Chat-GPT Aware Full-Custom Design of an 8-bit Content-Addressable Memory (CAM) Using a 9T SRAM Cell

Nafe Abubaker[1], *Abdullah Naser*[2], *Awad Sholi* [3], *Hatem Hussain*[4]

ENCS3330 - Integrated Circuits

1200047@*student.birzeit.edu*[1], 1202063@*student.birzeit.edu*[2], 1202080@*student.birzeit.edu*[3],1200894@student.birzeit.edu[4].

Abstract - **The primary objective of this project is to develop an 8-bit Content Addressable Memory (CAM) using the Electric Generic 22nm CMOS Library, with a focus on a 9T SRAM architecture. Specifically tailored for the 14nm process technology, the design emphasizes reducing the footprint of the CAM cell while optimizing power efficiency, especially during write operations. The overarching design strategy aims to strike a balance between minimal power consumption and low latency to maximize CAM performance. Key aspects of the project include the utilization of 9T SRAM, CAM cell design, addressing low power considerations, incorporation of XOR function, and the implementation of efficient read and write operations.**

I. INTRODUCTION

Random Access Memory (RAM) is a vital component within a computer's primary memory, facilitating direct access to the Central Processing Unit (CPU). Its primary function is to enable read and write operations on specific memory locations. RAM plays a crucial role in enhancing the speed of information processing by serving as a temporary storage area for actively used applications. It's important to note that RAM is a type of volatile memory, meaning that the stored data is lost when the power is turned off. Retrieving data from RAM involves specifying a memory address, and multiple cycles are required to access the relevant content word and retrieve the requested data.

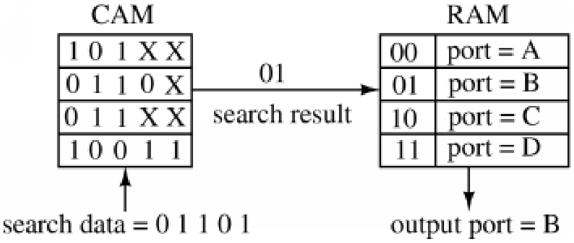


Fig. 1: Schematic illustration of CAM vs RAM [7].

On the flip side, when dealing with read and write operations targeted at specific addresses, Content Addressable Memory (CAM), also known as Associative Memory, bears a resemblance to Random Access Memory (RAM). However, it's worth noting that CAM incorporates additional searching features compared to RAM.

The process involves entering the data word, and then the remaining memory is parallelly compared by iteratively searching for matches, resulting in a list of locations where the data word is stored [2]. During this procedure, a matching line is activated when the incoming data matches the stored data. Subsequently, an encoder generates an encoded representation of the match location, typically expressed by log2 w bits. Despite the faster search speeds offered by CAM compared to RAM, its implementation requires additional circuitry, leading to increased power consumption and occupation of silicon space.

Schematics designed for SRAM can be effectively employed for implementing CAM, as both SRAM and CAM serve comparable functions. Various SRAM configurations, such as 6T, 8T, 9T, and 10T, can be chosen based on project specifications and specific requirements. In this project, the primary emphasis will be on designing and executing an 8-bit CAM using a 9T SRAM architecture. The 9T SRAM architecture is chosen for its advantages in terms of space efficiency, stability, and power usage. The schematic representation of the 9T SRAM will serve as the foundational framework for the CAM implementation, providing the necessary structure for subsequent phases of development and optimization.

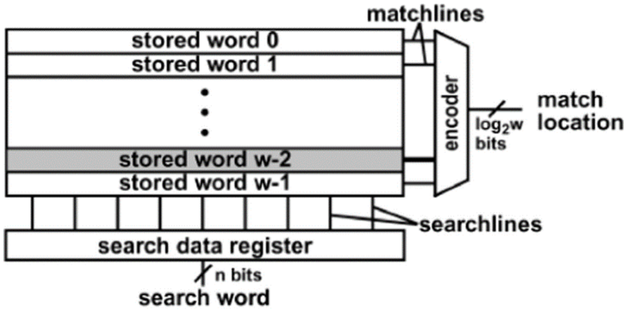


Fig. 2: illustration of a content-addressable memory [8].

