

CAREER EPISODE 2

INTRODUCTION

CE 2.1 During a period of 5 months starting from [REDACTED], I worked on a project titled 4x4 6T Static Random-Access Memory (SRAM) with 2 to 4 Decoder Design in the vibrant academic setting of [REDACTED]. Under the guidance of my supervisor, [REDACTED], I explored the complexities of CMOS VLSI Design. This experience at [REDACTED] not only broadened my technical knowledge but also improved my skills in conceptualizing and executing complex design methodologies in the field of electronics engineering.

BACKGROUND

CE 2.2 During the 6th semester of my undergraduate studies, I attended a course on CMOS VLSI Design, where I undertook a project on designing a 4x4 two folded SRAM array alongside a 2 to 4 decoder important for addressing the rows of SRAM. This project was important as SRAM, serves as a keystone in strong vital information in contemporary electronic devices like smartphones and laptops. I delved into the field of transistor level design using CMOS configurations. I managed the entire project, starting with the schematic design. I translated it into a corresponding layout. I conducted pre-layout simulations, refined and optimized the design to ensure efficiency. I applied the guidelines outlined in the book CMOS VLSI Design: A circuits and system perspective by Neil H.E Weste and David Money Harris, fourth edition, and conducted post layout simulations.

CE 2.3 To form the foundation of my project, I started research, dived into an in-depth exploration of existing literature and pertinent resources in the field of CMOS VLSI Design. I studied scholarly articles to gain insights into established methodologies, innovative design strategies and prevalent challenges in the domain of SRAM design. I planned the project, outlined its milestones, divided individual responsibilities among my team members and created a timeline. I had discussions with each team member to ensure that each member felt invested in my shared vision. I prepared the project report and presentation to display my work.

PROJECT REPORTING HIERARCHY

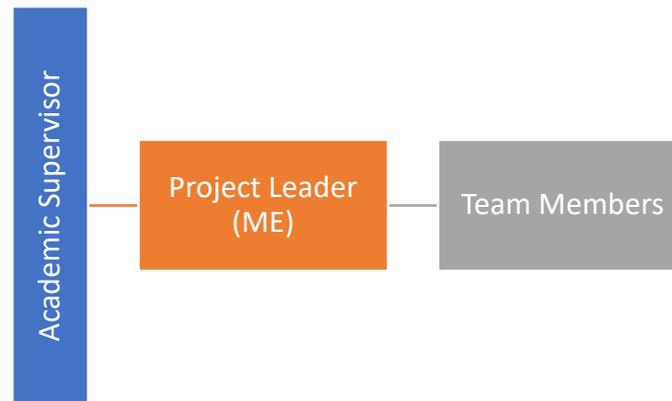


Figure 1: Hierarchy

PERSONAL ENGINEERING ACTIVITIES

- CE 2.4** I started my project by outlining the design overview, charting a sequential roadmap for execution guided by a block diagram. My main focus was on the complex design of a 2 to 4 decoder which was an element responsible for raising each of the 4 wordlines at a time depending on the use. I designed a column circuitry for column multiplexing and finally bitline conditioning circuitry which was responsible for raising the bitline or the complement of the bit line which passes by each cell in the RAM array. I broke down the project into essential phases; for the design of the 2-to-4 decoder, I estimated the number of stages and the sizes of transistors in each stage, delved into the schematic design of the decoder and pre-layout simulations, integrated lock and sleep circuitry for operational robustness, translated the schematic into a layout with post-layout simulations and scrutinized the compliance through essential Design Rule Checks (DRCs) and Layout Versus Schematic (LVS) checks.
- CE 2.5** In the design of 4x4 SRAM array, I created column circuitry schematic, including the write drivers and bitline conditioning mechanisms, while designing the schematic and pre-layout simulations for the SRAM cells. I structured the SRAM layout for coherence and optimal arrangement. In the final stages, I combined SRAM array and decoder performing necessary DRC and LVS checks for integration and conformity with standards. I performed layout simulations and verified the design against predefined expectations and performance benchmarks.
- CE 2.6** In decoder design phase, I implemented the specified truth table logic into a functional 2 to 4 decoder circuit. The operation of decoder was dictated by the inputs A0 and A1, translating into different wordline activations. To achieve this translation, I applied AND +NOT gate logic, using NBAND gates followed by NOT gates to realize the specified functionality. This

choice facilitated the optimization of parasitic delays in the decoder circuitry. The use of this logic configuration was aimed at synchronizing the operation of decoder with the global clock of the system.

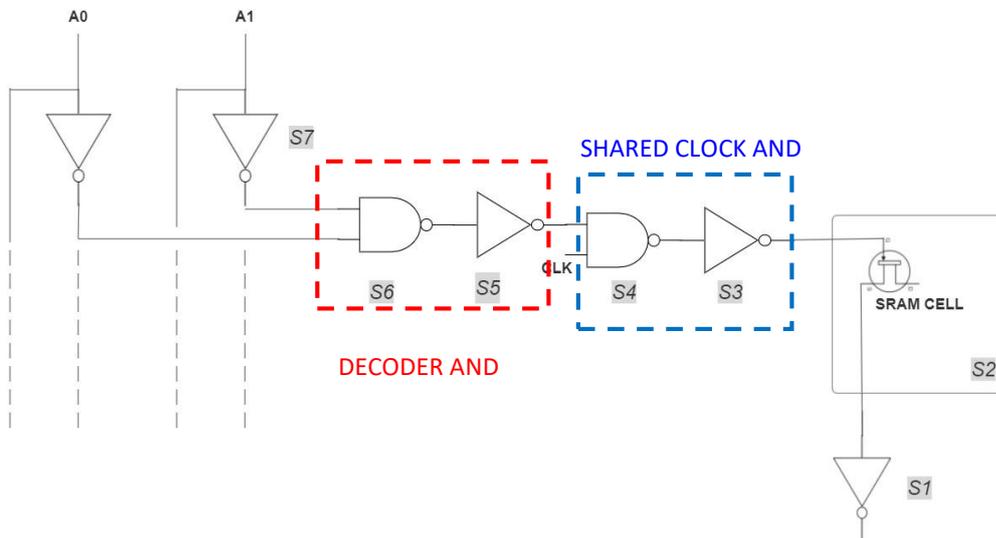


Figure 2: Row circuitry for a single word line indicating the total number of stages

CE 2.7 I performed several calculations and estimations for optimal functionality and efficiency. The estimation of the worst-case path delay was main in determining the ideal number of stages and the type of logic gates for the design. Using equations derived from the reference book, I estimated the path logical effort (G), branching effort (B) and path electrical effort (H) to ascertain the worst-case path delay which was calculated to be $64/3$. From this, I calculated the best number of stages as 2.2, rounded to 2. However, the topology already had more than 2 stages. Therefore, in my design the objective was minimizing the number of stages for good performance.

$$F = GBH$$

$$\text{Path logical effort, } G = \frac{n+2}{3 \times 2}$$

$$\text{Branching effort, } B = N = 2^{n+m}$$

$$\text{Path electrical effort, } H = \frac{1}{2};$$

$N = \text{number of array cells}$
(n rows and m columns)

$$\text{Best number of stages} = \log_4 F = 2.2$$

CE 2.8 I determined the path delay (D), which was calculated as 13.5τ , integrating parameters like the number of array cells and their respective rows and columns. I then, estimated load capacitance and input capacitance at each stage. The total load capacitance for each decoder line was estimated at $12C$, considering the sizes of transistor and wire capacitance, aiming

for accurate calculations and optimal stage by stage load analysis. Calculations for achieving minimum path delay included stage effort evaluations for each gate. This process required distinct stage counts for the first three rows 7 stages and the last row 6 stages, factoring in the bypass of the initial NOT gate. These calculations, resulted in the breakdown of transistor size for each stage, reflecting an understanding of the interplay between stage effort, capacitance and gate dimensions. I rounded off and adjusted values to achieve final sizes for each stage. I determined NMOS and PMOS transistor sizes for each stage which included special cases for the SLEEP and CLK stages.

