Analysis and Layout of a 4x4 6T SRAM Array

Armin Ekic

I. Introduction

In this report I will explain my process of creating a 4x4 Array of 6T SRAM cells and the characteristics measured. First, I will talk about a single SRAM cell with its accompanying schematic, symbol, and layout; then I will touch on the hold, read, and write static noise margins, as well as the static power dissipated while a single cell is holding a value. Next, I will touch on the process of creating a sense amplifier (with the corresponding schematic, symbol, and layout), as well as the read and write access times of a single 6T SRAM cell connected to a sense amplifier. Finally, I will explain how these components connect to create a 4x4 memory array and the layout; as well as what problems I ran into throughout the process with some proposed changes to my design.

II. 6T SRAM Cell

A. Schematic

An SRAM cell is the basic building block for many memory structures and is a fantastic way to store data. The schematic for a 6T SRAM cell is composed of two cross-coupled inverters with an access transistor (NMOS transistors) on either side. The gates of the access transistors are controlled by the word line (WL) and the source/drain terminals are connected to the bit line (BL) and the cross-coupled inverters. The circuit functions by allowing access through the 2 pass-gate NMOS transistors connected to the WL, while BL and BLn are used to read from and write to the cell.

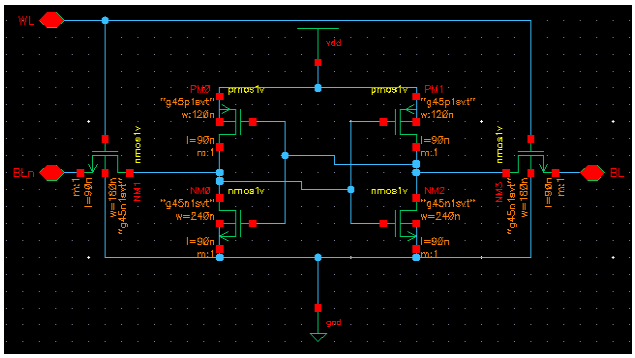

Figure 1: 6T SRAM Cell Schematic

The transistor sizing of an SRAM cells requires the following relationship to hold true to promote cell stability: $W_{pull\text{-}down} > W_{access} > W_{pull\text{-}up}$. Using a technology size of 90nm, the pull-up transistors were sized L = 90nm, W = 120nm, the access transistors were sized L = 90nm, W = 180nm, and the pull-down transistors were sized L = 90nm, W = 240nm.

B. Symbol/Layout

Once the 6T cell was created the symbol and layout were composed and tested accordingly. The layout used the metal 1 and poly layers to make the signal connections to the WL and BL and passed both DRC and LVS checks to ensure the layout is valid and equivalent to the schematic.

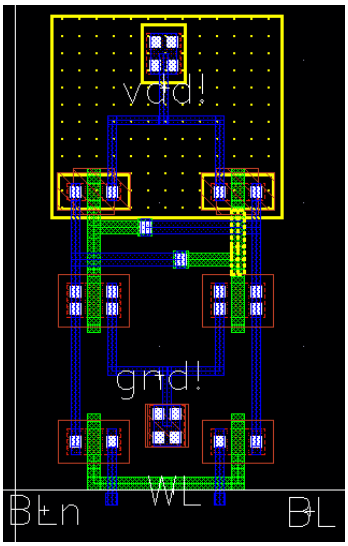

Figure 2: 6T SRAM Cell Layout

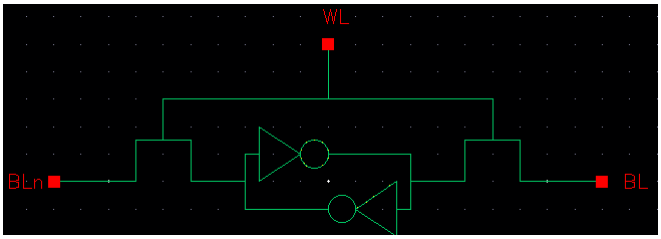

Figure 3: 6T SRAM Cell Symbol

As can be seen, the SRAM cell is only accessed through the WL, BL, and BLn input/output pins. These allow the reading from/writing to the SRAM cell.