II. DESIGN AND IMPLEMENTATION

In the implementation of this project's design, we utilized a 22nm CMOS library to ensure accurate timing and minimize signal skew. Our primary focus was on achieving balanced rise and fall periods for both PMOS and NMOS devices.

This required careful consideration of factors such as the output resistance and inherent delay of each transistor. Through extensive simulations and testing, we determined that the optimal approach for achieving consistent rise and fall timings within the 22nm process involved maintaining a ratio of approximately 2:1 between PMOS and NMOS devices.

As part of the design process, we introduced a 9T SRAM cell. Known for its simplicity, reliability, and fast operation, this type of memory cell is commonly utilized in digital systems. The incorporation of the 9T SRAM cell aimed to enhance the overall read perfomance of the CAM design. Figure 1 provides a schematic representation of the structural layout and connections of the 9T SRAM.

By combining these design considerations and leveraging the advantages of the 9T SRAM cell, our objective was to optimize the performance and functionality of the CAM design while adhering to the constraints imposed by the 14nm CMOS manufacturing technology. Therefore, the components are as follows:

- ***9T SRAM***

The provided image illustrates the 9T SRAM circuit, featuring four input lines: the word line (WL), the read line (Rd), and two additional bit lines (BL and BLB). This configuration enables the circuit to receive input signals and process them appropriately. Additionally, the 9T SRAM circuit incorporates two output lines, namely QB and Q, facilitating the retrieval and transmission of processed data within the circuit [1].

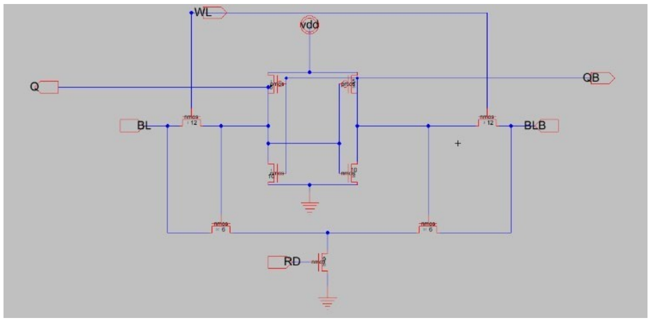


Fig. 3: The schematic of 9T SRAM.

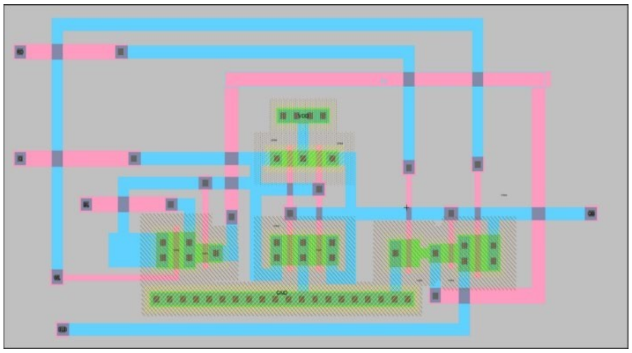


Fig. 4: The layout of 9T SRAM

- ***1-Bit CAM***

The 1-bit Content-Addressable Memory (CAM) circuit was subsequently built by integrating additional inverters, pass gates, and SRAM components for comparison, culminating in the generation of a match signal [9]. Figure 4 provides the schematic diagram of the 1-bit CAM circuit.