$$\begin{aligned}
 D &= 4\log_4 F + P \\
 &= 2(m + n) + 4\log_4 \left[\frac{n + 2}{3} \right] + n + \frac{2^{n+1}}{3} \\
 &= 13.5\text{C}
 \end{aligned}$$

$$\hat{f} = \frac{1}{F \text{Number of stages}}$$

$$C_{in} = \frac{gxC_{out}}{\hat{f}}$$

For a 7-stage system:

- Cout_S3 = 7.74C
- Cout_S4 = 6.66C
- Cout_S5 = 4.30C
- Cout_S6 = 3.70C
- Cout_S7 = 2.38C

Similarly, for a 6-stage system:

- Cout_S3 = 7.22C
- Cout_S4 = 6.21C
- Cout_S5 = 4C
- Cout_S6 = 3.44C

S7: NMOS = 0.8um, PMOS = 1.6 μm

S6: NMOS = 1.7um, PMOS = 1.7 μm

S5: NMOS = 1.1um, PMOS = 2.2 μm

S4: NMOS = 2.6um, PMOS = 2.6 μm

S7: NMOS = 1.2um, PMOS = 4.8 μm

SLEEP PMOS = 4.8 μm

CLK NMOS = 2.6 μm

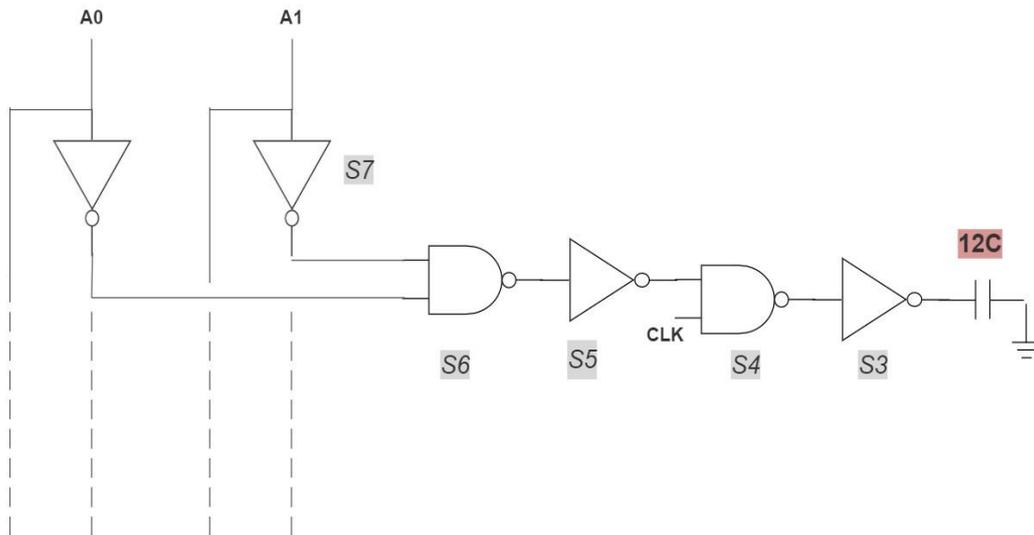


Figure 3: Load capacitance estimation for one of the word lines

CE 2.9 I used Cadence software, utilized its XC06 technology, specifically a 0.6 μm CMOS process to implement the decoder design. I studied the datasheet of XC06 technology to understand the available libraries and layout layers provided by it. I drew the topology in the schematic environment of Cadence. The layout configuration mirrored that of the stick diagram to ensure compatibility for a design that matched the height of the SRAM cells and provided the common utilization of VDD and GND rails for adjacent wordlines. The resulting layout showcased the representation of all four rows and a magnified view of the initial two rows. Following the layout implementation, I executed DRCs. These checks verified the adherence of layout to the layout laws specified by the XC06 technology for accuracy and precision important for subsequent fabrication. I rectified all identified DRC errors for compliance with layout regulations. I conducted LVS checks to validate the absence of electric disparities between the schematic and the layout representations.

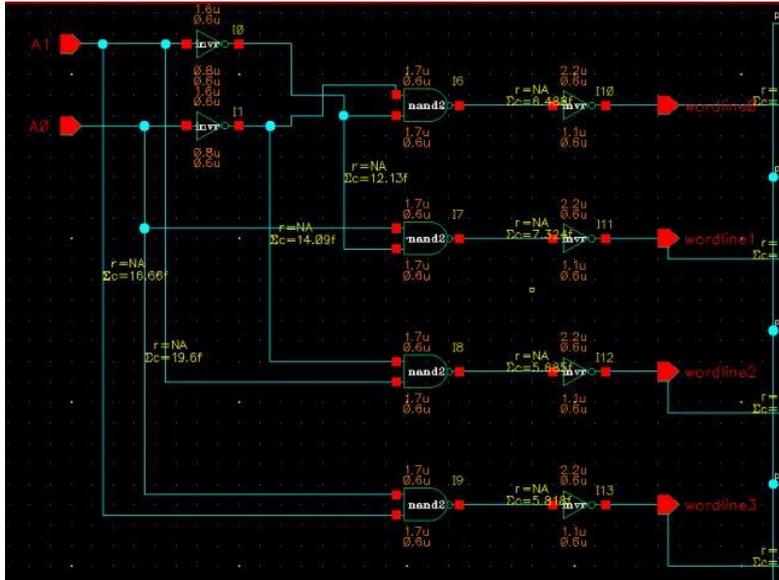
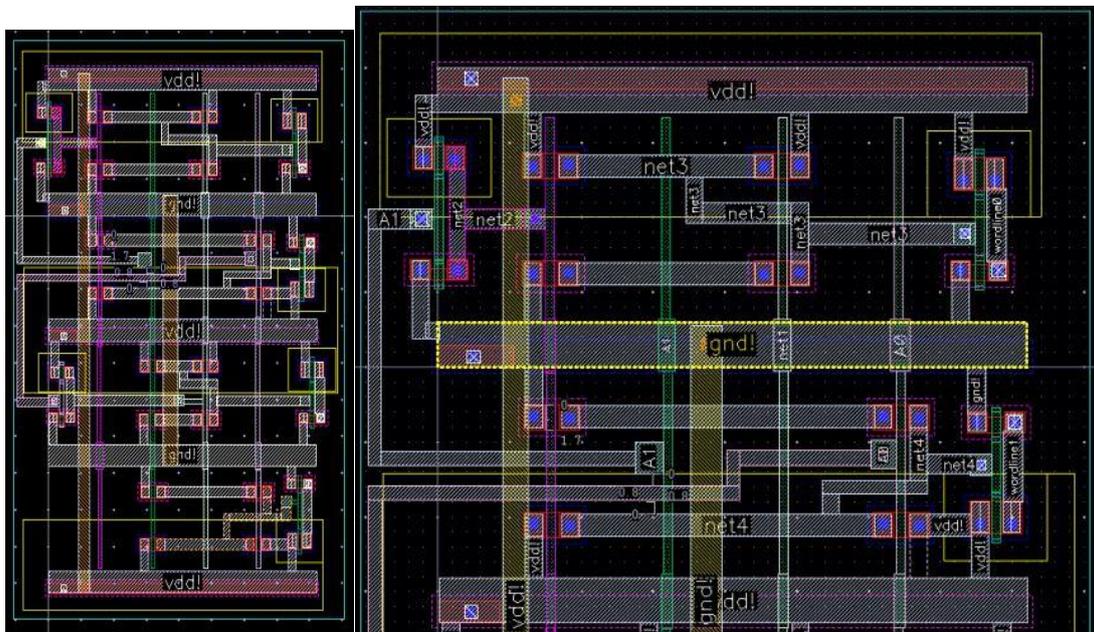


