Design of an 8-bit CAM using 9T SRAM cell

Jehad Hamayel - 1200348, Arwa Doha - 1190324, Anas Hawamda - 1202121
Department of Electrical and Computer Engineering, Birzeit University

Abstract **–** We present an optimized 8-bit Content Addressable Memory (CAM) design, leveraging the advantages of 9-Transistor (9T) Static Random-Access Memory (SRAM) cells. Traditional CAMs, with their parallel search capability, are effective in high-speed applications, but suffer from high power consumption (1). The 9T SRAM cell improves upon these aspects, offering enhanced stability and low-power operation (5,7). Our novel 8-bit CAM design incorporates the 9T SRAM structure for better read stability and write ability, crucial for high-speed search operations (5). The full-custom design approach further optimizes layout and performance parameters (4). Simulation results confirm improved speed, reduced power consumption, and increased stability over traditional CAM designs (6,8). This study establishes a pathway towards efficient, low-power, and high-speed CAMs, suitable for data routing and lookup applications.

Keywords— CAM, SRAM, optimization, layout, schematic.

I. Introduction

RAM, or Random Access Memory, is a form of computer memory that enables reading and writing data in any sequence at consistent speeds, regardless of the data's physical position within the memory. Being a volatile memory type, RAM erases all stored data once power is disconnected.

Two primary forms of RAM are Dynamic RAM (DRAM), which requires frequent refreshing, and Static RAM (SRAM), which is faster but more expensive and often used for cache memory. Having more RAM allows for more efficient multitasking and quicker data access, making it a critical component in all computing devices.(9)

Content Addressable Memory (CAM), also known as Associative Memory, has characteristics similar to RAM, such as the ability to read and write data at specific addresses (10). However, CAM adds a unique feature: the ability to search for data. It takes data as input, conducts a parallel comparison with stored memory content, and returns corresponding addresses.

As depicted in Figure 1, if incoming data matches stored data in a memory cell, a match line is activated. Then, an encoded location of the match, using log2 w bits, is produced by an encoder (6). Interestingly, in reading operations, CAM essentially performs the reverse operation of RAM, as shown in Figure 2 (2).

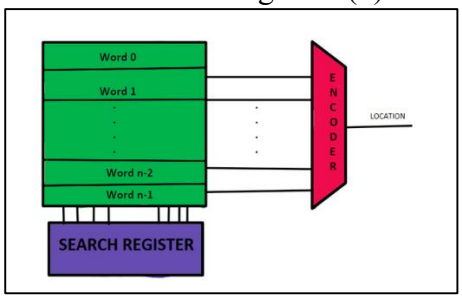

Figure 1 Conceptual view of a CAM.

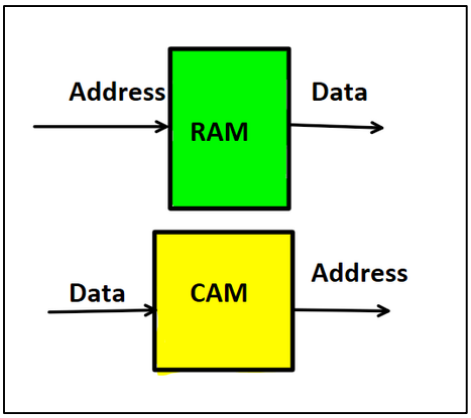

Figure 2 CAM vs RAM.

When it comes to speed, traditional RAM requires multiple clock cycles to locate the address of the desired data during read operations. In contrast, Content Address Memory (CAM) performs the search operation in parallel within a single cycle, resulting in faster search times compared to RAM. This advantage makes CAMs highly suitable for applications such as database management systems and network switching that demand rapid search functionality. However, the increased speed of CAM comes at the cost of larger silicon area and higher power consumption compared to regular RAM. This is due to the additional circuitry required for comparisons and the generation of search and match lines.

The implementation of Content Address Memory (CAM) can utilize the schematic of Static Random Access Memory (SRAM) due to their shared functionality. Moreover, SRAM can be designed with varying transistor configurations, including 6T, 8T, 9T, and 10T, depending on the specific requirements and specifications of the project or application. For our project, we will design and implement CAM using the 9T SRAM configuration.

II. Design and Implementation

In our vision to build the 8-bit CAM using 9T SRAM we build a lot of different components and used them as one block to make it easier for as to make the design and debugging like the second rule of architecture: *"Use Abstraction to Simplify Design"(11)*. And those components are:

A. 9T SRAM

As shown in the following figure, the 9T SRAM circuit has 4 inputs lines, which are the word line (WL), (RL), and other 2 bits line (BL and BLB). While the have two outputs (QB), and (Q).

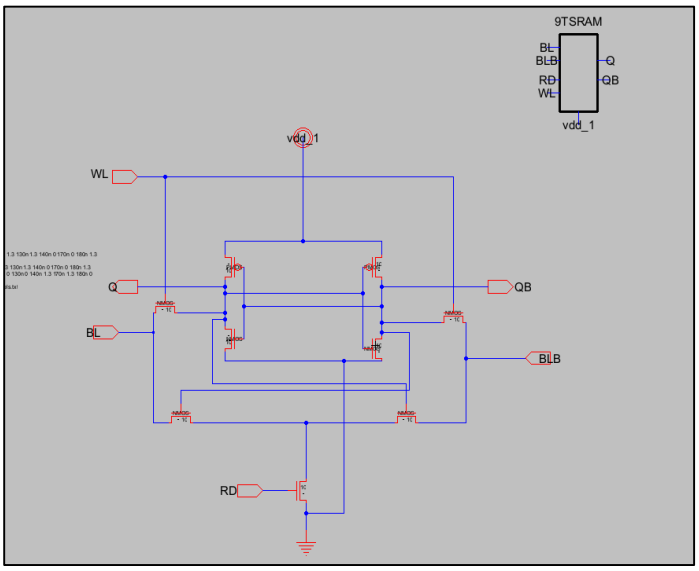

Figure 3: The schematic of 9T SRAM circuit.

