Designing and simulating a DRAM using CMOS transistors

Kristtiya Guerra

May 2021

1 Introduction

Dynamic Random Access Memory, also known as DRAM, is a type of semiconductor memory. It is able to store memory in bits using transistors and capacitors.

A DRAM unit has of 3 major functions aside from just storing data.

1. Write/Overwrite Data
2. Read Data
3. Refresh Data

It is faster to use than an SRAM and uses less power, however, it discharges over time so it much be refreshed using a sense amplifier every few clock cycles. The refresh cycle is automated.

2 Circuit Component Design

2.1 Memory Cell

There are two options for the cell layout. The most common type of memory cell setup is the use of a single transistor and capacitor with the transistor acting as the pass transistor that is controlled by the row and column selector. The capacitor stores data and discharges over time.

Different layouts affect the discharge time and stability of the memory cells. I decided to use the first cell layout due to the results of the simulation shown in Figure 9.

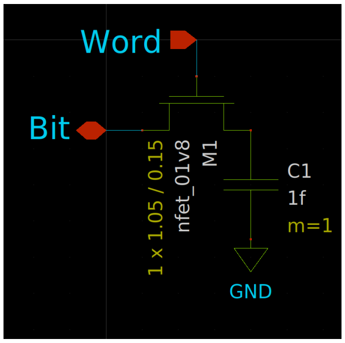


Figure 1: Memory cell schematic 1

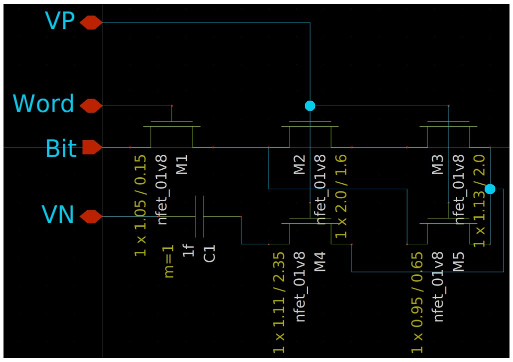


Figure 2: Memory cell schematic 2

2.2 Precharger

A precharger prepares a memory cell for writing, especially when a 1 must be written to memory since the transistors normally have strong 0's and weak 1's. This implementation does not require a complicated layout, just a pass transistor that ties the bitline to power when activated.

2.3 2 to 4-bit Decoder

The decoder consists of 3 subcircuits: a pre-decoder, 6 inverters, and a decoder with a pull-up tree design [4]. It is designed such that the width of the transistors near the root are larger than the widths of the lower transistors (leaves). I decided on a strength ratio of 2 between each level, with the minimum length of 0.15μm.

2.3.1 Pre-Decoder

The predecoder is rooted on 2 pMOS transistors with a width of $2\mu m$ and length of $0.15\mu m$. The remaining transistors have a widths of $1\mu m$ and lengths of $0.15\mu m$. Only the root transistors have dummy devices in order to simplify the layout of the circuit.