Figure 4: Decoder schematic using Cadence



(a) All 4 rows

(b) Zoomed in version of first 2 rows

Figure 5: Layouts (in Cadence) corresponding to Figure 4

CE 2.10 I expanded the schematic by including the essential clock and sleep signal circuitry, guided by the schematic reference from my book. Using the pre-determined transistor sizes, I integrated this additional circuitry into the existing decoder schematic in the Cadence platform. This process required an accurate alignment of the supplementary components to maintain operational coherence and compatibility with the function of decoder. Moving from the schematic improvement, I delved into manual layout process in Cadence to translate the augmented schematic into a tangible layout, depicting the added clock and

sleep signal circuitry. Following integration, I executed rounds of DRCs and LVS checks with each step aimed at validating the accuracy and compliance of the expanded layout with XC06 technology standards. Through iterative validations, I ensured error free results.

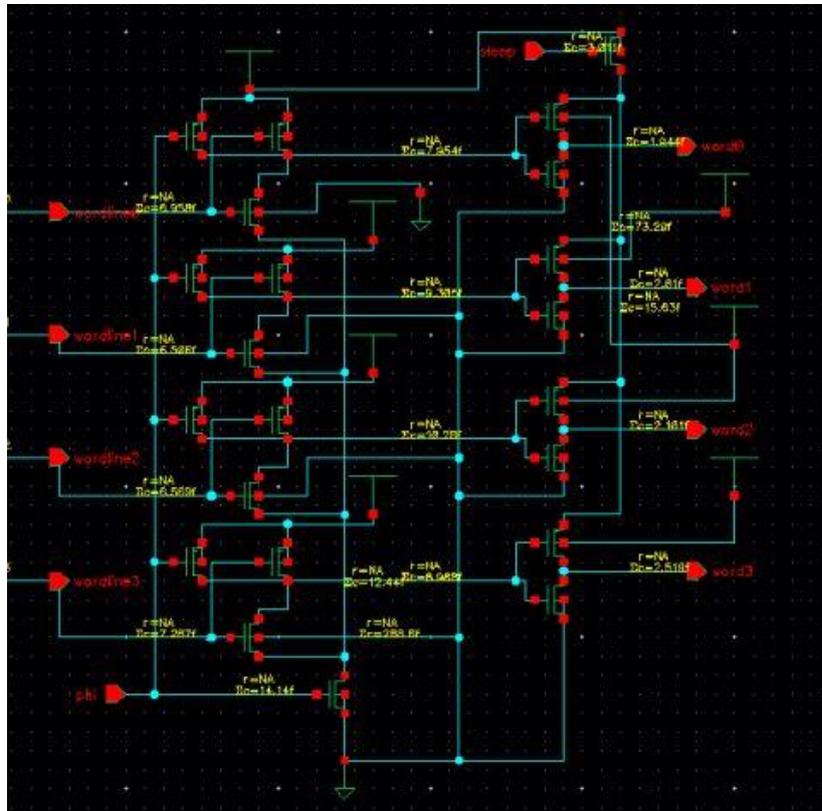


Figure 6: Clock and sleep signal circuitry in Cadence

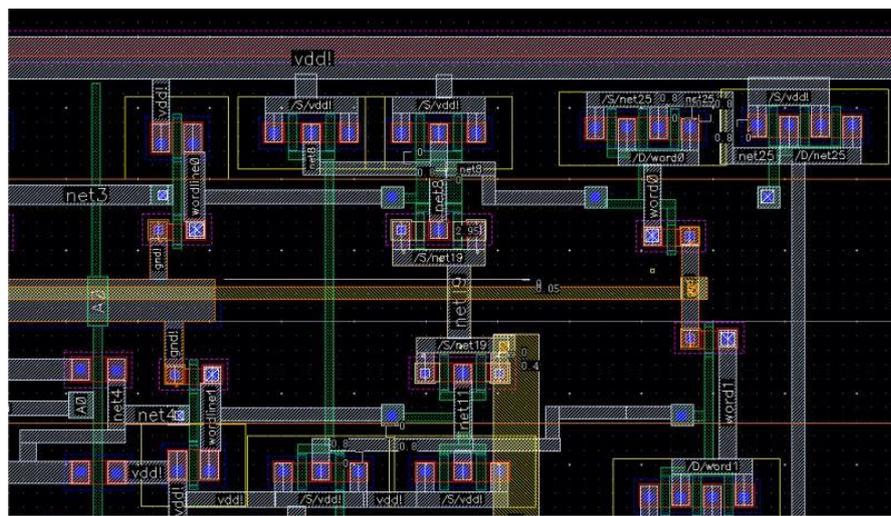


Figure 7: Layout of first two rows corresponding to Figure 6 in Cadence

CE 2.11 In the completion phase, extracting parasitic capacitors from the complete decoder layout was important for further analysis. I used the extracted file to perform the post layout simulations using a designated test bench. These simulations scrutinized the propagation

delay and power consumption which were important in evaluating the performance of decoder. With the clock signal (ϕ) held high and the sleep signal consistently low, I simulated a 12C output load using NMOS width of $3.2\ \mu\text{m}$ and PMOS width of $6.4\ \mu\text{m}$. Manually cycling different combinations of A0 and A1 bits confirmed the accurate activation of wordlines as expected. This analysis revealed detailed insights into propagation delay and power consumption across each wordline. The total area of the decoder was also calculated to be $8470\ \mu\text{m}^2$ derived from accurate measurements of $78.35\ \mu\text{m} \times 108.1\ \mu\text{m}$.

CE 2.12 Transitioning to the design phase of the two folded SRAM array, there were to be 8 words in total with 2 bits in each word. Using NMOS pass transistors and NOT gates, I created the column circuitry within the Cadence environment. The multiplexing was done such that if A2 is low, *column 0* and *column 2* are accessed; while if it is high, *columns 1* and *3* are accessed. The write drivers' circuitry was done together with the column multiplexing circuitry.