C. Hold Static Noise Margin

The hold margin is the static noise margin (SNM) observed when the cell is holding some value. This is determined by the butterfly diagram which is found by plotting the transfer characteristics across each inverter against each other. As a result of the inverters being identical, the DC transfer curves are mirrored along Vm. Once the curves are plotted, the hold state SNM is determined by the length of the largest square that can be inscribed between the curves [1].

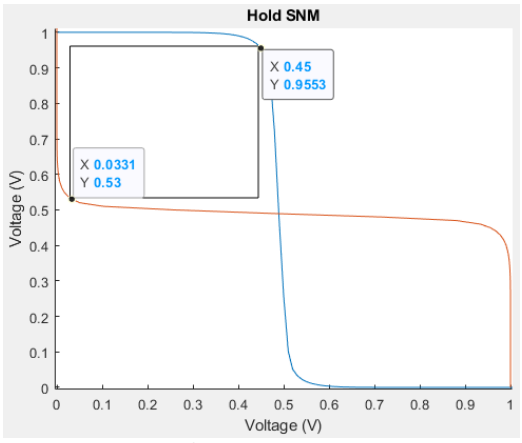

Figure 4: Hold SNM for single cell

As can be seen from the graph, the hold SNM for a single 6T SRAM cell is roughly 0.45V, which is the side length of the given square. This means that 0.45V is the voltage threshold that the cross-coupled inverters hold.

D. *Read Noise Margin*

The read margin is similar to the hold SNM, and is found using the same methods, but finds the noise margin characteristics when a "1" or a "0" is being read from the SRAM cell. When a read is being done, we need to pre-charge the bit lines and observe which bit line loses it's charge and goes to "0", meaning that the read margin is dependent on the strength of the pull-down transistor in relation to the strength of the access transistor [1]. This means that the plotted curves for the read noise margin will not be uniform as the hold SNM curves are, but we will still see some symmetry because of the inverters' uniformity.

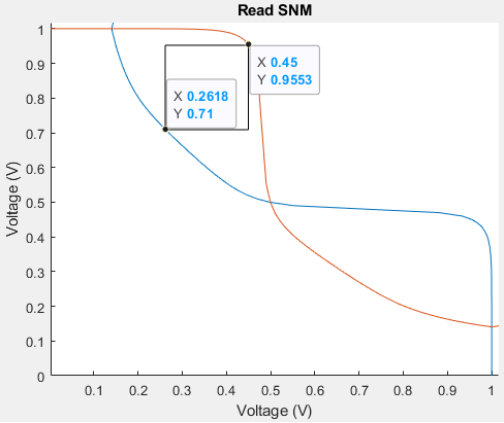

Figure 5: Read SNM for single cell

As can be seen with the above graph, the design observed a read noise margin of about 0.19V.

E. *Write Noise Margin*

The final noise margin measurement taken shows the write margin, which is simulated by pulling one access transistor to "1" and the other to "0". This margin is improved as the access transistor becomes stronger and the pull-up transistor becomes weaker, or as the WL voltage increases [1].

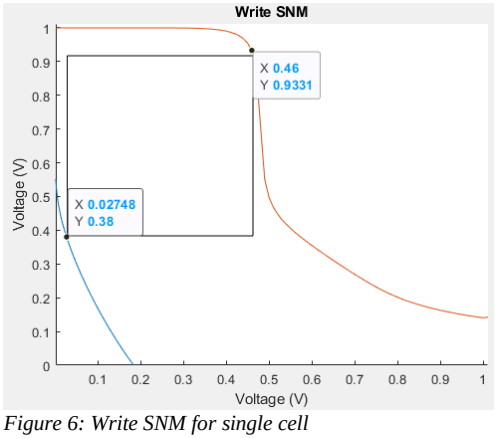

Figure 6: Write SNM for single cell

This graph shows a write margin of about 0.43V, which is very similar to the hold SNM characteristic.