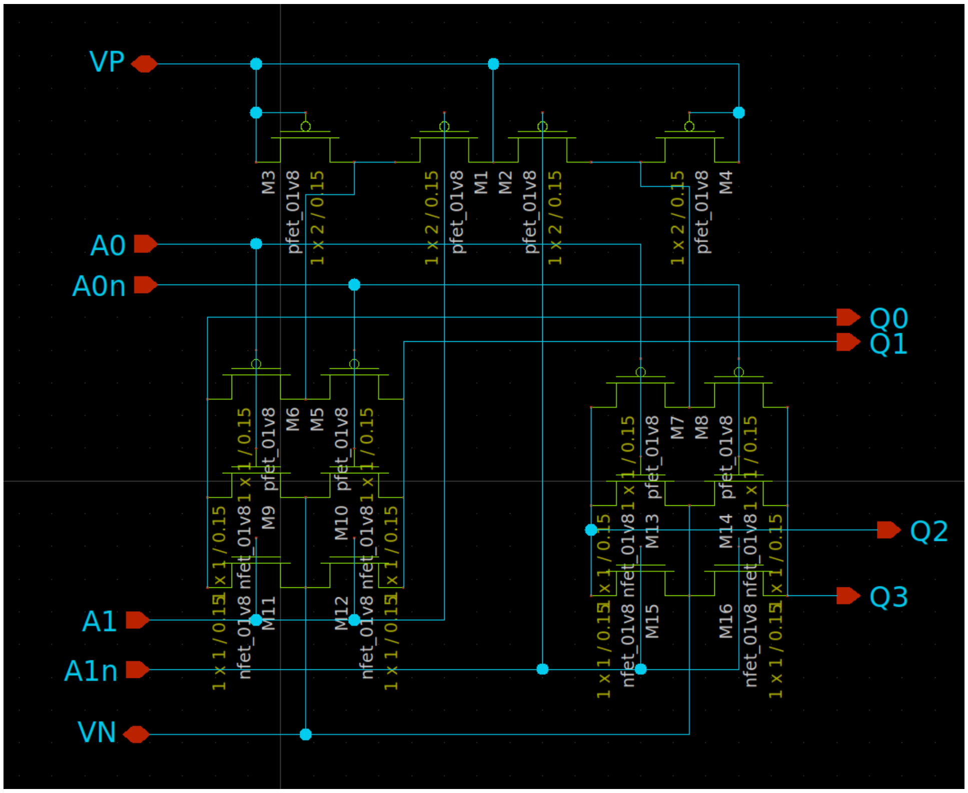


Figure 3: Schematic for the Pull-Up Tree Decoder.

2.3.2 Pull-Up Tree Decoder

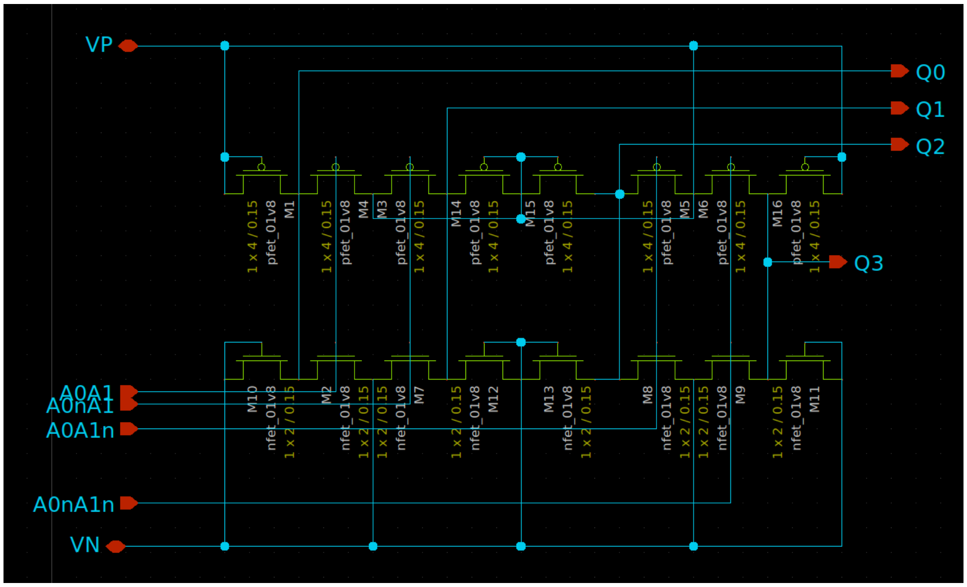


Figure 4: Schematic for the Pull-Up Tree Decoder.