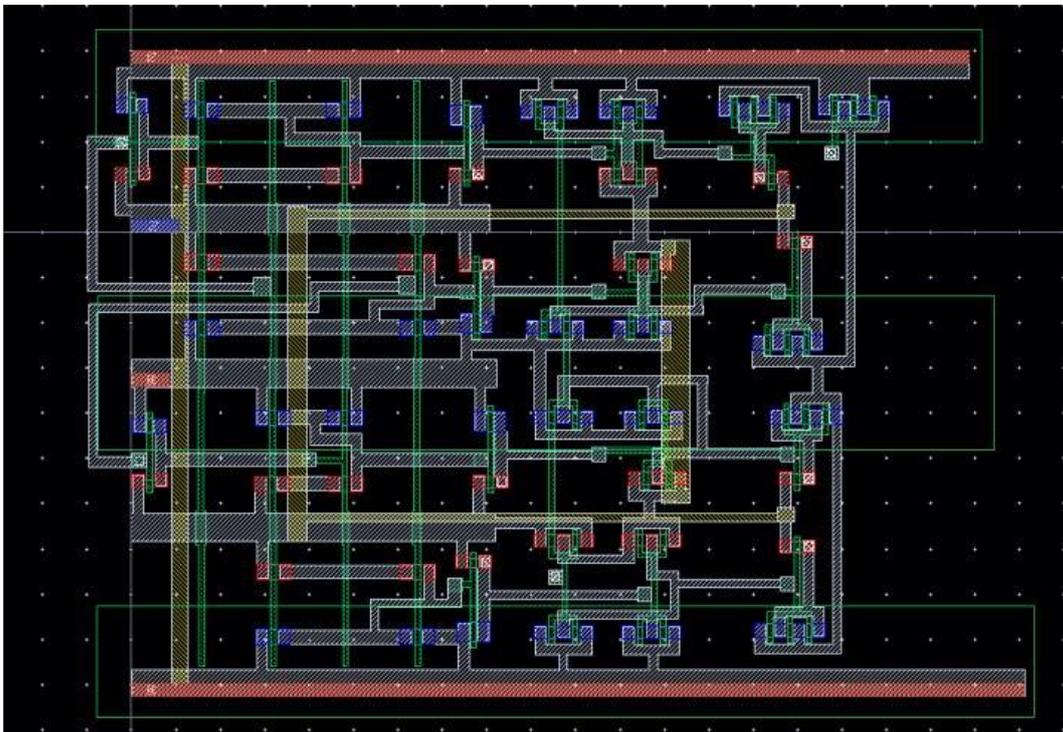


Figure 8: Full layout of 2-to-4 decoder in Cadence

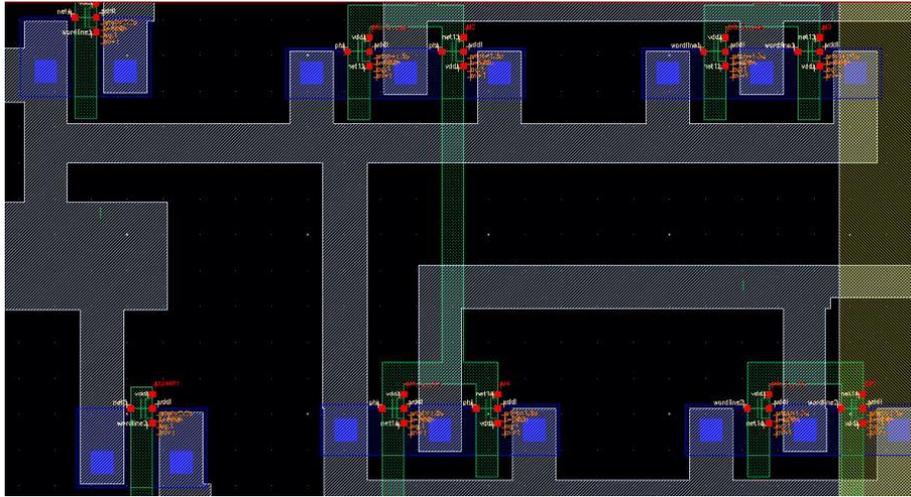


Figure 9: Zoomed in version of extracted parasitic capacitances

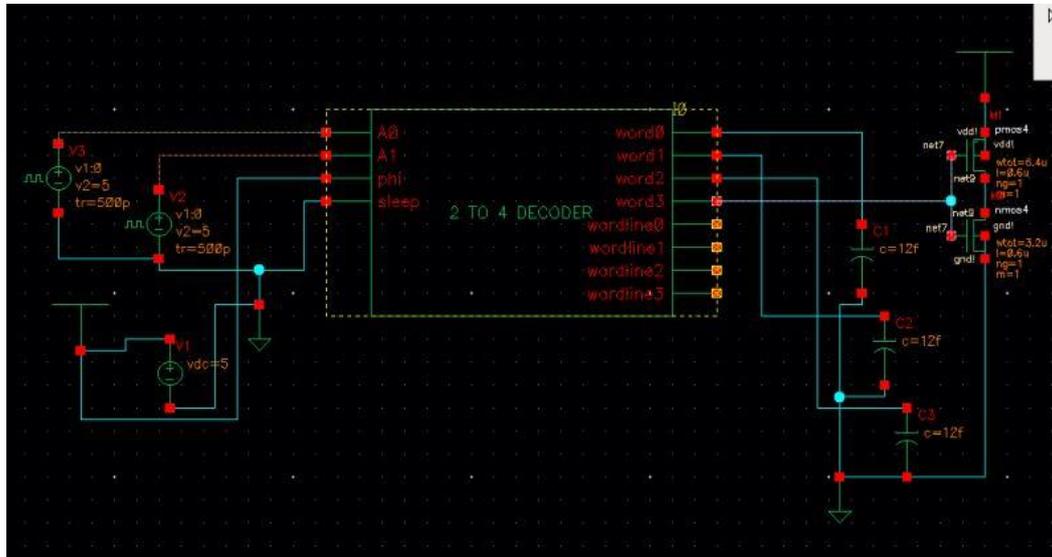


Figure 10: Test bench for post layout decoder simulations

Wordline	Propagation delay	Power consumption
wordline0	1.22 ns	235.2 μ W
wordline1	1.06 ns	210.9 μ W
wordline2	1.07 ns	212.7 μ W
wordline3	745.4 ps	233.7 μ W
Average	1.023 ns	223.15 μW