F. *Static Power*

For the single 6T SRAM cell, the last component I tested was the static power dissipated when the design is in the hold state. This was tested by creating an expression to represent the average power drawn through the voltage supply over 25 ns (an arbitrary length), and by displaying the transient voltages and currents for each net in the system when the voltage source is applied.

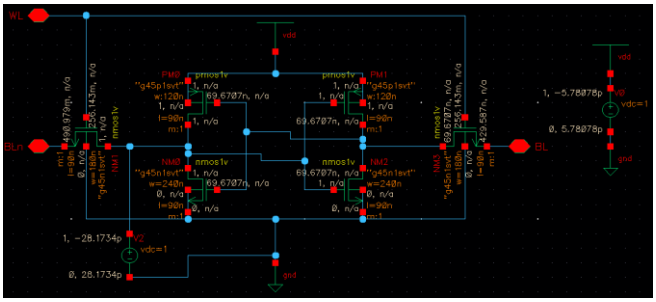

Figure 7: Static Power Dissipation

It may be difficult to see in the figure, but the power dissipated from the voltage source and the value held in the cell is 5.8 pW and 28.2 pW respectively, which means the total power dissipated is in the order of about 34 pW.

III. Sense Amplifier

A. *Schematic*

A sense amplifier is a circuit that amplifies the difference between two given signals, which in this case are BL and BLn. The circuit used features two cross coupled inverters, an equalizing transistor (EQ), and two select transistors (SE, SEn). The equalizing transistor is used to pre-charge BL and BLn to the same value, while SE and SEn are used to connect the cross coupled inverters to Vdd and GND, enabling the circuit.

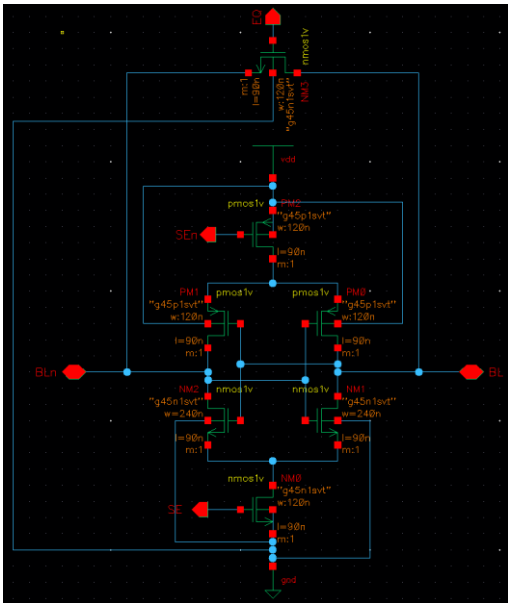

Figure 8: Sense Amplifier Schematic

The cross coupled inverters behave similar to and operate under the same constraints as those within the 6T SRAM cell, so they are sized identical to those described in

section B of header II. Furthermore, as the equalizing and selecting transistors do not hold any hard constraints, they are sized minimally for our design (L = 90nm, W = 120nm).

B. Symbol/Layout

The layout and symbol of the sense amplifier is straightforward, and contains 3 inputs of EQ, SE, and SEn, with 2 input/output signals of BL and BLn. Similar to the single cell layout, the sense amplifier underwent both DRC and LVS checks.

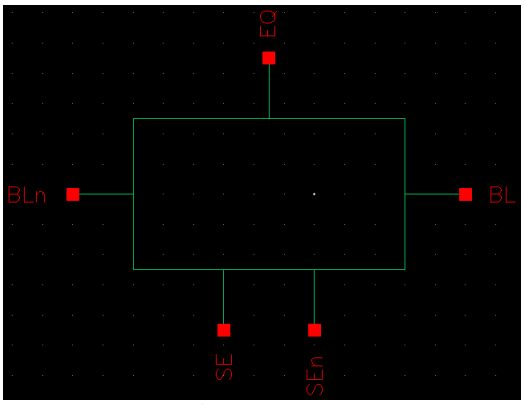

Figure 9: Sense Amplifier Symbol

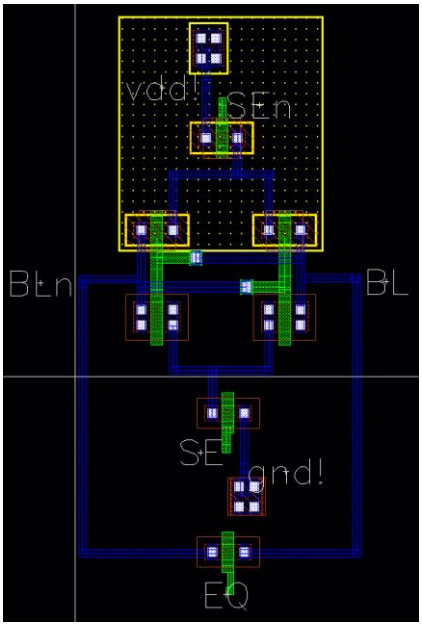

Figure 10: Sense Amplifier Layout

The sense amplifier layout was made similarly to the single 6T SRAM cell layout (not minimally sized, reflective of the orientation of the schematic).

C. Read Access Time

The read access time of an SRAM cell is the time that it takes to read the value being held in the cell. This was found by initializing the stored value in an SRAM cell to “1”, connecting the cell to a sense amplifier, then performing a read operation to graph the change in voltages at each of BL and BLn.

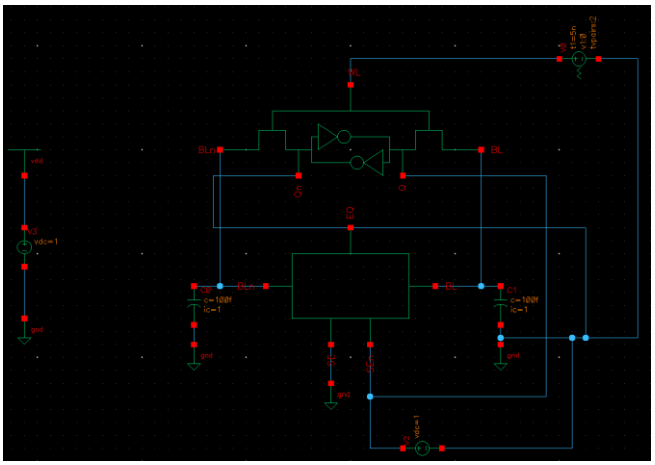

Figure 11: Read Access Time Circuit

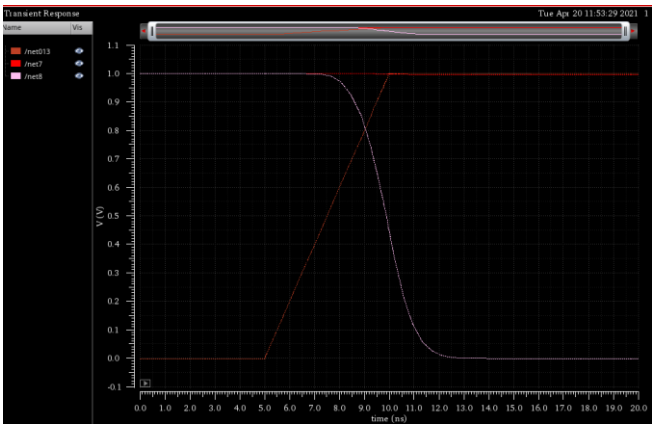

Figure 12: Read Access Time

As can be seen from the produced graph, the read access time is about 5ns, which is calculated observed as the amount if time it takes the BL with a difference in voltage to discharge.

D. Write Access Time

The write access time of an SRAM cell is the period it takes to successfully set the stored bit of an SRAM cell to a particular value. The operation is dependent on the access transistors being stronger than the pull-up transistors, so that the desired value can be forced into the cell. This means that the write access time measures how long it takes for this operation to occur.

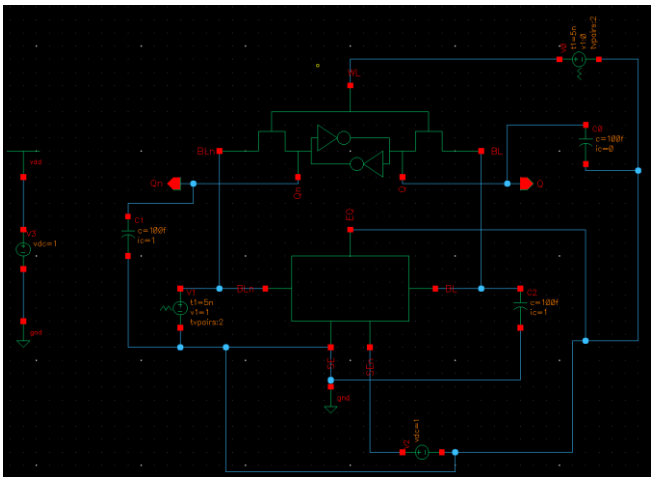

Figure 13: Write Access Time Circuit

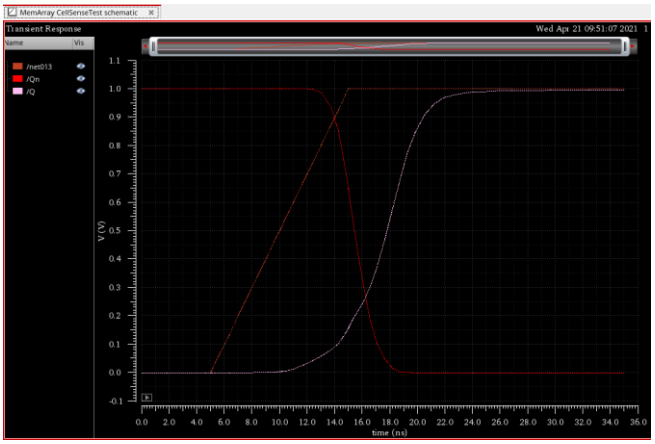

Figure 14: Write Access Time

As can be seen from the produced write access time graph, the estimated time to force a value into the cell is about 12ns, or a little over double the read access time. To improve this time constraint, we could make the pull-up transistors weaker or make the access transistors stronger.

IV. 4x4 Memory Array

A. Schematic

The schematic for a 4x4 memory array was created using the symbols of the 6T SRAM cell and the sense amplifier shown above. This array has 4 sets of word lines and 4 sets of bit lines, allowing us to access any of the 16 individual cells without altering any of the others. Given this 4x4 array, sense amplifiers can be connected to the bottom of the array to read the value of each cell for its corresponding column. The operations for this array are performed identically to a single cell setup and can be expanded to create much larger memory arrays.

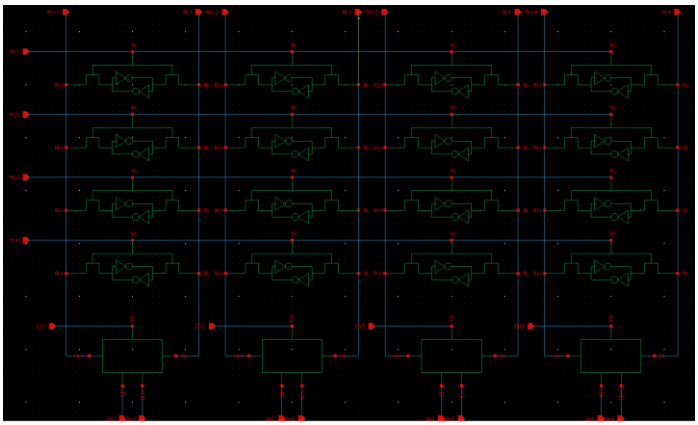

Figure 15: 4x4 Array Schematic

This schematic shows all the relevant signals needed for a memory array to function properly but is still a simple implementation with little optimizations.

B. Layout

The layout for the 4x4 memory array is not optimized for space but is still a valid implementation. My main goal in creating the layout was to make sure all DRC and LVS checks were satisfied, and that any additions to the layout were not damaging/different to the overall design.

As can be seen, the design is identical to the schematic except for the shared ground line among all the cells and sense amplifiers. This means that the LVS resulted in a warning when comparing the layout and schematic which originated from the required GND line tying all individual GND signals to a single port. This warning was ignored because it was required to satisfy the DRC and did not affect the implementation.

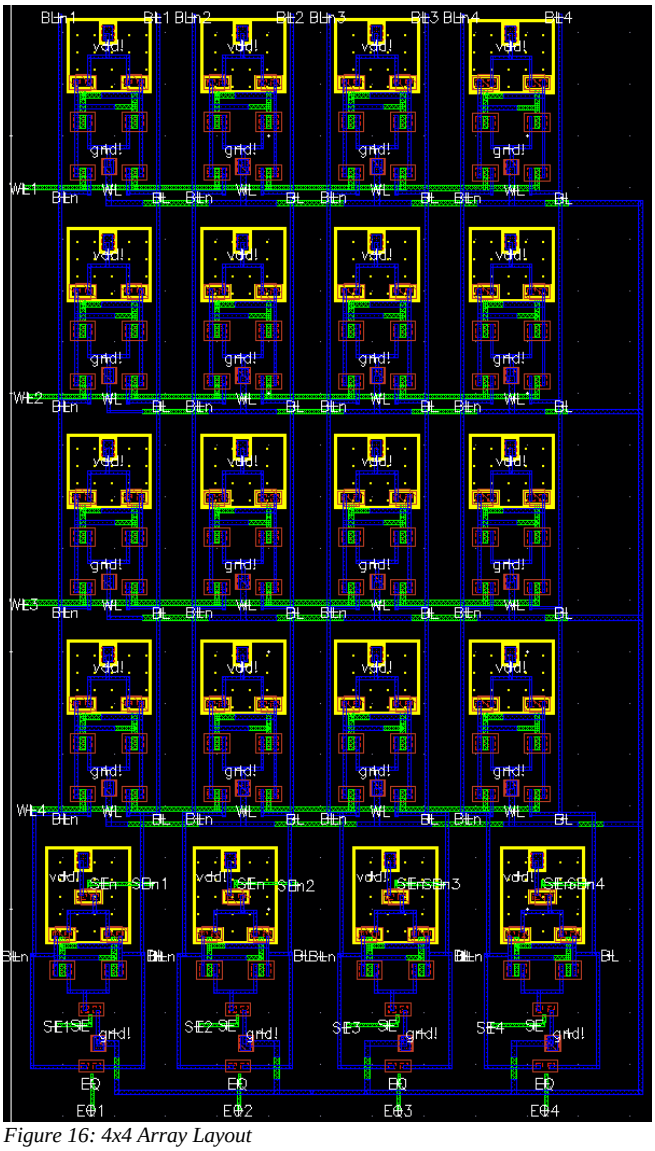

Figure 16: 4x4 Array Layout

V. Reflections

A. Single 6T SRAM Cell

While the schematic and layout for the single cell produced valid data and functioned as intended, there is still much room for improvement in the layout. The proposed layout does not take advantage of space and can be much more compact. This can be done by building the cell "tall" and minimizing the horizontal space the cell takes up. The following design by Jaydeep P. Kulkarni shows an example of this, and in comparison, to my design, it can be seen that WL, BL, and BLB are all essentially set up as "bars" that run through the rest of the layout, whereas my design treats

them similarly to how they were realized in the schematic [5].

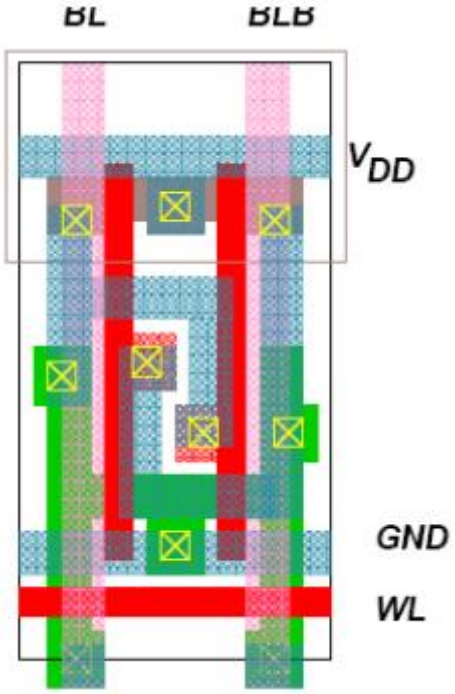

Figure 17: Compact 6T SRAM Cell

This did not only occur with WL, BL, and BLn in my design, because the transistor placement of the compact design minimizes the area that each takes up by placing relevant connections and nets closer to one another. My design simply places the components in a similar fashion to how the schematic was designed. My intention when creating the layout was to learn how everything functioned and to ensure it was valid, rather than making sure it was as compact or efficient as possible.

Another area that I can focus on improving within the layout is with the connection layers. All the connections in my design were created using the metal 1 or the poly drawing layer, whereas a better method would involve using the metal 2 layer for the BL and BLn nets, and for any necessary vias. This is a more standard method of design, and as can be seen with the SRAM array, using the poly layer too much within a single cell can exacerbate the issue.

B. 4x4 6T SRAM Array

The memory array come design, schematic, and layout were produced as expected, but there is still much room for improvement. One major issue I ran into from the simulation aspect is with my LVS checks. Looking at the design, it's clear to see that a common GND rail was used to connect all cells and sense amplifiers in the array. This is a connection that did not exist in the schematic, so the LVS check produced a warning that was not able to be fixed. While this posed no issues for the design, it made it harder to create an extracted schematic and simulate a post-layout design.

A change that I would make if I were to recreate the project again would be to replace a poly drawing connections with metal 2 layer connections. As I said previously, this does not directly impact the design, but is better practice and is a very easy and realizable fix.

C. General Issues/Roadblocks

This project covered a topic that was completely new to me, and much of the concepts came with learning curves. The first thing that tripped me up was transistor sizing for the single cell. The cell ratio is a ratio of the width divided by the length of the pull-down transistor size, and the width divided by the length of the access transistor size. This has a major effect on maintaining a reasonable node voltage, which if too high can rise over the transistor threshold voltage and negatively impact the operation of the SRAM cell. To achieve a desirable cell ratio without seeing diminishing returns, a value greater than 1.2 should be achieved.

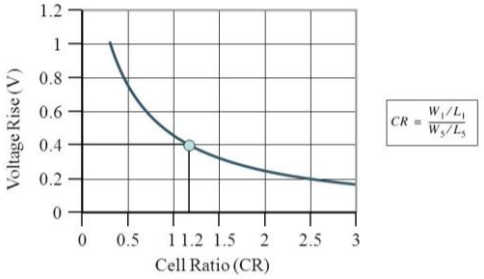

Figure 18: Cell Ratio of SRAM Cell

The cell ratio for the design is the first thing that tripped me up, because the constraints on driving strengths needed to operate an SRAM cell efficiently didn't occur to me. After reading up on this aspect of SRAM design, I went with a design that holds a cell ratio of 1.3. This means that all sizing constraints are met while also attempting to minimize the overall size of the device.

The next issue I ran into was concerned with running simulations to obtain the cell's characteristics. I am still new to using Cadence, so I found it difficult at times to collect relevant data for the read/write access times, and the SNMs. For example, after creating the schematic, symbol, and layout for a single cell, I realized that I could not set the value within the cell to a "1" or "0" because if left untouched it stayed in a metastable state. This is fixed by creating a signal connected to the cross-coupled inverters that would represent the output, and the inverted output. These outputs were used to tie values to the cell, as can be seen in figures 11 and 13. The issue that this caused arose when the layout no longer matched the schematic and the symbol for the 4x4 array contained 32 new signals that are never meant to be used in any actual application. This meant that I had to change the design again and "retrace" my steps of altering the design. I believe this is one example that shows my inexperience with Cadence (which was the biggest roadblock in the completion of the project), but also my ability to fix these issues in my own way.

The last major issue I ran into is concerning the overall 4x4 memory array. As I said previously, the post-layout extracted schematic was not created which made any final data collection more difficult. This is something that I

wanted to complete, but nonetheless I believe much of the characteristics obtained for a single SRAM cell are indicative of how the array operates. With my implementation, we will not be accessing more than a single cell at any given time so the hold SNM, read/write margins, read/write access times, and static power dissipation can be extrapolated across the entire 4x4 array. These graphs and data points paint a great picture of how the entire array functions on a cell-by-cell basis.

VI. Conclusion

I chose to create a 4x4 6T SRAM array because this is a topic that is incredibly interesting to me and is incredibly relevant in modern industry. My goal was to gain a greater understanding of how SRAM cells are created, how they function, what characteristics to look for, and how they operate. To do this, I identified transistor size, static noise margin, static power dissipation, and read/write access times as major measurements that do a good job of indicating the performance of an SRAM cell. Through this I gained a deeper understanding of Cadence and how to gather/understand data using the tools provided.

The data collected was analyzed and compared to expected measurements found in previous papers and textbooks. This gave me a way to understand not only how/why the data is collected, but also to ensure the validity of everything gathered. For example, using the CMOS VLSI Design textbook by Weste and Harris, the method of collecting SNM data and the expected values shown were used as a reference when running my own simulations.

While I focused a lot on a single 6T SRAM cell when collecting data, I believe that this gives a fantastic representation of how a larger memory structure functions. Creating the schematic and layout for a 4x4 memory was the ultimate goal and I accomplished this while also being able to highlight what any given read/write/or hold operation would look like for a single cell.

In the future I want to look at more efficient designs of SRAM arrays (focusing on compact layouts and better noise margins for a start) while expanding on the simulations that were on here. There is still lots of room for improvement and I hope I can continue to grow on what I have built.

References

[1] N. H. E. Weste and D. M. Harris, *CMOS VLSI design: a circuits and systems perspective*, 4th ed. Boston, MA: Addision-Wesley, 2011.

[2] P. Avula, V. Chary, and G. Sreenivasulu, "Improvement of SRAM stability using read and write assist techniques", *International Journal of Engineering and Technology*, Aug. 2017

[3] D. M. Kwai, "SRAM Cell current in low leakage design", presented at the IEEE Workshop on Memory Technology, Design, and Testing, Nov. 12, 2015

[4] K. Itoh, *VLSI memory chip design*, 1st ed. New York, NY: Springer, 2001.

[5] J. P. Kulkarni (2008), SRAM-Basics [Online PDF], Available: https://engineering.purdue.edu/~vlsi/courses/ee695kr/s2008/Lecture4.pdf