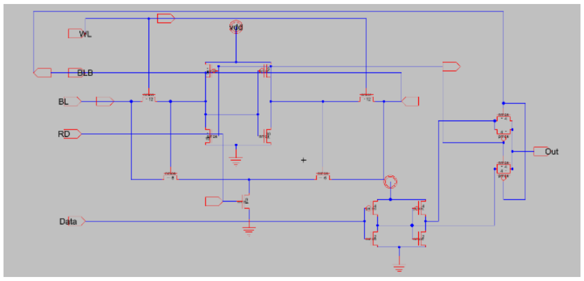


Fig. 5: The schematic of 1-Bit CAM circuit

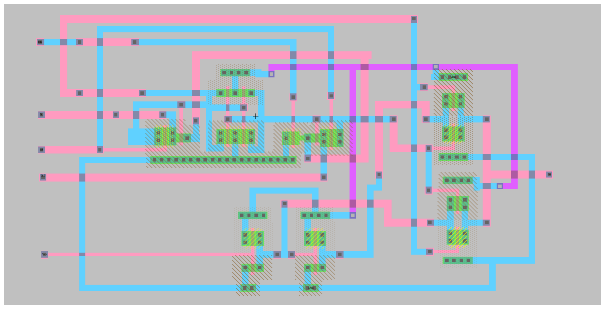


Fig. 6: The layout of the 1-bit CAM

- ***3-input NOR***

The project heavily utilized the concept of the three-input NOR gate. This logic gate accepts three input signals and executes logical NOR operations. Its output signal is contingent on the input combination, producing a low output only when all three input signals are high, and a high output for any other input combination [10].

- ***The 3x8 decoder***

In the quest for the suitable 1-bit CAM cell, the 3x8 decoder within the 8-bit CAM implementation selects one of eight options. By decoding the input address, it triggers the activation of the output line corresponding to the specified cell. The configuration of the decoder is evident in the CAM diagram through its design and layout. The inclusion of the 3x8 decoder enables the CAM to efficiently retrieve the required data from the 8-bit CAM array [3].

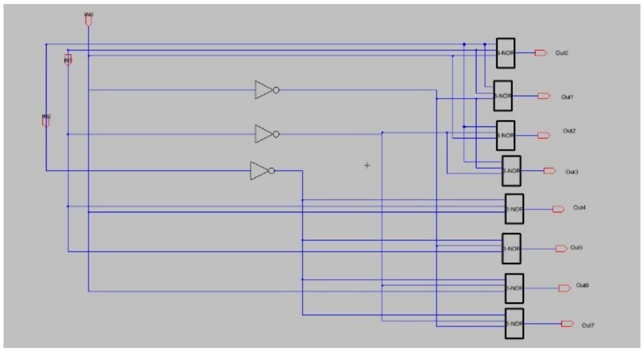


Fig. 7: The schematic of the 3x8 decoder

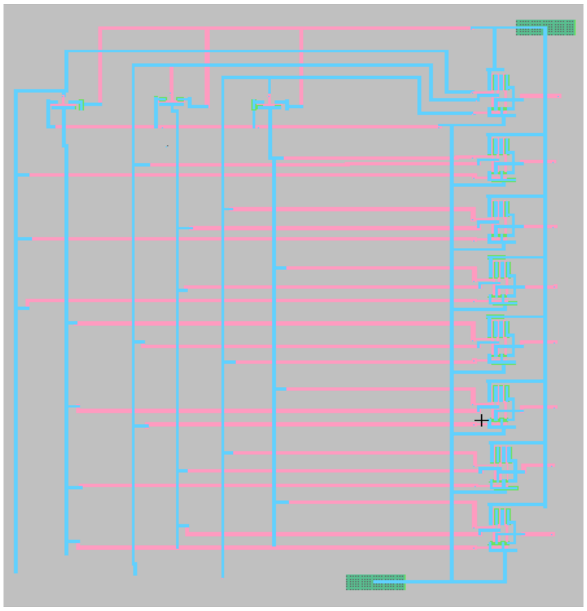


Fig. 8: The layout of the 3x8 decoder

- ***The 8-input NAND***

The implementation of the 8-bit CAM will involve the utilization of an 8-input NAND gate when interconnecting the 8 blocks of the 1-bit CAM to form the ultimate output. The inclusion of the 8-input NAND gate allows the CAM system to effectively merge the outputs from the individual 1-bit CAM blocks, resulting in the generation of the final output [6].

