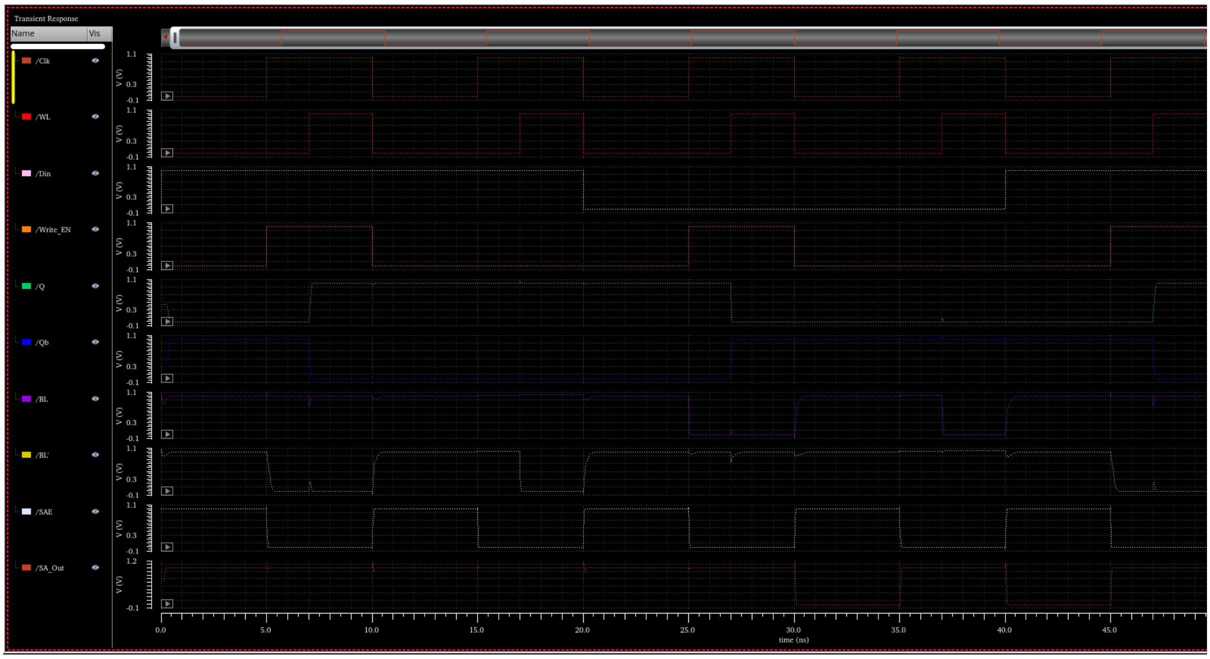


Figure 53: SRAM cell read and write operation

The top signal is the CLK, the second the WL, and the third is the input data. The fourth signal is write enable, WE. Below those, we see the Q and Qb signals, the values stored in the SRAM cell. We see that Q starts at 0. When CLK, WL, and WE are high, we see that a 1 is written into the cell. Below, we see the BL, BL', SAE, and SAE_out signals. The sense amplifier is enabled when the clock is low. We see that the output of the sense amplifier is 1 while the cell stores a 1.

Later on, we see that the input data becomes a 0. When CLK, WL, and WE are high, we see that a 0 is written into the cell. When CLK goes low, SAE is high, and the sense amplifier outputs a 0, which is the new value store in the cell.

Note that precharge occurs every time the clock is low in this waveform.

Part 7: System Integration

Below is the schematic of our fully integrated system:

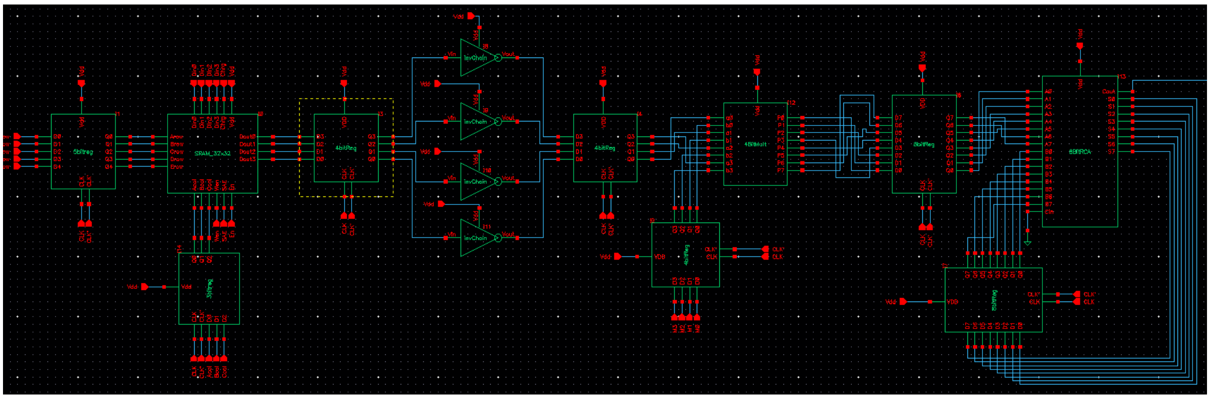


Figure 54: Integrated system

We tested all individual parts of the integrated system separately, and found that they worked to the correct specifications for system integration. Unfortunately, due to the size of the system, simulation of the fully integrated system too far too long.