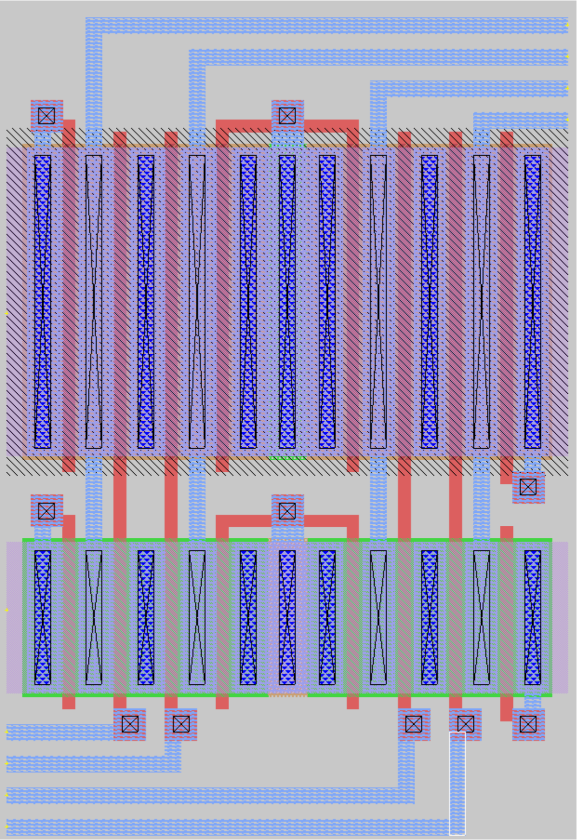


Figure 5: Magic Layout for the Pull Up Tree

2.4 Sense Amplifier

Inspired by the sense amplifier design in John Poulton's CMOS DRAM implementation [1], the sense amplifier used in the lab is made up of 3 nMOS transistors and 2 pMOS transistors. It is tied to power, with a pass transistor that activates the sense amplifier when enabled. It is connected to the even and odd bitlines.

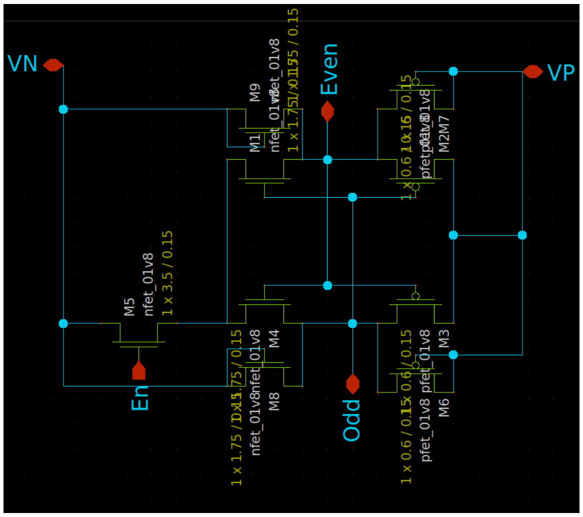


Figure 6: Schematic of Sense Amplifier

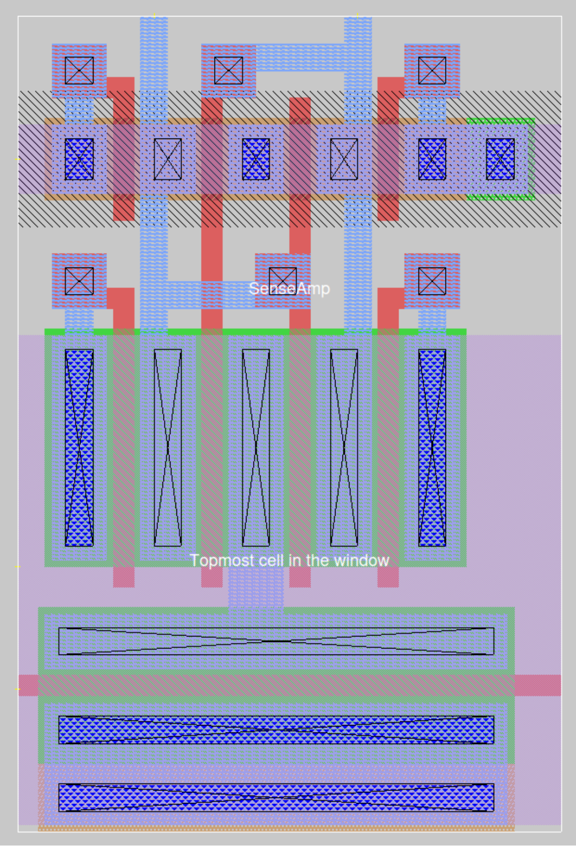


Figure 7: Magic Layout of Sense Amplifier

2.5 Refresh Cycle

Using CSRL-D-Flip Flops, the refresh cycle of the memory cell array is controlled with just the clock as a input. The refresh circuit is made up of 5 D Flip-Flops arranged as frequency dividers, dividing the system clock by 2 in each stage, and 4 D-Flip flops which act as the controls that select the word row of the cell array. There is an nMOS pass transistor such that when the $4^t h$ component in the frequency divider is high, the shift register is loaded and over the next 4 clock cycles, a single word is selected at a time. The $5^t h$ flip flop in the frequency divider activates the sense amplifiers for a refresh to occur.