Table 2: Decoder propagation delay and power consumption

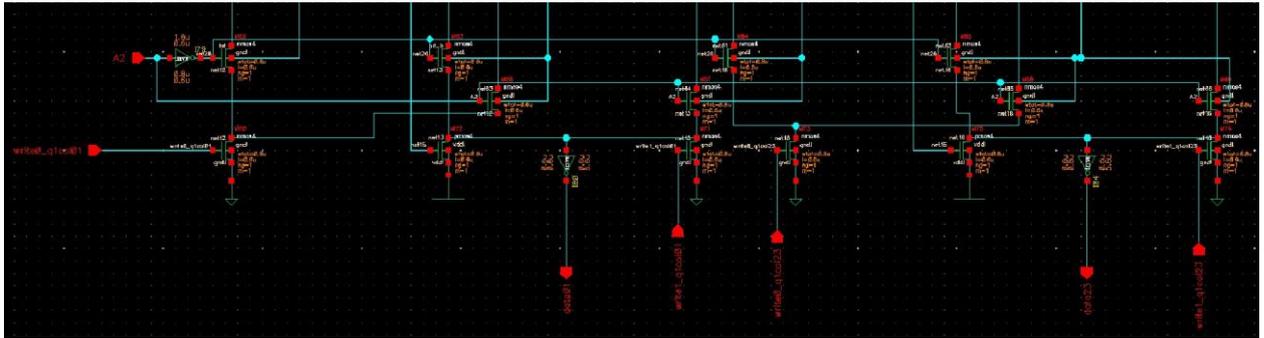


Figure 11(a): All 4 columns of the column circuitry

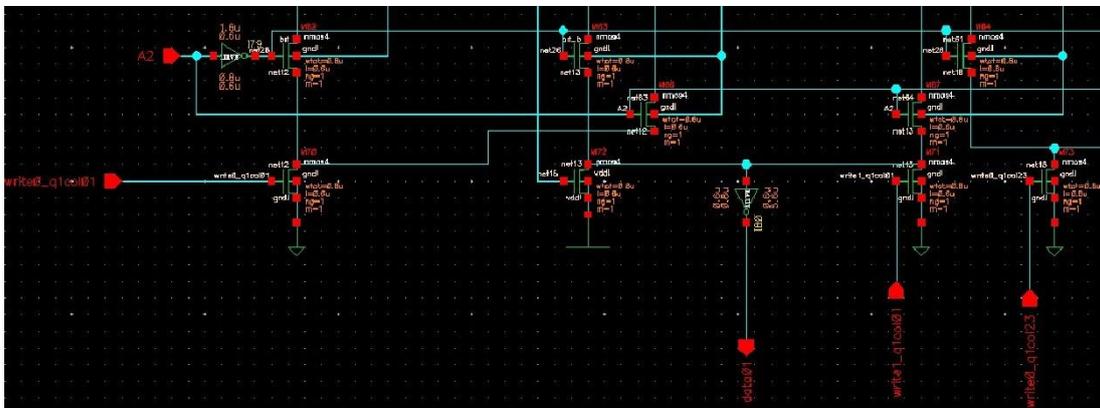


Figure 11(b): First 2 columns of the column circuitry

CE 2.13 In the final stages, I integrated and refined the SRAM array design. I started by using the standard 6T SRAM cell architecture, ensuring adherence to the read stability and writability constraints outlined in the figures from Neil H.E. Weste’s book. I chose the transistor sizes for the SRAM cells based on these constraints; $P1=P2=0.8\ \mu\text{m}$, $A1=A2=1.0\ \mu\text{m}$ and $D1=D2=3.0\ \mu\text{m}$. I created the column write circuitry including bitline conditioning and write driver circuits with all transistors, both NMOS and PMOS set at $0.8\ \mu\text{m}$. Through this choice, I aimed to optimize chip size which was a critical factor influencing production costs in the semiconductor industry. I completed the schematic design of the complete SRAM array, integrated with the decoder in Cadence.