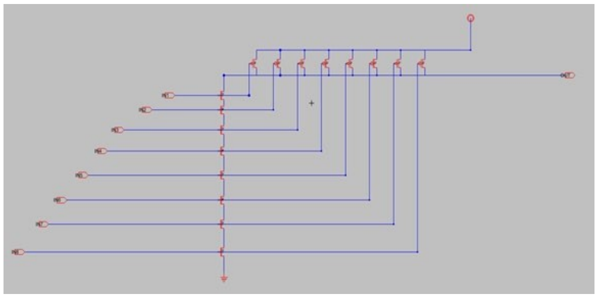


Fig. 9: The schematic of 8-input NAND

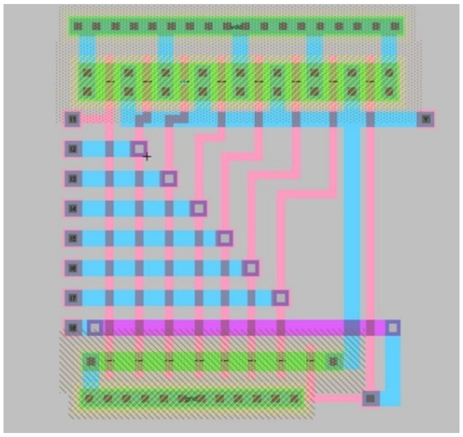


Fig. 10: The layout of the 8-input NAND

- ***Invertor***

The inverter circuit serves as the last essential component for the 8-bit CAM circuit. The following illustrations depict schematic and layout representations of the inverter component.

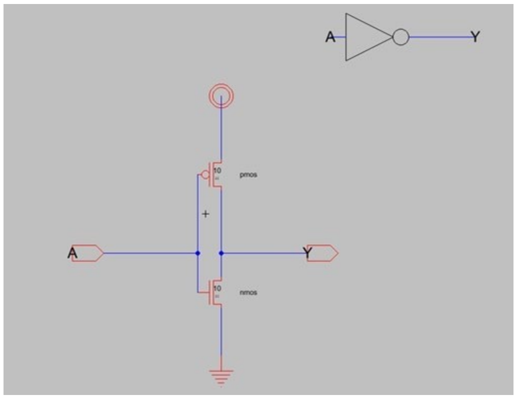


Fig. 11: The schematic of Inverter

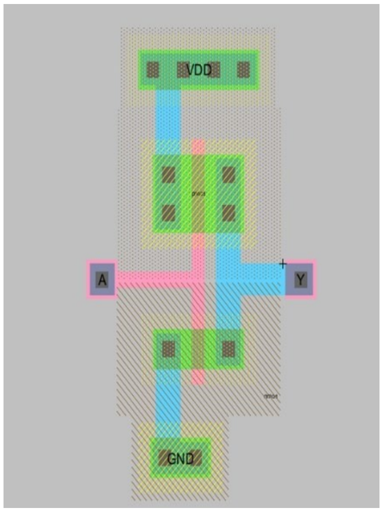


Fig. 12: The layout of Inverter

- ***8-Bit CAM***

The 8-Bit CAM circuit, utilizing 9T SRAM, has been successfully implemented as depicted in the accompanying figure. The design and implementation of the 8-bit CAM's output involved the incorporation of a decoder, 1-bit CAM, NAND gates, and an inverter.