3 Simulations

3.1 Memory Cell

The testbench is a transient simulation that loads data as a pulse that is 1.8V, and allows the cell to discharge once the read pulse is high.

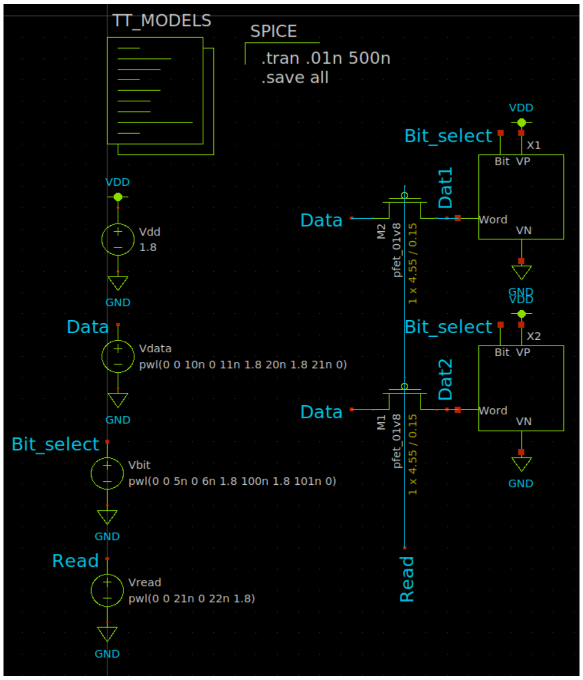


Figure 8: Testbench used to measure the discharge rate of the cells

As shown in the results, the discharge of the two cell types are similar, however, the blue line, representing cell type 1, drops from 1.8V to 1.2V to 1 V, which is higher than cell type 2, which tapers off to around .7V.

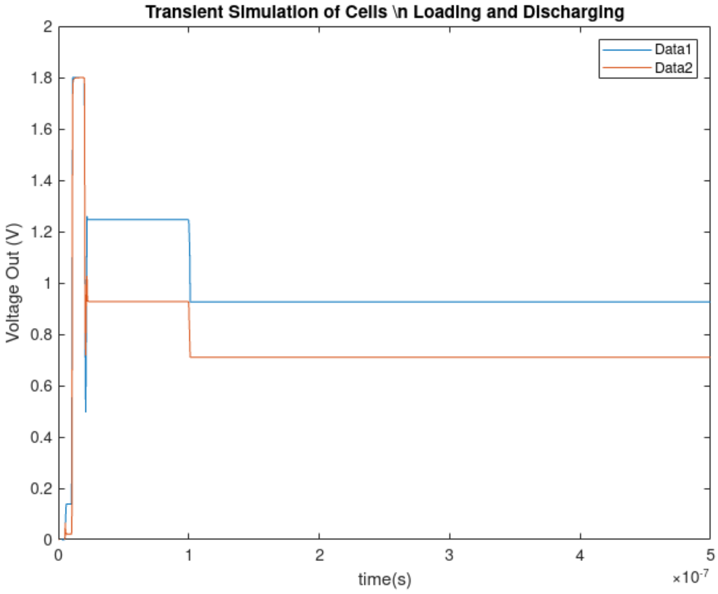
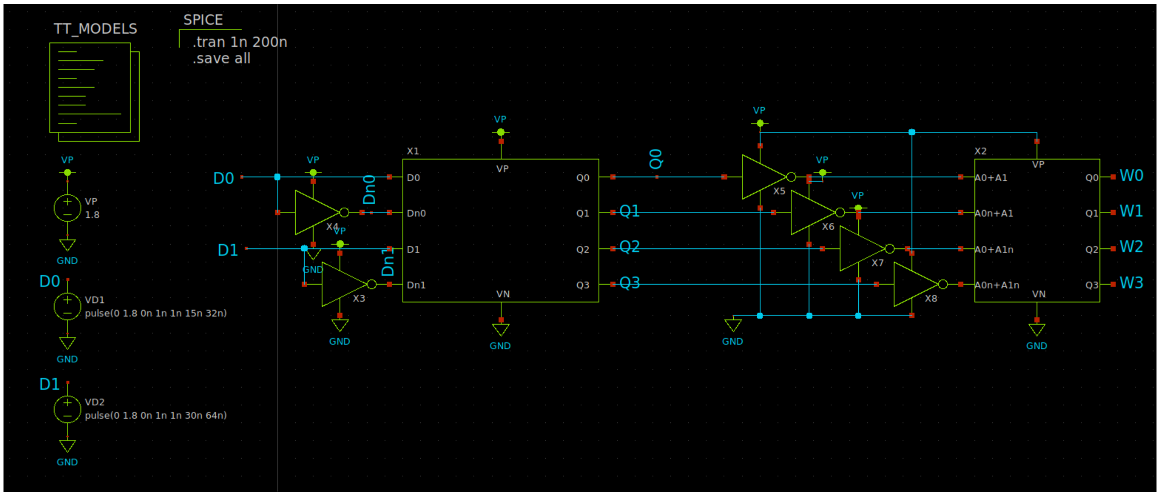