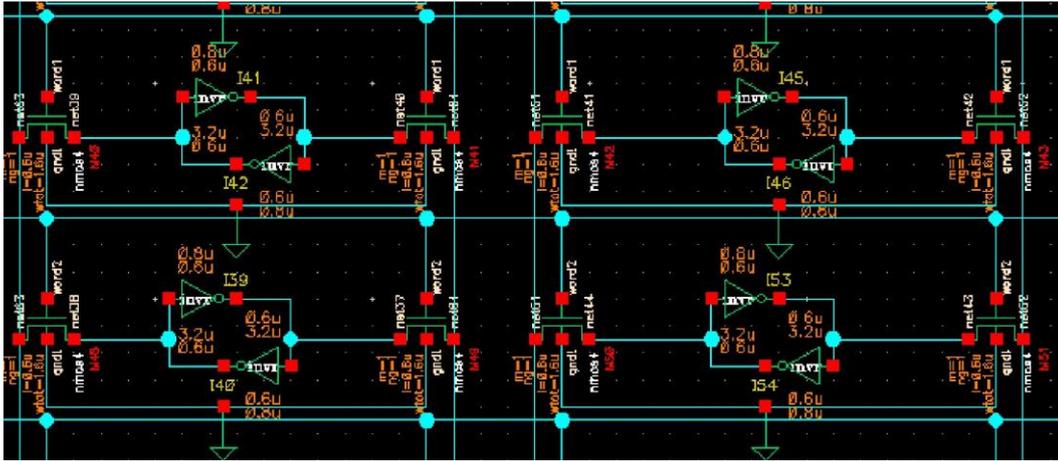


Figure 12 (a): 4 cells of 6T SRAM array schematic in Cadence

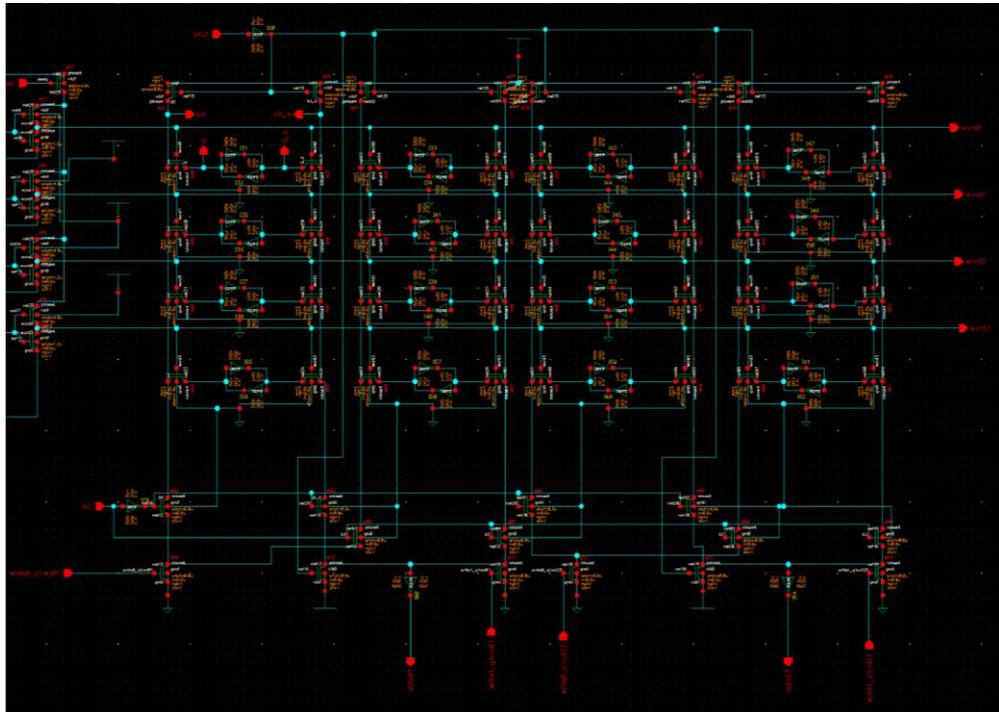


Figure 12 (b): Full 6T SRAM array schematic in Cadence

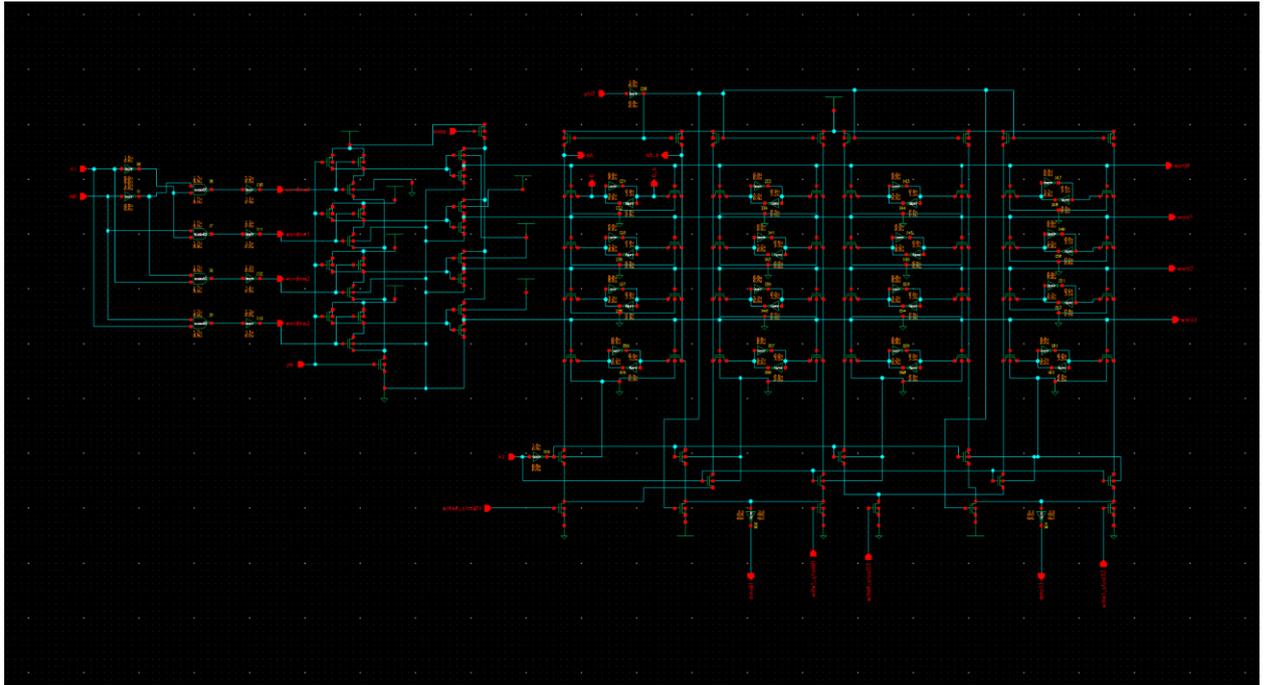


Figure 12 (c): Full schematic including decoder in Cadence

CE 2.14 Moving to the simulation tests, I set up the pre-layout simulations test bench. I conducted thorough checks for read and write operations, gauging responses against expected behavior, validating the read stability and writability constraints of the SRAM array. In the layout phase, I implemented a thin cell layout for the 6T SRAM cells. The manual layout of the complete 4x4 SRAM array was executed in the Cadence environment, capturing a single cell layout and the full array layout. Integrating all layout components which included the decoder and the SRAM array, the full design layout was achieved and analyzed for consistency and accuracy through DRC and LVS checks. I set the clock period at a minimum of 10 ns to determine the maximum operational frequency. I also conducted post layout simulations to account for RC parasitic in the full chip, extracting 1463 resistive and 11434 capacitive parasitics.

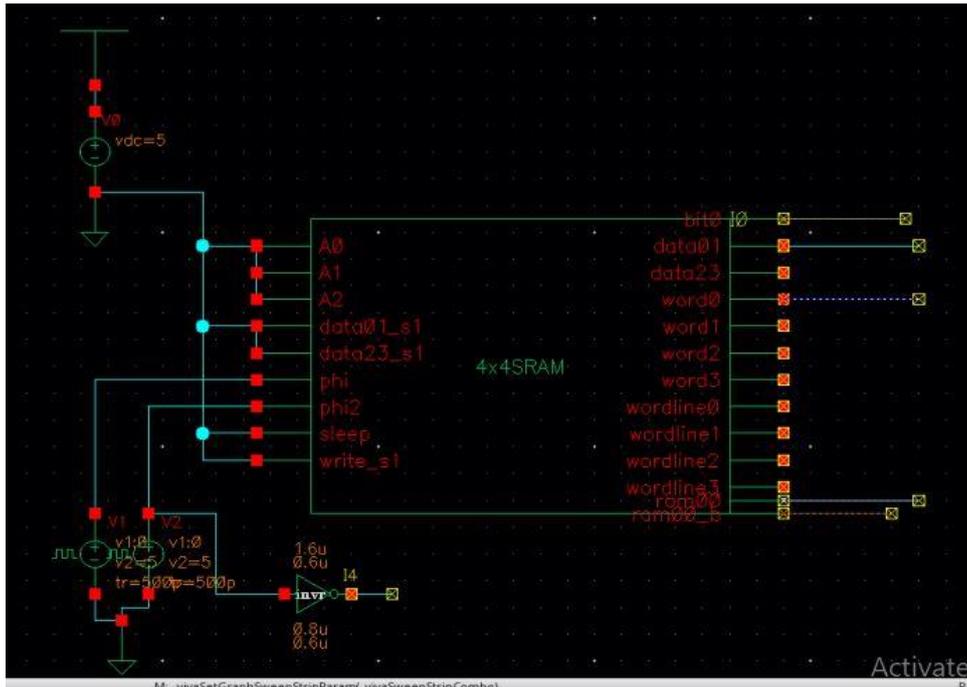
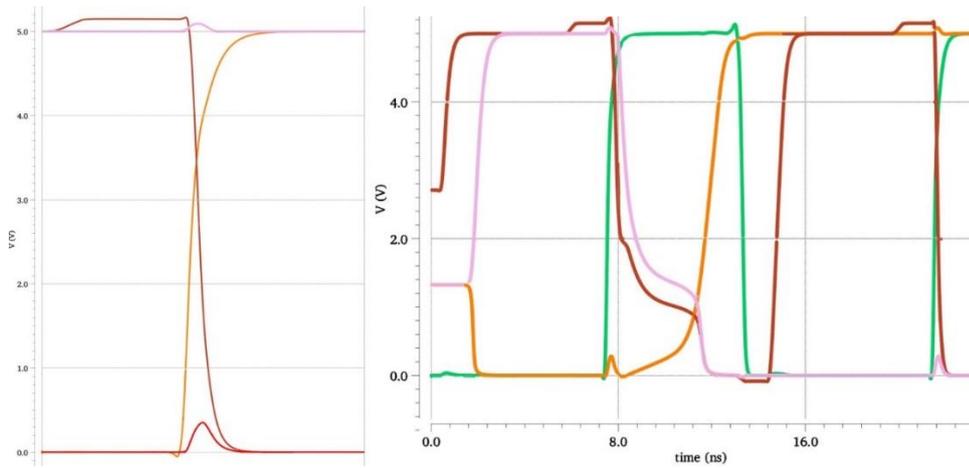


Figure 13: Test bench for pre-layout simulations



a)

b)

Figure 14: (Read (a) and write (b) pre-layout simulation responses)