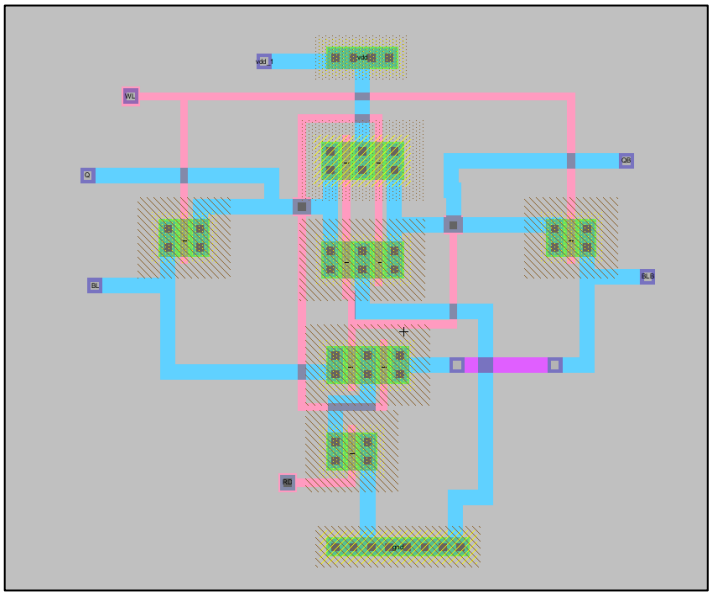

Figure 4: The layout of 9T SRAM

```
vdd vdd 0 dc 2.5
vbl BL 0 pwl 0n 0 10n 0 10.001n 2.5 20n 2.5 20.001n 0 30n 0 30.001n 2.5 40n 2.5 40.001n 0 50n 0
vblb BLB 0 pwl 0n 2.5 10n 2.5 10.001n 0 20n 0 20.001n 2.5 30n 2.5 30.001n 0 40n 0 40.001n 2.5 50n 2.5
vrd RD 0 pwl 0n 0
vwl WL 0 pwl 0n 2.5 30n 2.5 30.001n 0 50n 0
cload out 0 200fF
.tran 50n
.include C:\elec\22nm.txt
```

Figure 5 The code of 9T SRAM

B. 1-Bit CAM

The circuit of a 1-bit CAM was designed as shown in the figures below we added two inverters and two pass gates so we can di a comparison operation and at the end of the circuit the output represent the matching signal.

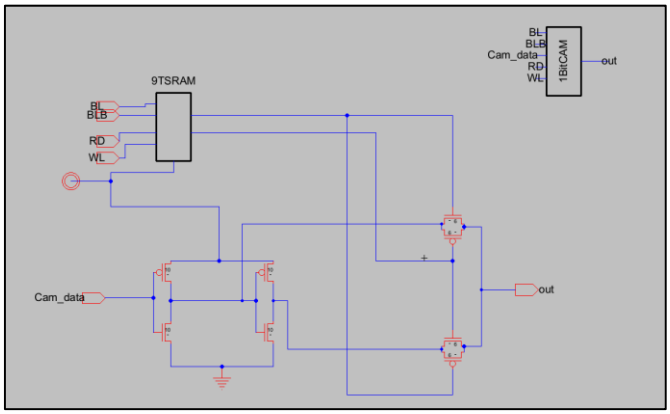

Figure 6: The schematic of 1-Bit CAM circuit.

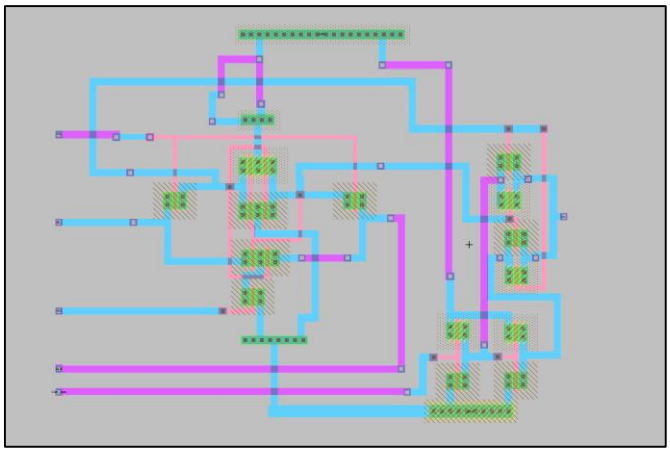

Figure 7: The layout of the 1-bit CAM

```
vdd vdd 0 dc 2.5

vblb BLB 0 pwl 0n 0 10n 0 10.001n 2.5 20n 2.5 20.001n 0 30n 0 30.001n 2.5 40n 2.5 40.001n 0 50n 0
50.001nn 2.5 60n 2.5 60.001n 0 70n 0 70.001n 2.5 80n 2.5 80.001n 0 90n 0 90.001n 2.5 100n 2.5

vbl BL 0 pwl 0n 2.5 10n 2.5 10.001n 0 20n 0 20.001n 2.5 30n 2.5 30.001n 0 40n 0 40.001n 2.5 50n 2.5
50.001nn 0 60n 0 60.001n 2.5 70n 2.5 70.001n 0 80n 0 80.001n 2.5 90n 2.5 90.001n 0 100n 0

vrd RD 0 pwl 0n 0 30n 0 30.001n 2.5 40n 2.5 40.001nn 0 60n 0 60.001n 2.5 100n 2.5

vwl WL 0 pwl 0n 2.5 20n 2.5 20.001n 0 40n 0 40.001nn 2.5 60n 2.5 60.001n 0 100n 0

vcam Cam_data 0 pwl 0n 2.5  100n 2.5

cload out 0 200fF

.tran 100n

.include C:\elec\22nm.txt
```

Figure 8: The code of the 1-bit CAM

C. 3x8 Decoder

In out 8-bit CAM we need the 3x8 Decoder so we can use it to select the right or the appropriate 1-bit CAM cell from the 8-bits for the search operation the below figures show the schematic and the layout of the decoder we designed:

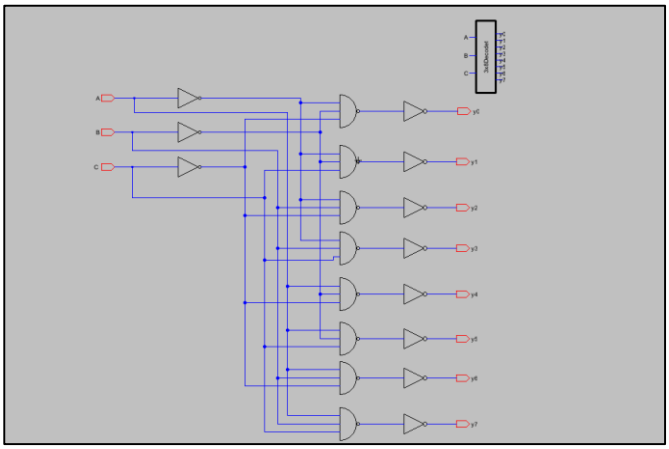

Figure 9: The schematic of the 3x8 Decoder.