Figure 9: Results from cell discharge simulation

The discharge rate of the two memory cells also vary. The discharge rate of data cell 1 is approximately $-1.2743 * 10^{-9}$, whereas the discharge rate of the second layout was around $-2.9299 * 10^{-9}$.

3.2 Decoder

To test the functionality of the decoder, two pulses were used, one with a period of 31.25MHz, and one with a frequency of 15.63MHz. They represent the digital data that is decoded to determine the word or bitline that the system is to use. They align such that all 4 logical input combinations occur, as shown in the first two rows of the simulation (Figure 12).



```
.tran 1n 200n
.save all
```

```
pulse(0 1.8 0n 1n 1n 15n 32n)
pulse(0 1.8 0n 1n 1n 30n 64n)
```

Figure 10: Testbench setup of the decoder simulator

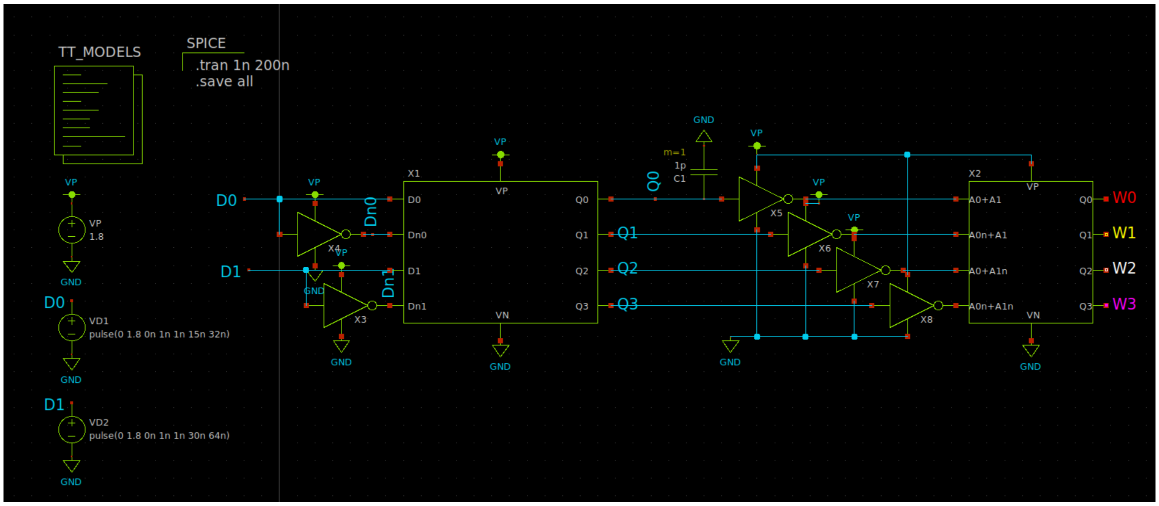


Figure 11: Decoder setup with pull-down capacitor at the output Q0 of the pre-decoder.