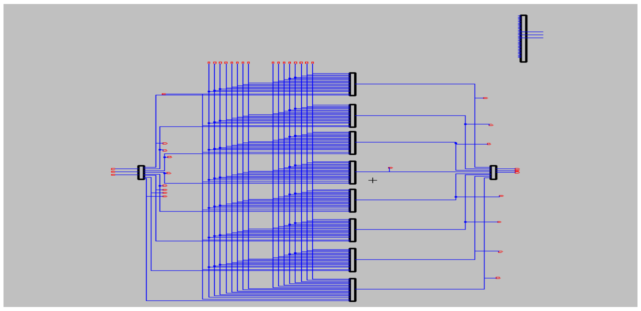


Fig. 13: The schematic of the 8-Bit CAM using 9T SRAM.

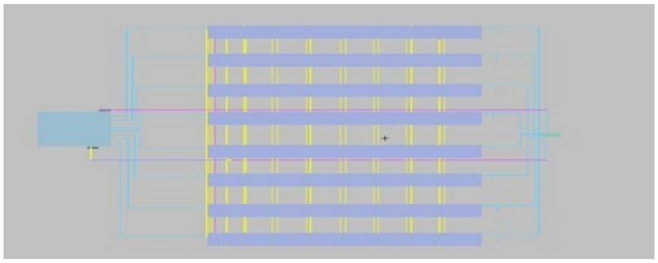


Fig. 14: The layout of the 8-Bit CAM using 9T SRAM

III. RESULTS

The Write Line (WL line) governs write operations in SRAM. When the Write Line is set to a high state, indicating that the memory cells are write-enabled, new data can be received and stored, facilitating data updates. Conversely, when the Write Line is low, the memory cells function in read-only mode, preventing any modifications to the stored information. This approach ensures data consistency and integrity by safeguarding against unintended alterations. The regulation of the Write Line in SRAM establishes a selective and controlled approach to data management and updating, thereby ensuring the reliability of stored information.

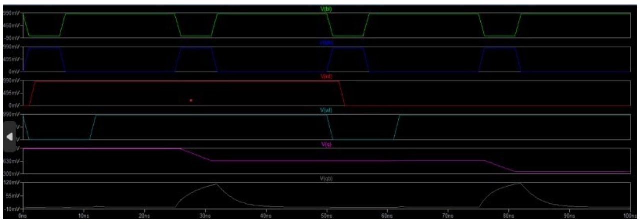


Fig. 15: The simulation of 9T SRAM

The simulation of the 1-bit CAM yielded positive results, indicating a successful match between the input data and the stored data. The obtained outcomes affirm the effective operation of the CAM, as the expected result aligns with the desired output. This outcome substantiates the reliability and efficiency of the CAM in retrieving and identifying stored data, underscoring its potential for practical applications.

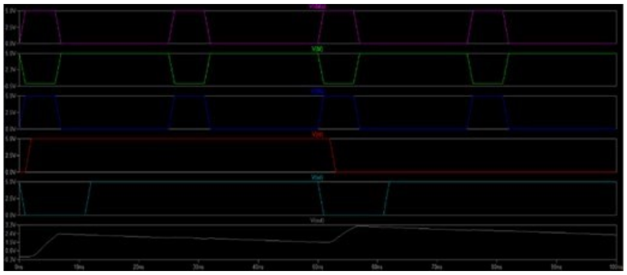


Fig. 16: The simulation of 1bit CAM

During the three-input NOR simulation, the gate's output demonstrated a true state when all three input signals were false, and it showed a false state when at least one of the inputs was true. This observation aligns with the truth table, showcasing its ability to perform logical negation on multiple input signals. The successful simulation underscores its versatility in various logic circuit designs and highlights its reliability in logical operations. As we know, the decoder's output must have just one active in each scenario, as shown in the picture. So, the findings are correct.

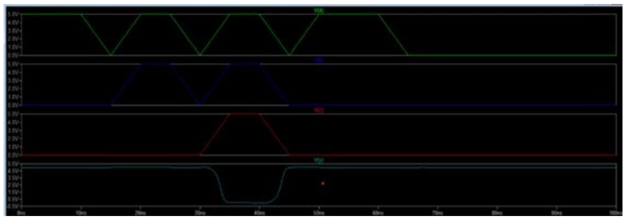


Fig. 17: The simulation of 3-input NOR

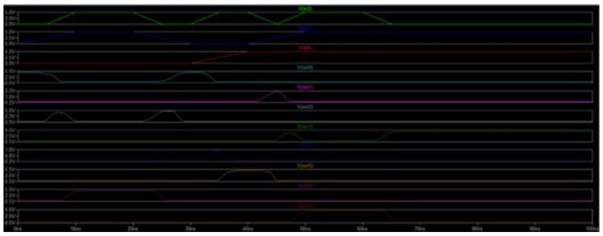


Fig. 18: The simulation of 3x8 decoder

The inverter circuit's simulation with a 22nm technology node demonstrated the predicted behavior and functionality. The inverter's output was found to be the logical counterpart of the input signal, in accordance with logic inversion rules.

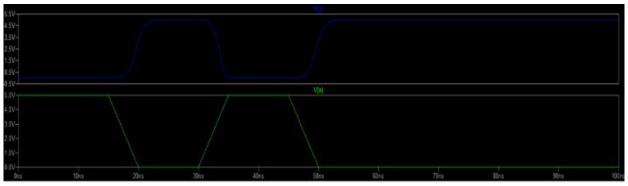


Fig. 19: The simulation of Invertor (22nms)

When all eight input signals were false (logic 0), the NAND gate's output was true (logic 1), resulting in a voltage level of 2 volts. In all other input combinations, the result was false (logic 0). This result is consistent with the truth table, demonstrating its capacity to conduct logical conjunctions and generate the logical negative of the combined inputs.

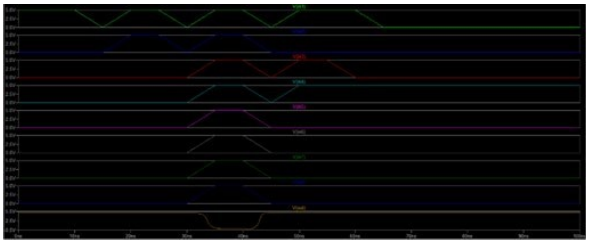


Fig. 20: The simulation of 8-input NAND

A comprehensive computational study is employed to elucidate the operation and performance of the 8-Bit Content-Addressable Memory (CAM) circuit implemented using a 9T static random-access memory (SRAM). The primary objective of this simulation is to validate the functionality and performance of the 8-bit CAM architecture, specifically in terms of how rapidly and accurately it can retrieve data when provided with a search key.

In comparison to traditional 6T SRAM cells, the 9T SRAM implementation in the simulation presents several advantages, including lower leakage power and enhanced stability. Through the simulation process, designers gain a deeper understanding of the performance characteristics of the 8-bit CAM. This knowledge allows them to fine-tune its architecture and configuration to suit the requirements of specific applications.

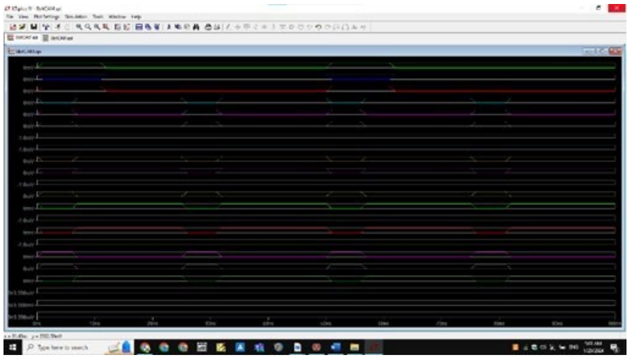


Fig. 21: The simulation of the 8-Bit CAM using 9T SRAM.

IV. OPTIMIZATION OF AREA, POWER, AND DELAY

On the other hand, we have made the circuit more economical and efficient without sacrificing its performance or output quality in terms of space or cost optimization. For example, let us look at the 8-Input NAND gate depicted in the picture. Seven 2-NAND gates were needed at first, for a total of 28 gates. To save on the number of gates required without sacrificing the circuit's performance or usefulness, we have made several adjustments.

Therefore, we created the 8-NAND gate employing 8 parallel PMOS and 8 series NMOS transistors, totaling 16 transistors, in order to decrease both areas and costs.

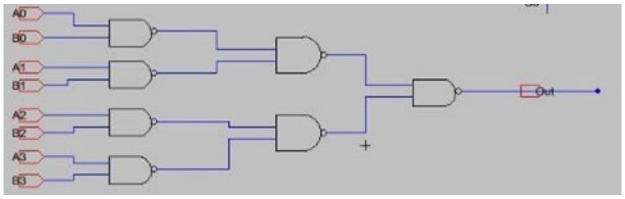


Fig. 22: 8-NAND gate

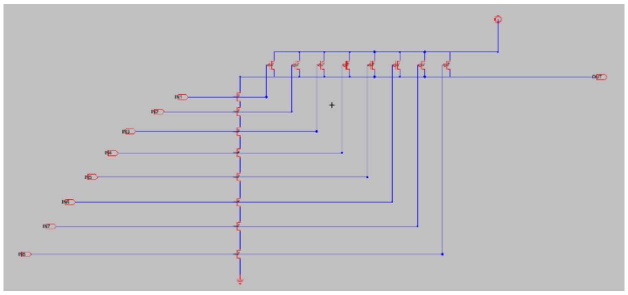


Fig. 23: 8-NAND gate using PMOS And NMOS

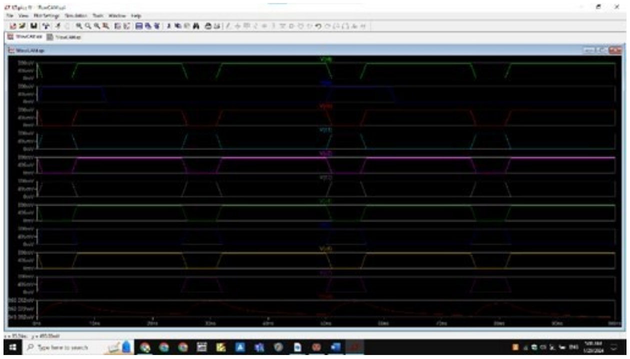


Fig. 24: One Bit CAM simulation

	Delay	Power
9t-SRAM	25.62u	5ps
1bit-CAM	21ps	43.789µW
8bit-CAM	38ps	54.168µW
Transistor Weight 10	0.052ns	1.26µW
Transistor Weight 2	0.521	0.13µW
Transistor Weight 12	0.27ns	1.22µW

TABLE 1: Output of changing the size of the CMOS.

VI.CONCLUSION

In conclusion, this paper has offered a thorough examination and detailed exposition of the design and implementation stages of the 8-Bit Content-Addressable Memory (CAM) circuit. Each component utilized in the final circuit was meticulously presented, providing both schematic circuit representations and layout details. The pursuit of minimizing power consumption, optimizing area utilization, and reducing latency in the 8-Bit CAM involved strategic approaches, ensuring that these enhancements did not compromise the performance or search capabilities of the Content Address Memory (CAM). Notably, the simulation results for various circuits were rigorously collected, meticulously examined, and effectively communicated. This comprehensive study not only enriches our understanding of the 8-Bit CAM architecture but also lays the groundwork for further advancements in optimizing its functionality for diverse applications.

REFERENCES

[1] Design of High Speed 9T SRAM Cell at 16 nm Tech- nology with Simultaneous Read-Write Feature — https. https://ieeexplore.ieee.org/document/9929643.

[2] Difference between Random Access Memory (RAM) and Content Addressable Memory (CAM) - GeeksforGeeks — geeksforgeeks.org. https://www.geeksforgeeks.org/difference-between-random-access- memory-ram-and-content-addressable-memory-cam.