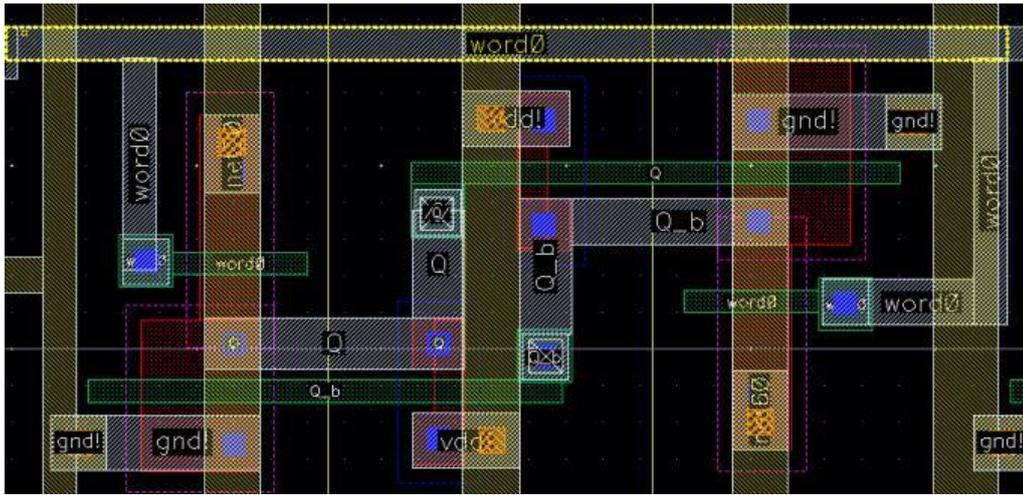


Figure 15(a): One single cell layout of the SRAM array in Cadence

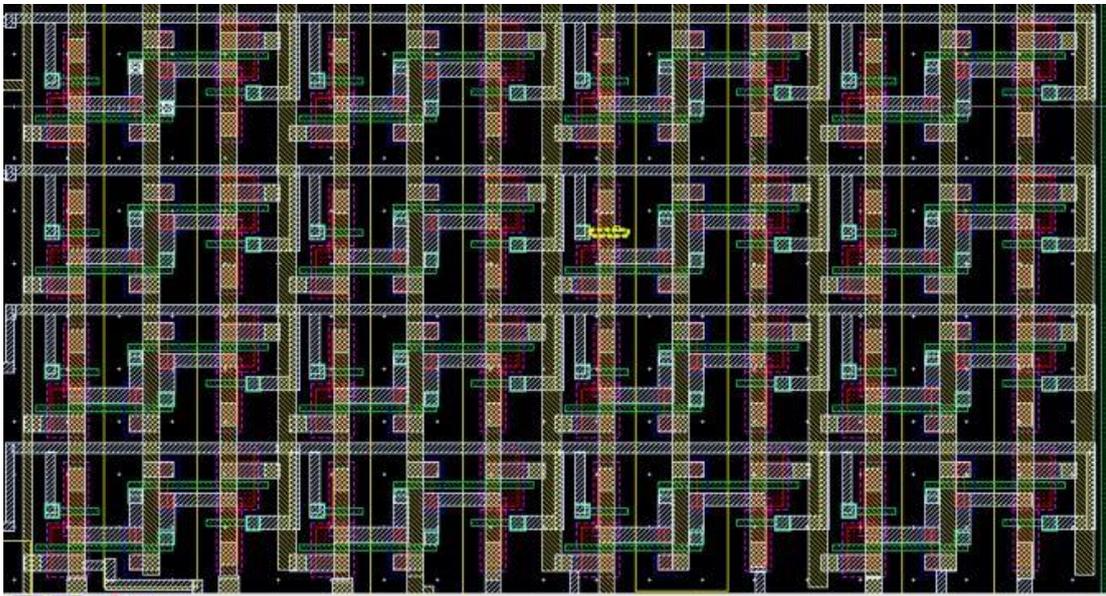


Figure 15(b): Full 4x4 layout of the SRAM array in Cadence

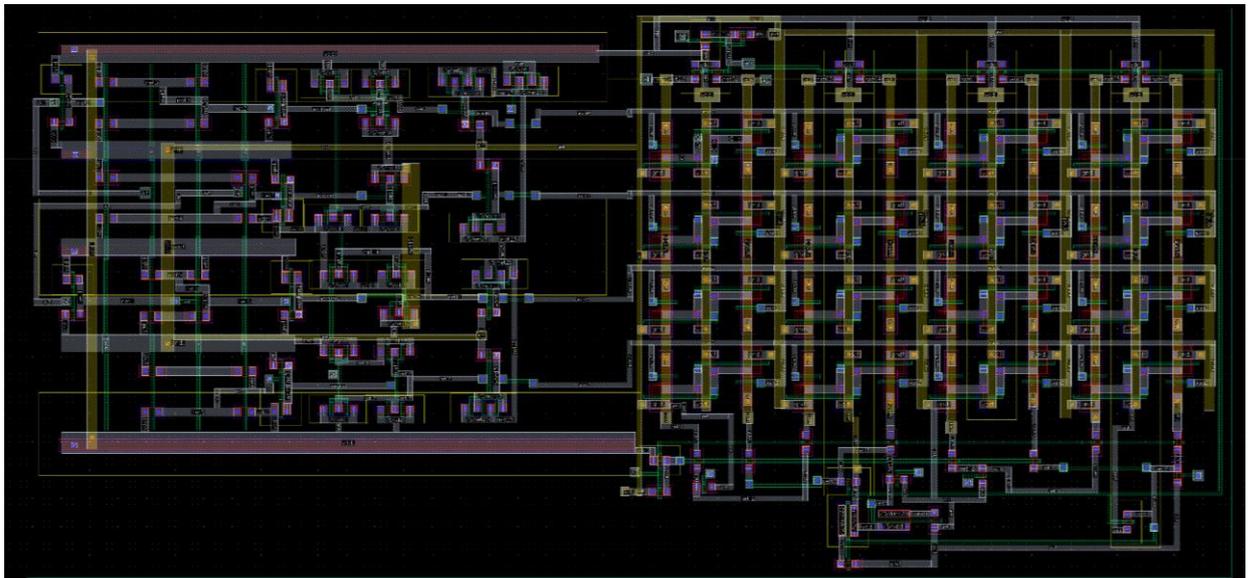


Figure 16: Full layout of the design in Cadence

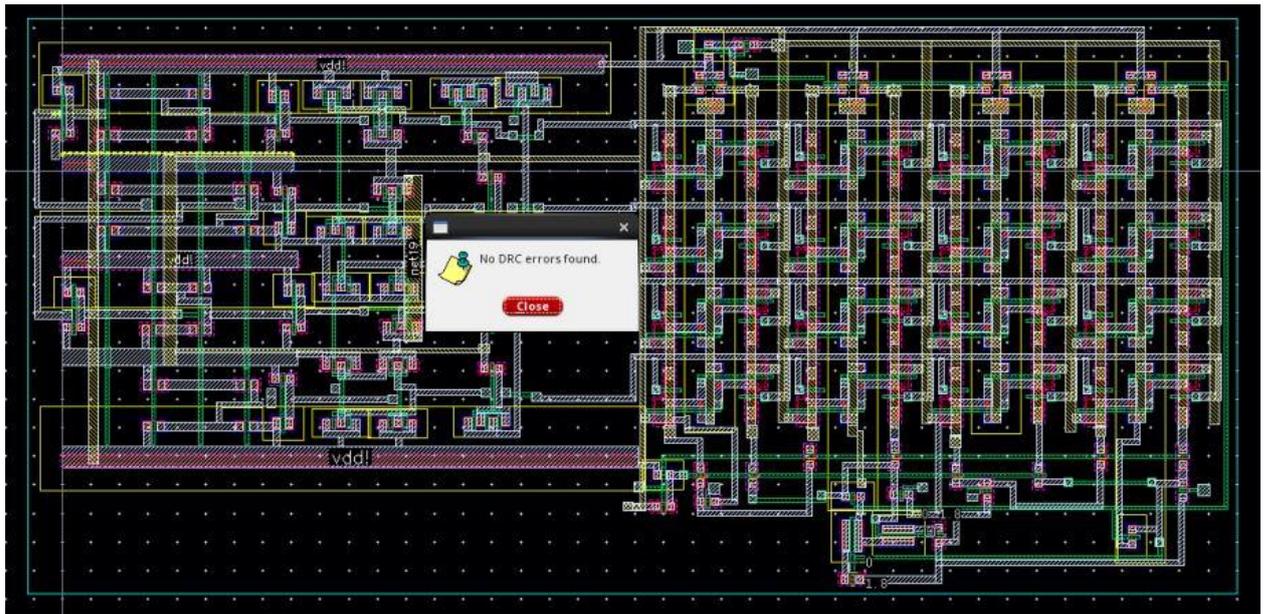


Figure 17 (a): No DRC errors

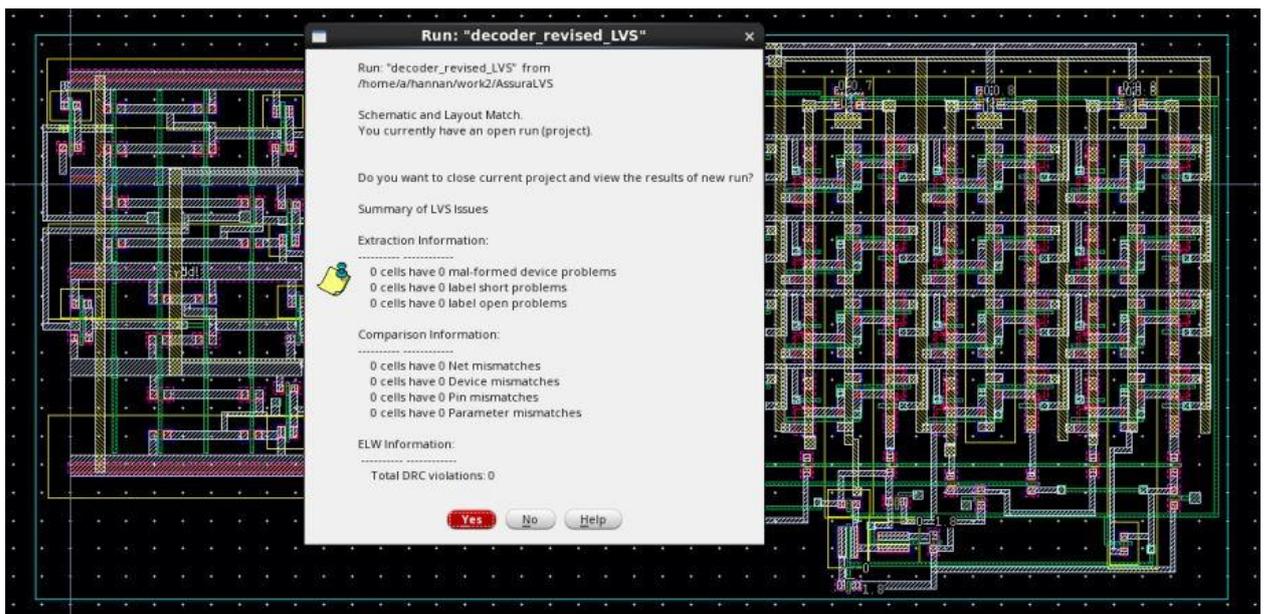


Figure 17 (b): No LVS mismatches

CE 2.15 I generated a new symbol and test bench for post layout simulations. In the simulated operations, the write delay for transitioning a cell from 0 to 2 was measured, resulting in a write delay of 1.02 ns. The read delay for retrieving a 0 from a cell was determined to be 0.28 ns. The determination of the total area of the chip, calculated from the P&R boundary, confirmed a total area of 22253 μm^2 with only 2208 μm^2 remaining unused. This utilization indicated an efficient use of approx. 90% of the chip area. I designed the 4x4 6T SRAM array plus its decoder using the Cadence software. I maintained the cell stability while writing a 1

or reading a 0 to or from a cell respectively. I verified the functionality of the decoder and set up the test benches to mimic real life working by making sure the operations were clock qualified. However, it should also be noted that even though the post layout simulations were performed, the performance of the chip cannot be fully verified before fabricating it and performing actual measurements. In reality, any two chips having the same design may show different performance characteristics owing to fabrication uncertainties and different experimental scenarios. In essence, the conclusion of this project resulted in the design of a 4x4 6T SRAM array integrated with a decoder, emphasizing functionality validation, test bench setups and efficient chip area utilization.

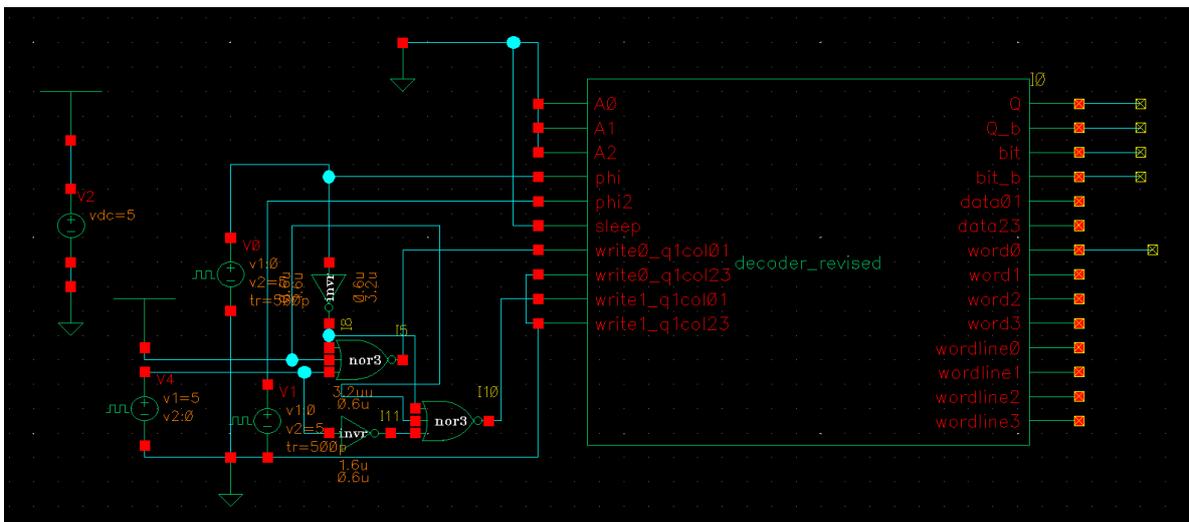


Figure 18: Testbench including symbol for post-layout simulations

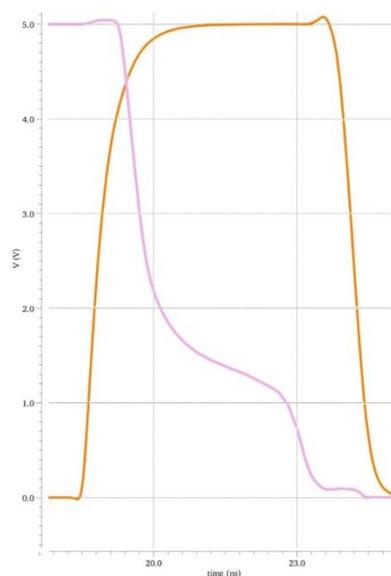


Figure 19: Write operation post layout simulation response

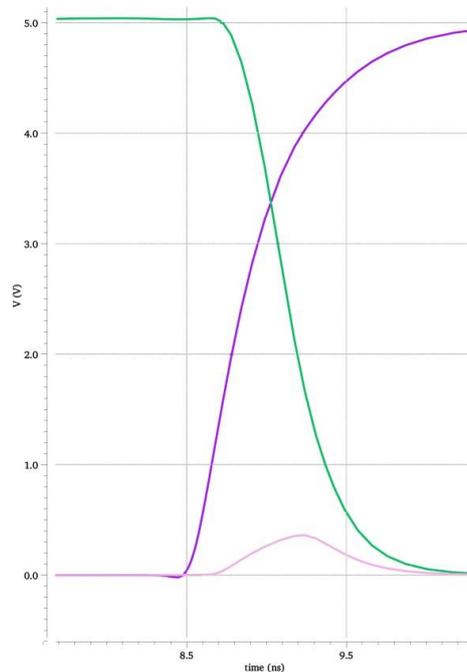


Figure 20: Read operation post layout simulation response

SUMMARY

CE 2.16 I designed and implemented a 4x4 6T SRAM array alongside a 2 to 4 decoder using Cadence software. Starting with planning and topology selection, I designed the schematics and layout design from accurate adherence to stability constraints. I executed simulations, confirming the operational functionality and validating read/ write operations. Post layout simulations further refined the performance insights, yielding a write delay of 1.02 ns and a read deal of 0.28 ns. Design strategies ensured nearly 90% chip area usage and test bench setups. However real-world fabrication remained crucial to fully verify the performance of the chip.