The overall layout is based off of Richard F. Lyon and Richard R. Schediwy's implementation of a CMOS DRAM [4].

The simulation runs for 200ns. In Figure 12, there seems to be a spike on the output of D0 and W0. This is due to the time where all inputs are 0 for a quick few nanoseconds. We do not want W0 to be high during the transient response, so a 1pf capacitor was added (Figure 11)

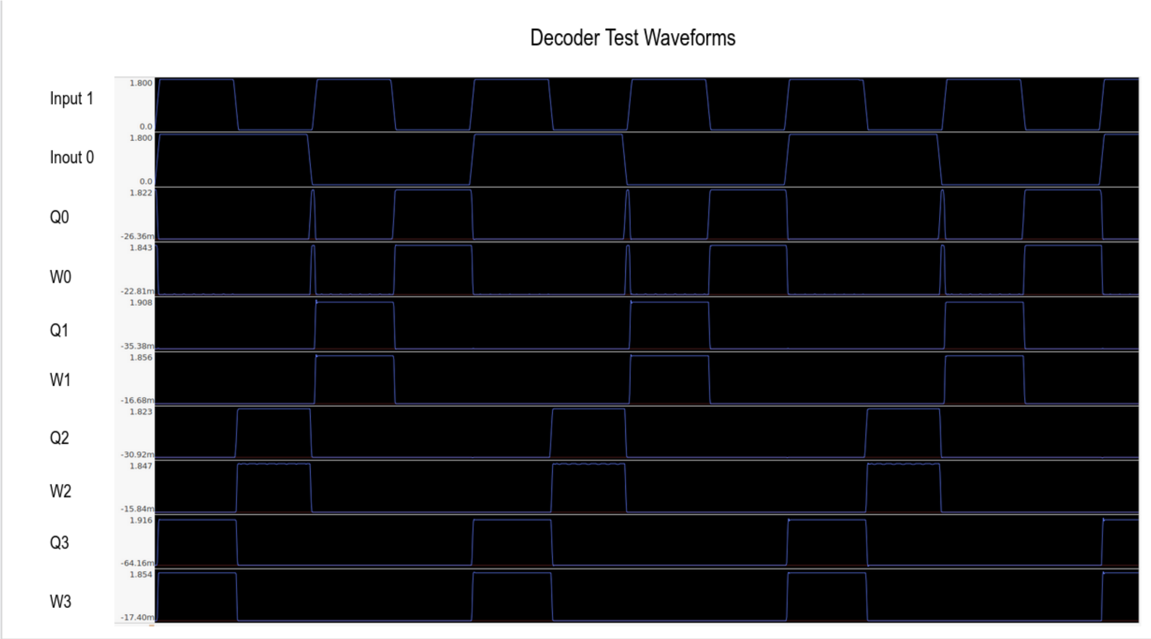


Figure 12: Waveforms of the decoder simulation, with Q0,Q1,Q2,and Q3 representing the output of the pre-decoder, and W0,W1,W2, and W3 representing the output of the decoder.

By adding a pull-down capacitor at the output of the pre-decoder, the transient spike where both inputs were low was reduced to an acceptable level. When both inputs were intentionally low, the output for Q0 was able to reach 1.8V. Although the rise-time increased, the output of the pull-up tree decoder was still a 1.8V pulse.