[3] Logic diagram of a 3-to-8-line decoder using NOR, NOT gates, enable input — Intro. to Logic Design — youtube.com. https://www.youtube.com/watch?v=Nz3ya iDiE.

[4] NOR Gate in Digital Electronics - Javatpoint — javatpoint.com. https://www.javatpoint.com/nor-gate-in-digital-electronics.

[5] FAHAD, E. Decoder, 3 to 8 Decoder Block Diagram, Truth Table, and Logic Diagram — electroniclinic.com. https://www.electroniclinic.com/decoder-3-to-8-decoder-block-diagram- truth-table-and-logic-diagram.

[6] KUMAR, M., HUSSAIN, M., AND PAUL, S. Performance of a two input nand gate using subthreshold leakage control techniques.

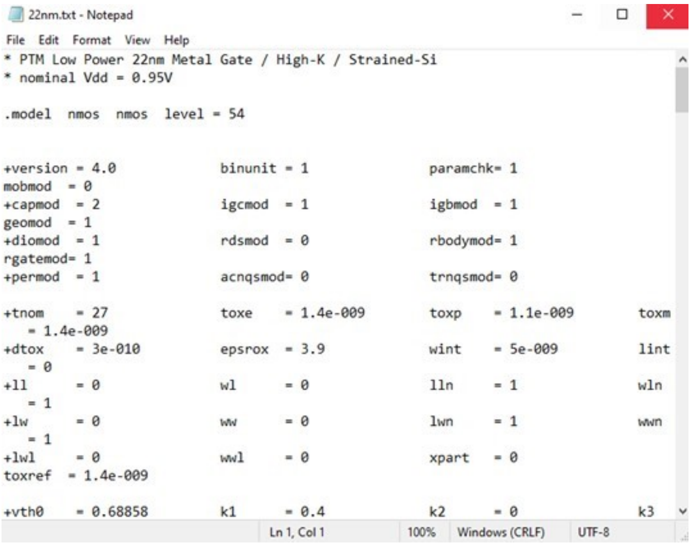
[7] MUJAHID, O., ULLAH, D. Z., HAFEEZ, A., AND FOUZDER, T. Design Exploration of LH-CAM with Updating Mechanism. 05 2020, pp. 477– 486.

[8] PAGIAMTZIS, K., AND SHEIKHOLESLAMI, A. Content-addressable memory (cam) circuits and architectures: A tutorial and survey. IEEE journal of solid-state circuits 41, 3 (2006), 712–727.

[9] S.A, S., SWEDHA, A., AND NAVEEN, D. Survey of content addressable memory. 1516.

[10] TAN, M., LENTARIS, G., AND AMARATUNGA, G. Device and circuit- level performance of carbon nanotube field-effect transistor with bench- marking against a nano-mosfet. Nanoscale research letters 7 (08 2012), 467.

APPENDIX – 22nm file



```
* PTM Low Power 22nm Metal Gate / High-K / Strained-Si
* nominal Vdd = 0.95V

.model  nmos  nmos  level = 54


+version = 4.0           binunit = 1           paramchk= 1
mobmod   = 0
+capmod   = 2            igcmod   = 1          igbmod   = 1
geomod   = 1
+diomod   = 1            rdsmod   = 0          rbodymod= 1
rgatemod= 1
+permod   = 1            acnqsmod= 0           trnqsmod= 0

+tnom     = 27           toxe     = 1.4e-009   toxp     = 1.1e-009   toxm
   = 1.4e-009
+dtox     = 3e-010       epsrox   = 3.9        wint     = 5e-009     lint
   = 0
+ll       = 0            wl       = 0          lln      = 1          wln
   = 1
+lw       = 0            ww       = 0          lwn      = 1          wwn
   = 1
+lwl      = 0            wwl      = 0          xpart    = 0
toxref   = 1.4e-009

+vth0     = 0.68858      k1       = 0.4        k2       = 0          k3
```