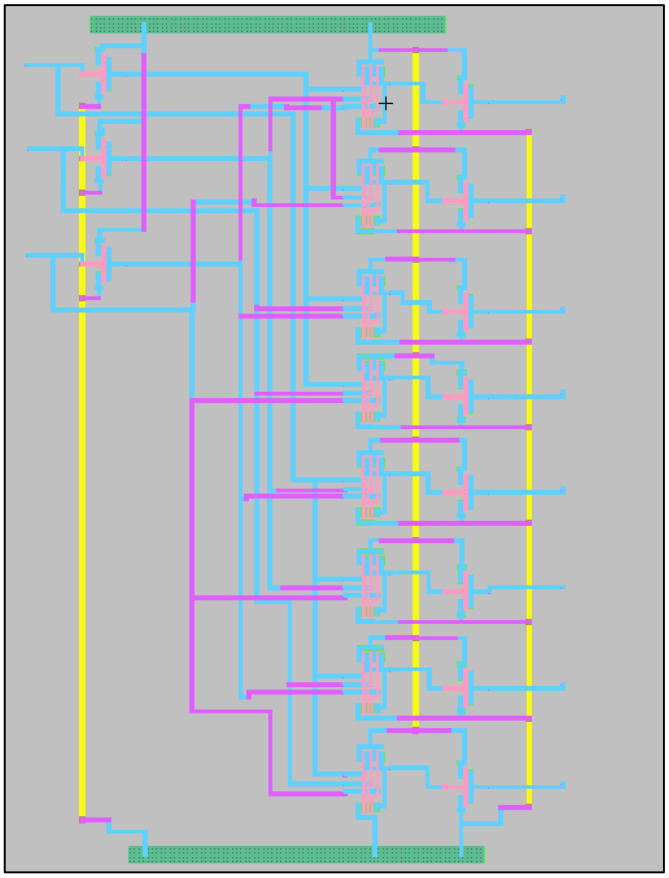

Figure 10: The layout of the 3x8 Decoder.

```
vdd vdd 0 dc 1.3
va A 0 pulse 1.3 0 1p 1p 1p 150n 300n
vb B 0 pulse 1.3 0 1p 1p 1p 300n 600n
vc C 0 pulse 1.3 0 1p 1p 1p 600n 1200n
cload out 0 200fF
.tran 2400n
.include C:\elec\22nm.txt
```

Figure 11: The code of the 3x8 Decoder.

D. 8-inputs NAND gate

This circuit will be used in the implementation of the 8-bit CAM and that when connecting the 8 blocks of the 1-bit CAM cell into the output the figures below show the design in both schematic and layout.

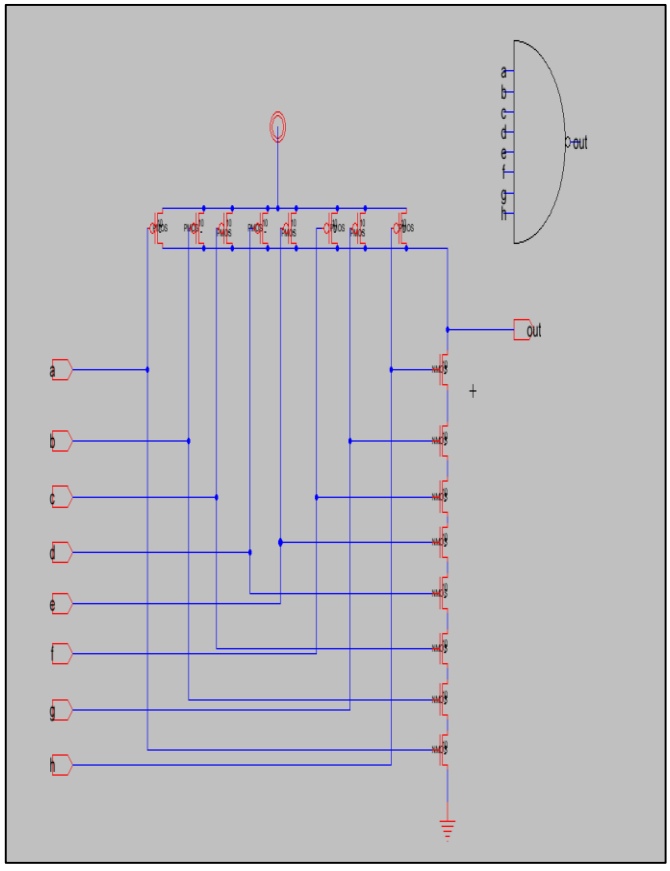

Figure 12: 8-inputs NAND gate schematic.

```
vdd vdd 0 dc 1.3
va a 0 pulse 1.3 0 1p 1p 1p 150n 300n
vb b 0 pulse 1.3 0 1p 1p 1p 300n 600n
vc c 0 pulse 1.3 0 1p 1p 1p 600n 1200n
vd d 0 pulse 1.3 0 1p 1p 1p 1200n 2400n
ve e 0 pulse 1.3 0 1p 1p 1p 2400n 4800n
vf f 0 pulse 1.3 0 1p 1p 1p 4800n 9600n
vg g 0 pulse 1.3 0 1p 1p 1p 9600n 19200n
vh h 0 pulse 1.3 0 1p 1p 1p 19200n 38400n
cload out 0 200fF
.tran 76800n
.include C:\elec\22nm.txt
```

Figure 13: 8-inputs NAND gate code.

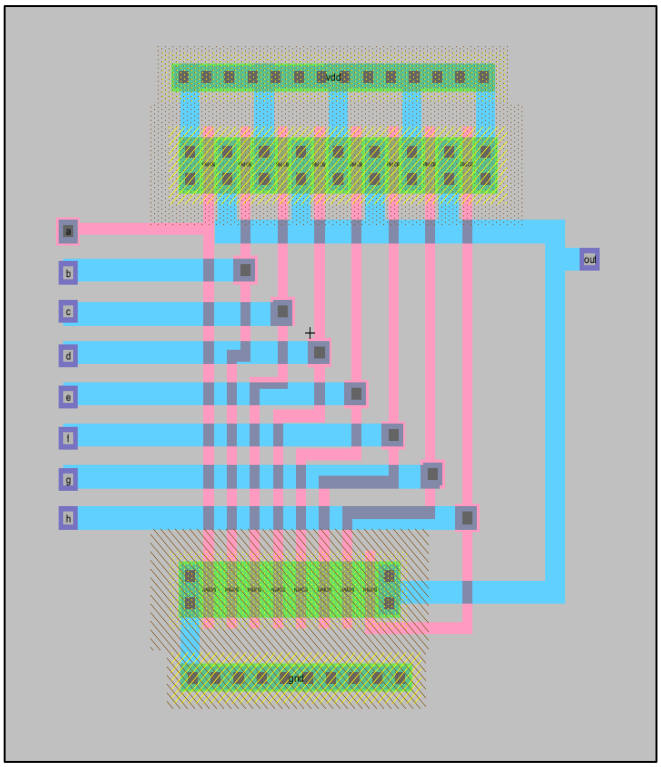

Figure 14: 8-inputs NAND gate layout.

E. 3-inputs NAND gate

In this circuit, it was employed for the purpose of designing a larger circuit intended for incorporation into the 8-bit Content Addressable Memory (CAM). The utilization of this circuit facilitates enhanced levels of abstraction during the debugging phase as well as during simulation processes. Below the schematic and the layout:

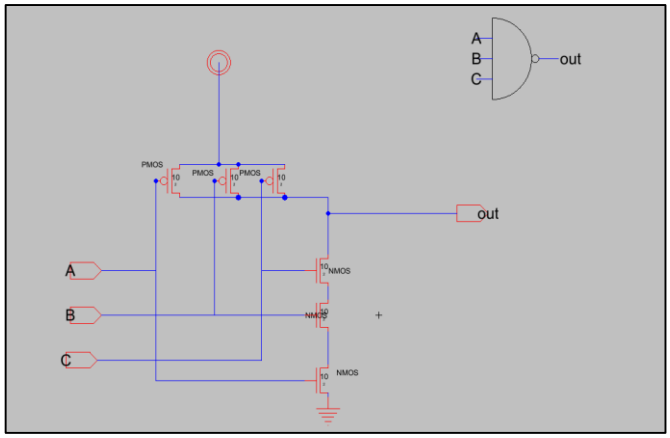

Figure 15: 3-inputs NAND gate schematic.

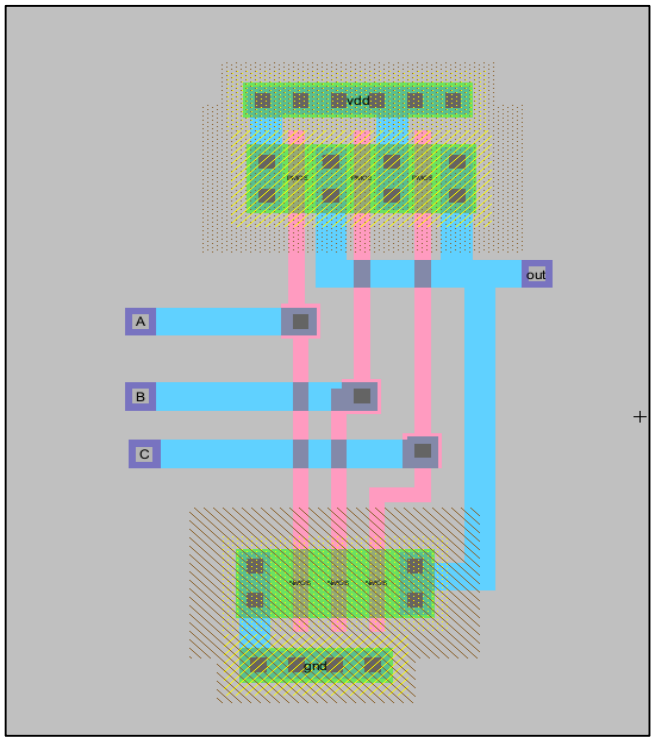

Figure 16: 3-inputs NAND gate layout.

```
vdd vdd 0 dc 1.3
va A 0 pulse 1.3 0 1p 1p 1p 150n 300n
vb B 0 pulse 1.3 0 1p 1p 1p 300n 600n
vc C 0 pulse 1.3 0 1p 1p 1p 600n 1200n
cload out 0 200fF
.tran 2400n
.include C:\elec\22nm.txt
```

Figure 17: 3-inputs NAND gate code.

F. Invertor

Finally, the last component we designed was the inverter the figures below show the schematic and the layout of it:

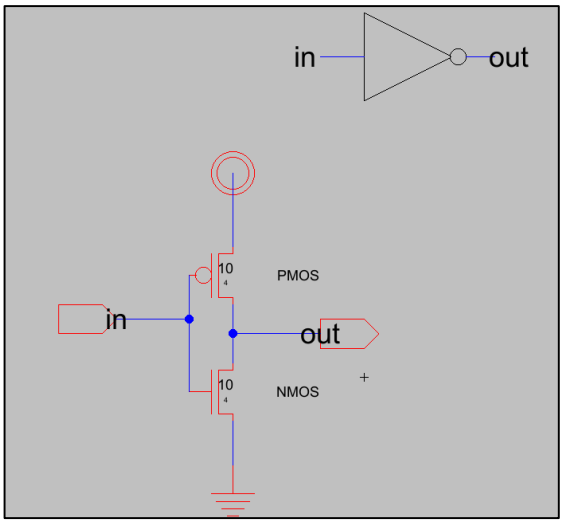

Figure18: Invertor schematic.

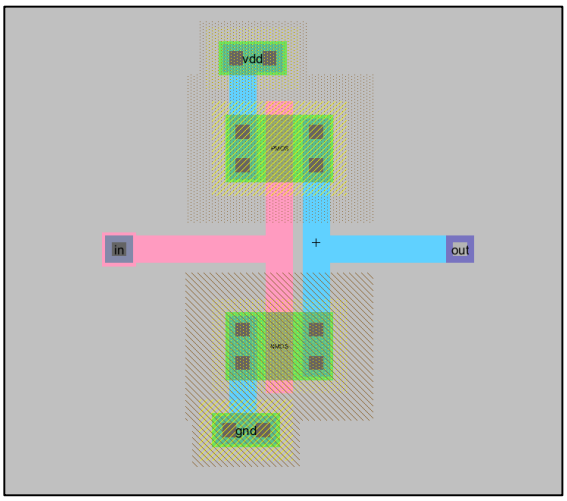

Figure19: Invertor layout.

```
vdd vdd 0 dc 1.3
vin in 0 pulse 1.3 0 1p 1p 1p 300n 600n
cload out 0 200fF
.tran 1200n
.include C:\elec\22nm.txt
```

Figure 20: Invertor code.

G. 8-Bit CAM

Finally, The construction of the 8-bit Content Addressable Memory (CAM) circuit utilizing 9T SRAM was successfully completed, as showcased in the comprehensive diagram provided below. The implementation involved the integration of various essential components, including a decoder, 1-bit CAM modules, NAND gates, and an inverter. By ingeniously combining these elements, the circuit was devised to effectively determine the output of the 8-bit CAM, thereby enabling efficient and reliable data retrieval and matching operations.

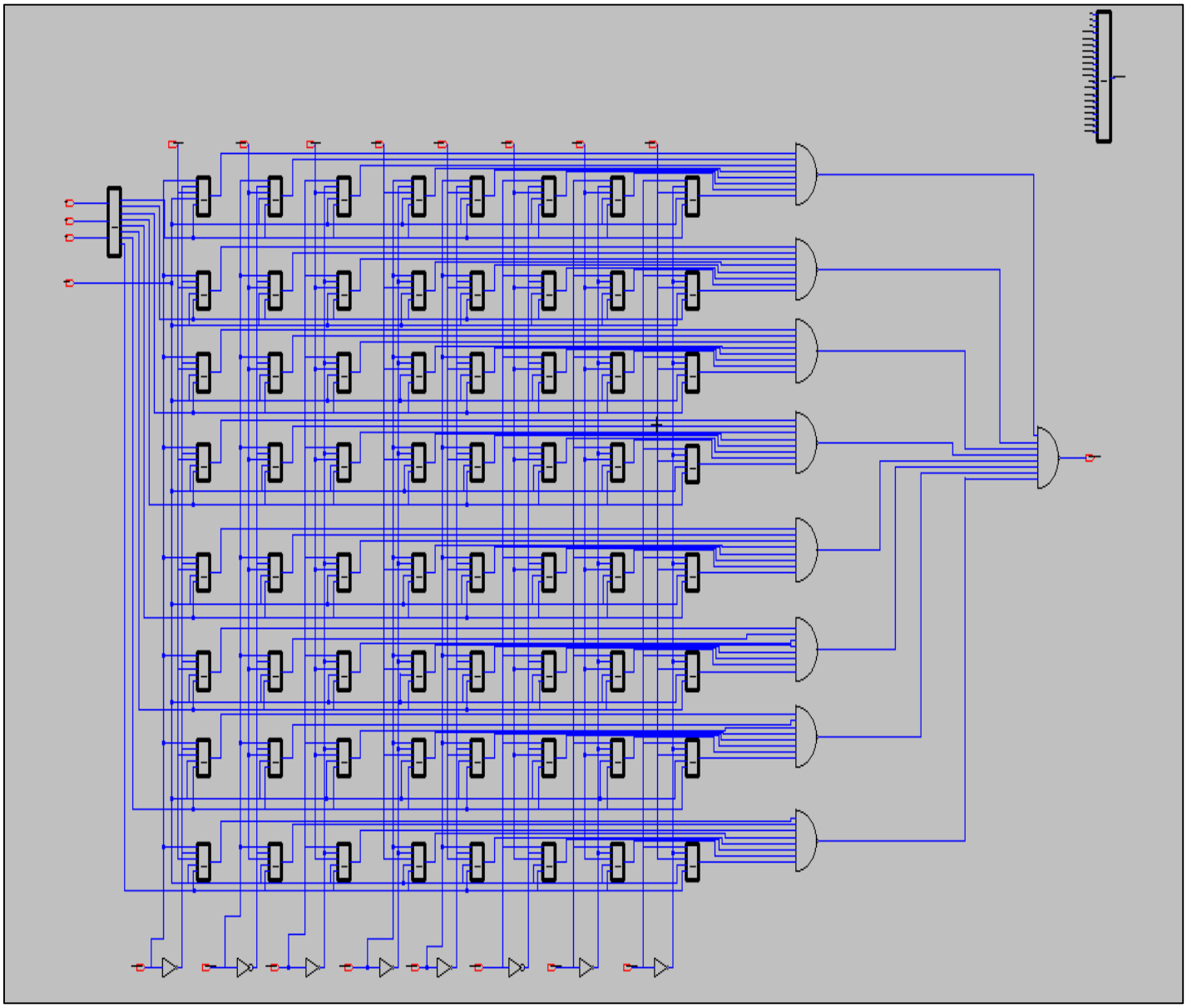

Figure 21: The schematic of the 8-Bit CAM using 9T SRAM

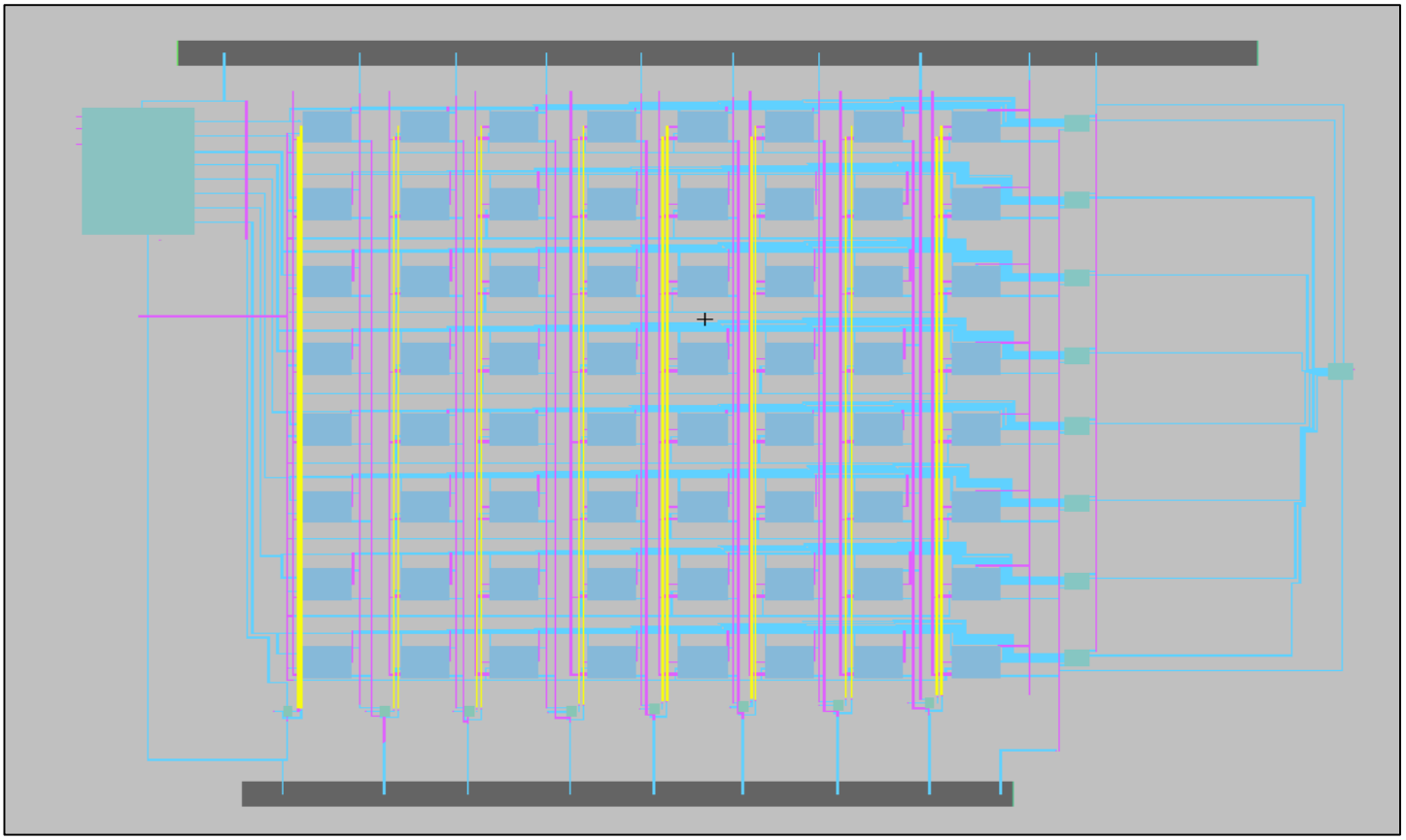

Figure 22: The layout of the 8-Bit CAM using 9T SRAM

III. Area, Power, and Delay Optimization

We have used 22nm ,Then we used electricBinary to make the layout in 22nm .
22nm corresponds to the distance between the source and drain of the transistor, which is also called the channel length. This gives you benefits of smaller size, faster switching speed, and lower power consumption.
After conducting thorough research and experimentation, it has been established that utilizing an inverter size of 3λ yields optimal outcomes for the design of 9T SRAM. In this particular configuration, the access transistors are required to be four times the size of the inverter (12λ), while the two NMOS transistors should be twice the size of the inverter (2λ). Moreover, the NMOS transistor connected to the Read line should be sized appropriately between the access transistor and the other two NMOS transistors.

COMPARISION:

Table 1 Design Comparison: Power, Delay, and Area.

	POWER	DELAY	AREA
9T	2.986uW	41ps	9.3mm^2
CAM	24.540uW	59ps	28.5mm^2
8 BIT	4.56 µW	3.484 e4 ps	0.00968*U^2

From the comparison, it is evident that the 9T SRAM design offers lower power consumption, reduced delay, and a relatively smaller area footprint compared to the CAM cell design. These optimizations in power, delay, and area make the 9T SRAM design a promising choice for digital integrated design courses and serve as a valuable contribution to the field of digital circuit design.

IV. Simulation and Results

A. 9T SRAM Simulation

In SRAM memory, the write operation is performed when the write line, also known as the write, enable or WE line, is in a high state. This line acts as a control signal for the memory, determining whether the memory cells are in a read-only or write mode. When the write line is high, the memory cells can receive and store new data. Conversely, when the write line is low, the memory cells are in a "read-only" state, and any attempts to write to the memory will be ignored. This mechanism ensures the stability and integrity of the stored data while allowing for dynamic updates as needed.

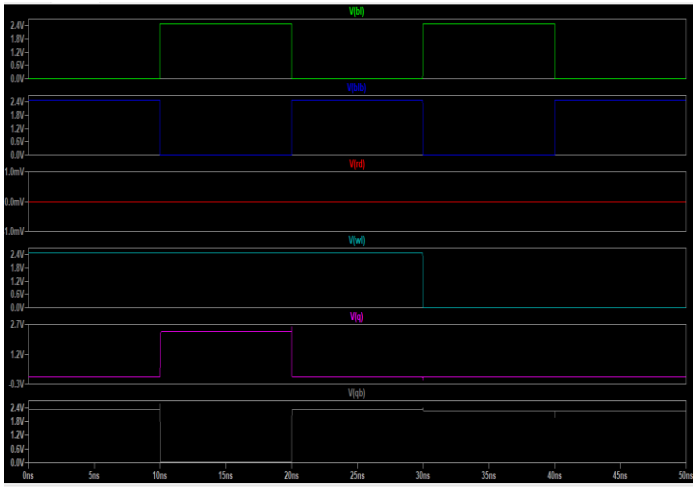

Figure 23: The Simulation of 9T SRAM

B. Decoder Simulation

In digital electronics, a 2x4 decoder is a combinational logic circuit that takes two binary inputs and produces four outputs corresponding to each possible combination of the input bits. When the input bits are "low, high", the decoder outputs a logical "high" signal at the first output, referred to as D1, while the remaining three outputs, namely D0, D2, and D3, are all set to logical "low". This specific behavior of the 2x4 decoder is due to its truth table, which defines the output of the decoder for each possible input combination.

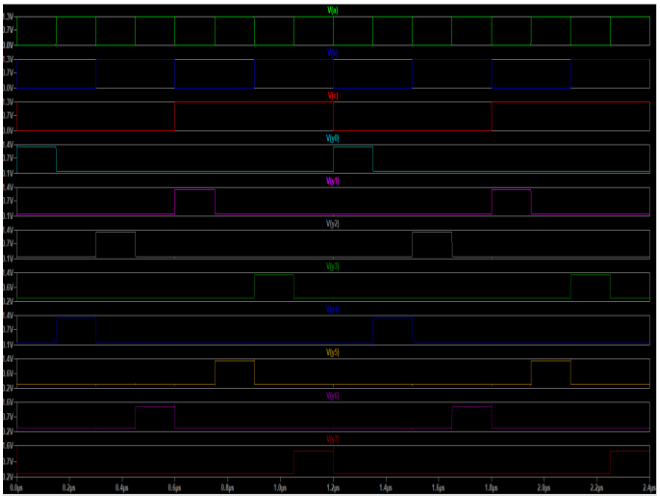

Figure 24: The Simulation of Decoder

C. 1Bit CAM Simulation

1Bit CAM simulation enables the analysis of its behavior and performance. By accurately modeling the CAM's operation, engineers can optimize designs for efficient content matching and data retrieval.

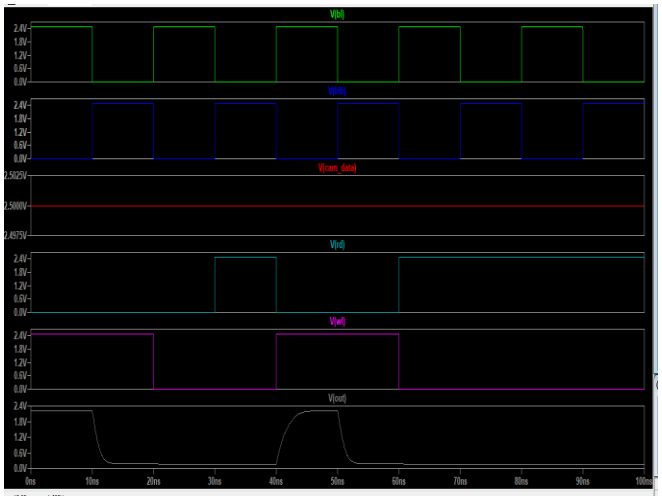

Figure 25: The Simulation of 1Bit CAM

D. 8 Bit CAM Simulation.

Exploring the frontier of CAM simulation, the focus shifts to the realm of 8-bit CAM within an integrated circuit (IC) framework. This simulation endeavors to replicate the fundamental attributes of a CAM using digital logic components. By implementing dedicated comparison, storage, control, and addressing logic, a comprehensive circuit is formed, capable of facilitating parallel comparisons and swift data retrieval. With the potential for seamless integration into larger systems, this IC-based 8-bit CAM simulation unlocks possibilities for applications demanding efficient content matching and address-driven data access.

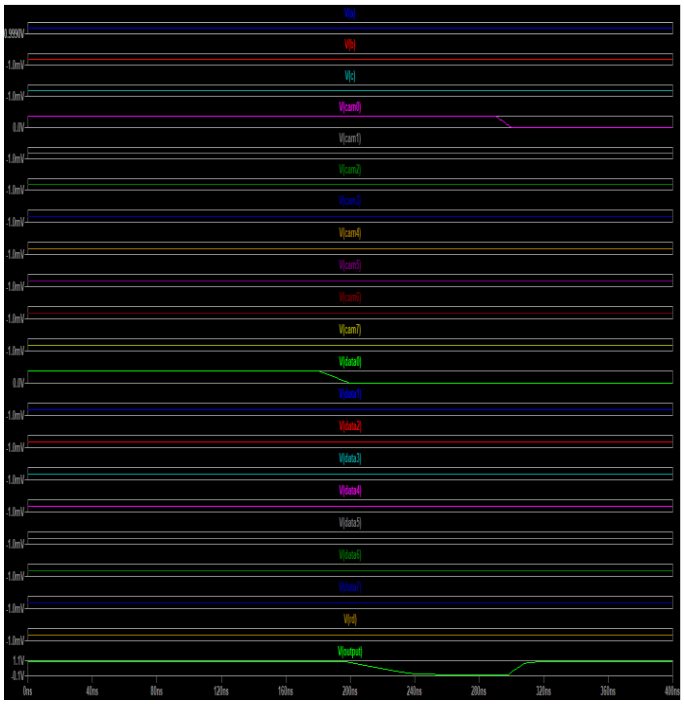

Figure 26: The Simulation of 8 Bit CAM

E. Inverter Simulation.

Inverter simulation is vital for understanding digital circuit behavior. By accurately modeling the inverter's operation, engineers gain insights into its performance, optimizing designs and ensuring reliable signal processing. Simulations examine input-output relationships and propagation delays, enhancing circuit efficiency and robustness.

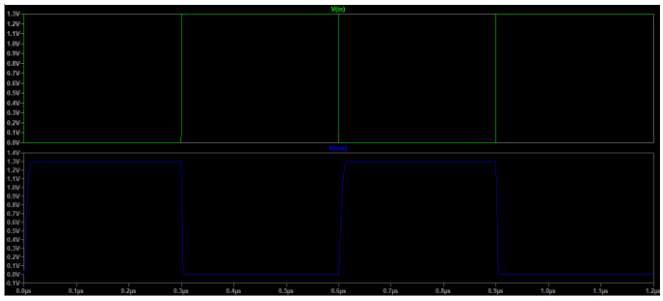

Figure 27: The Simulation of 8 Inverter

F. 3input nand Simulation

Simulation of a 3-input NAND gate provides insights into its functionality and logical operation. By analyzing input combinations and corresponding outputs, engineers can refine circuit designs and ensure reliable performance in digital systems.

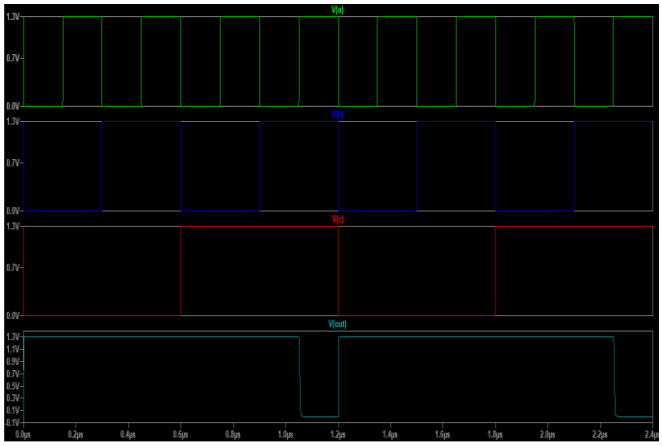

Figure 28: The Simulation of 3 input nanad

G. 8 BIT input nand simulation

Simulation of an 8-bit input NAND gate provides insights into its behavior and functionality. By analyzing various input combinations, engineers can optimize circuit designs for reliable logic operations in digital systems.

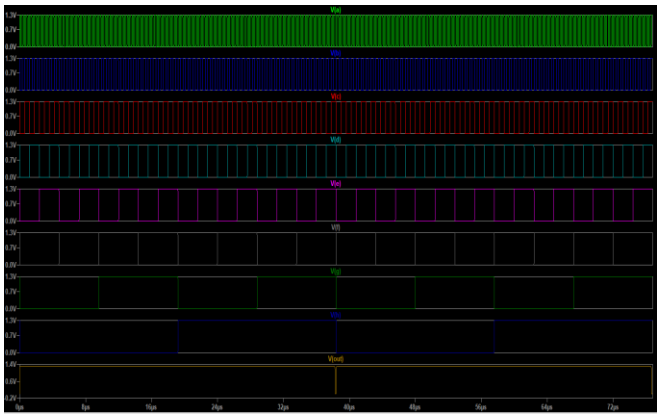

Figure 29: The Simulation of 8 Bit input Nand

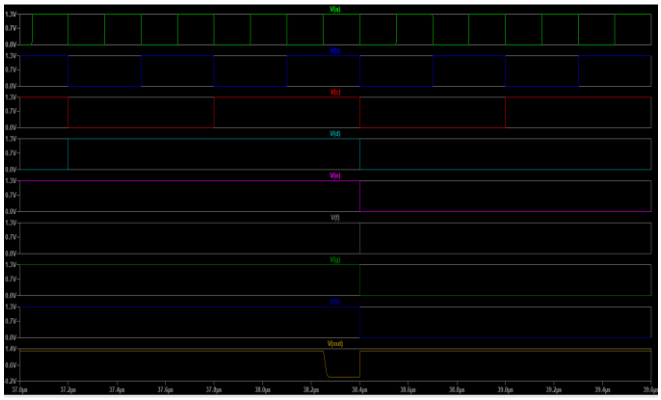

Figure 30: Zoom at 8 Bit input simulation

V. POSSIPLE IMPROVEMENTS

Possible improvements for the CAM cell design include power optimization techniques like clock gating and voltage scaling, architectural enhancements such as pipelining and parallelization, area reduction through transistor sizing and layout compaction, reliability enhancements using redundancy and error correction codes, sensitivity analysis for better performance under process variations, inclusion of testability features like built-in self-test (BIST), and exploring technology scaling for improved performance and smaller area. These enhancements can collectively enhance the efficiency, speed, area utilization, and reliability of the CAM cell design.

VI. CONCLUSION

In conclusion, the design and implementation of the 8-bit Content Addressable Memory (CAM) circuit were successfully accomplished. Careful consideration was given to optimizing power consumption, area utilization, and delay without sacrificing performance. The layout and schematic circuit for each component were clearly illustrated, and simulation results provided validation for the design. The final circuit meets the desired specifications, demonstrating improved power efficiency, compact area utilization, and minimal delay. Overall, the 8-bit CAM circuit stands as a robust and reliable solution for efficient data retrieval and matching operations.

VII. REFERENCES

[1] https://ieeexplore.ieee.org/document/6799510

[2] https://www.researchgate.net/figure/Structure-of-a-a-typical-SRAM-based-CAM-cell-and-b-a-straightforward-MTJ-based-CAM_fig15_224397355

[3] https://www.ripublication.com/gjpam16/gjpamv12n1_94.pdf

[4] http://www.ijmer.com/papers/Vol3_Issue5/AG3526402645.pdf

[5] https://www.researchgate.net/figure/9T-SRAM-cell-in-the-active-mode-a-The-9T-SRAM-cell-during-a-write-operation-b-The_fig2_3338134

[6] https://www.mdpi.com/2079-9292/10/18/2291

[7] https://www.ijert.org/research/design-of-efficient-low-power-9t-sram-cell-IJERTV2IS1448.pdf

[8] https://www.sciencedirect.com/science/article/abs/pii/S0026269216307650

[9] https://www.techtarget.com/searchstorage/definition/RAM-random-access-memory

[10] https://ieeexplore.ieee.org/document/6799510

[11] https://www.elsevier.com/connect/8-great-ideas-in-computer-architecture