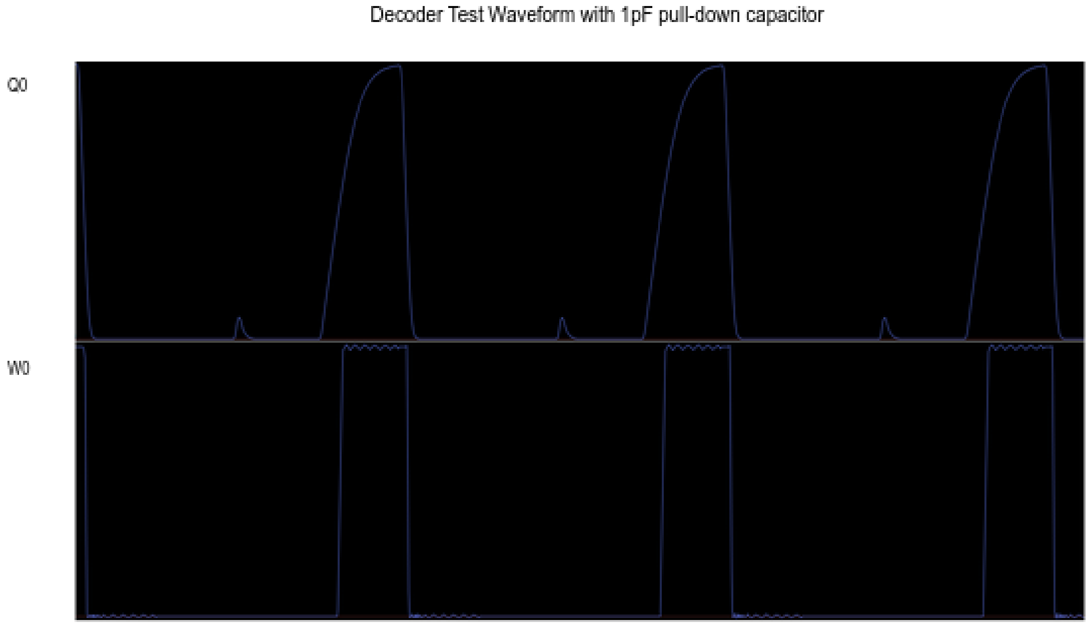


Figure 13: Waveform of Q0 and W0 with a 1pF pull-down capacitor implementation

4 Netlist

In order to compare the schematics and layouts of each component, I made layout-driven schematics, and then created the layout in Magic. I then used Netgen, and the provided command, lvs, to compare the netlists between the Magic layout and the Xschem schematic. Upon running the comparison, I then looked for the the line:

Circuits match uniquely.
Netlists match uniquely.

Along with the above lines, I also made sure number of nets and number of transistors matched to eliminate any mismatches. My results can be viewed in my repository under the Netlist folder.

5 Reflection

I was unable to fully implement a CMOS DRAM. This is largely due to overscoping. I wanted to create a 4x4 CMOS DRAM, however, I likely should have started with just a single column and started debugging. As a result, I was only able to make the components, but could not find a way to put them all into one layout. I successfully created the decoder circuit which I feel very achieved about. It takes in a digital inputs, 0 and 1.8V, and outputs an analog output that is around 1.8V when high and 0 when low. The analog outputs are accurate enough to become inputs to the wordlines and load the cell arrays.

I was unable to implement a counter that could automate the refresh cycle of the DRAM. The CSRL D-Flip Flops worked, however, when the D output was loaded onto any other subcircuitry, it lost the behavior recorded when measured on its own. I messed around with an ADC design, which can be viewed in my repository, to see if it could make the output of my flip flops my outputs solid, however, it only produced high signals.

I will likely continue working on this project over the next week to get a working circuit or my benefit as I do feel close to a working circuit, however, I am stuck at designing my refresh timer and trying to create an automated simulation.

Link to view repository:
https://github.com/Kristtiya/CMOSDRAM

6 References

1. Poulton. J, ”An Embedded DRAM for CMOS ASICs,” Department of Computer Science, University of North Carolina at Chapel Hill, Sept. 16, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.45&rep=rep1&type=pdf

2. Garg. M. R, Tonk. A, *A study of Different Types of Voltage and Current Sense Amplifiers used in SRAM.* International Journal of Advanced Research in Computer and Communication Engineering. 2015. May. [Online]. Available: https://www.ijarcce.com/upload/2015/may-15/IJARCCE%207.pdf

3. Lapedus. Mark,” DRAM Scaling Challenges Grow,” Semiconductor Engineering. 2019. [Online]. Available: https://semiengineering.com/dram-scaling-challenges-grow/

4. Lyon. R. F, Schediwy. R. R, ”CMOS Static Memory with a New Four-Transistor Memory Cell,” Schlumberger Palo Alto Research, Palo Alto, California, 1987. [Online]. Available: