# Full-Custom Design 8-bit CAM using 9T SRAM

Oday Ziq

***Abstract***—Content Addressable Memories (CAMs), unlike Random Access Memories (RAMs), are Memory units that perform content matching instead of address matching, in CAMs, the input is the data to be searched for, and the output is the address of the given data, given it exists. This paper's main focus is designing and implementing of a cutting-edge Content Addressable Memory (CAM) cell using a 9T Static Random Access Memory (SRAM) configuration. CAM cells are essential components in high-performance computing systems, enabling fast data retrieval and matching operations. The utilization of the 9T SRAM configuration offers advantages in terms of area efficiency, power consumption, and performance trade-offs. We use 22nm process for CMOS in LTSpice tool. Index Terms—CAM, SRAM, utilization, power consumption.

## I. INTRODUCTION

### *A. SRAM*

Static Random Access Memory (SRAM) is a type of semiconductor memory that is widely used in digital integrated circuits for storing and retrieving data. SRAM provides fast access times and retains data as long as power is supplied to the circuit (volatile memory type). There are many designs implementations for an SRAM, including 6T, 7T, 8T 9T, which indicates the number of transistors used to design the SRAM Cell.

### *B. 9T SRAM*

A 9T SRAM consists of Nine Transistors, including 2 inverters that play the Keeper's role in the Memory Unit, the other 5 Transistors are used to control other operations on the Memory unit such as Write Read. A 9T SRAM has four input ports, WL, RL, BL, and BLB and has two output ports Q QB. SRAM uses two inverters to store data (Keeper), hence the two output ports Q QB. When writing on the SRAM Data, the WL signal works as an enable to the writing operation, by having a high logic (One), the WL enables 2 NMOS.

Transistors to transfer Data from BL, BLB to Q, QB respectively, the values on BL (Bit line) and BLB (Bit Line Bar) should be the opposite of each other, this is done to allow the Keeper to change the values easily.

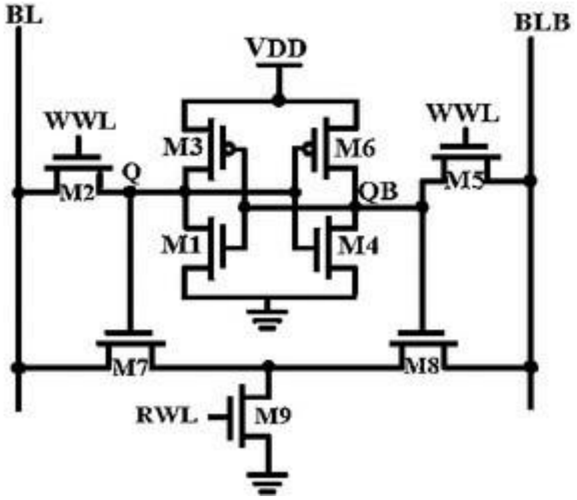


*Figure 1:  9T SRAM*

### *C. One Bit CAM*

Content-Addressable Memory (CAM) is a specialized type of memory that allows for efficient searching and retrieval of data based on its content, rather than relying on specific memory addresses. It offers a unique associative memory function, enabling rapid data matching and pattern recognition operations. At its core, a CAM cell consists of two main components: a data storage array and a comparison circuit. The data storage array stores the content or patterns to be searched, while the comparison circuit compares the input data with the stored content and retrieves the corresponding memory location or addresses if the data matched.

### *D. Larger CAM Cells*

A CAM Cell of Width W and Length L can be built using W*L Single Bit CAM Cells, the final Cell consists of L Rows where each row consists of W Single Bit CAMs.

Since each row Consists of W CAM Cells, there are W Match outputs for each bit in the row, in order to check that the Entire Row matches the Entered Data, we must AND all the match bits & the result will be representative of the Matching between the stored Data & input CAM data.

What Makes CAMs so useful is that all comparisons can be done at the same time, i.e., in parallel. Meaning we can compare an input data to all addresses in the memory in one Cycle or Moment. It can be represented in the figure below.

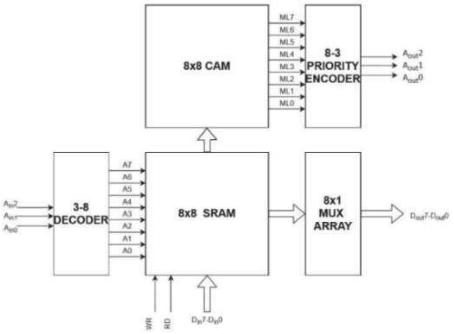


Figure 2: Larger CAM cells

## II. DESIGN IMPLEMENTATION

### A. 9T SRAM

The Schematic Design for 9T SRAM is as shown:

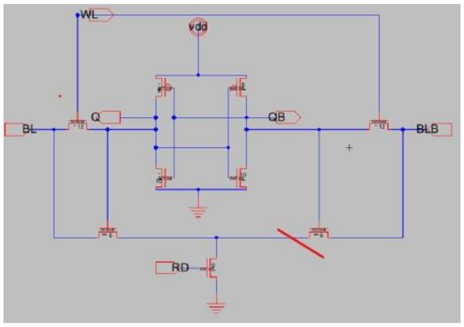


Figure 3:9T SRAM schematic

The Layout for the shown design is as follows:

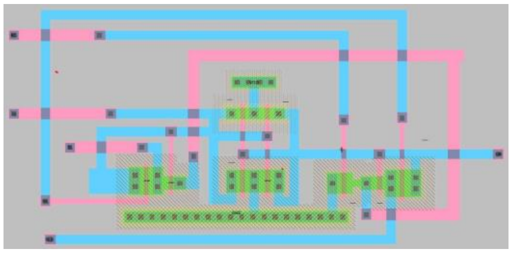


Figure 4: 9T SRAM layout

### B. One bit CAM

As we know, a one bit CAM uses an SRAM alongside a comparison circuit to determine where a match has occurred or not, the schematic for one but CAM is shown below.

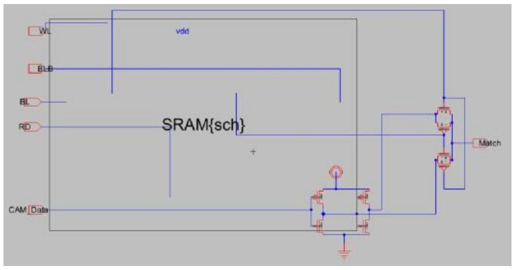


Figure 5: One bit CAM schematic

The Layout for the shown design is as follows:

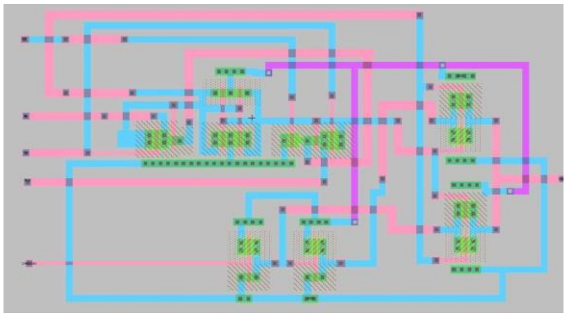


Figure 6: One bit CAM layout

### C. NAND

An 8-bit NAND Is used to create an AND gate to determine where all bits of a given row matched the input CAM data or not.

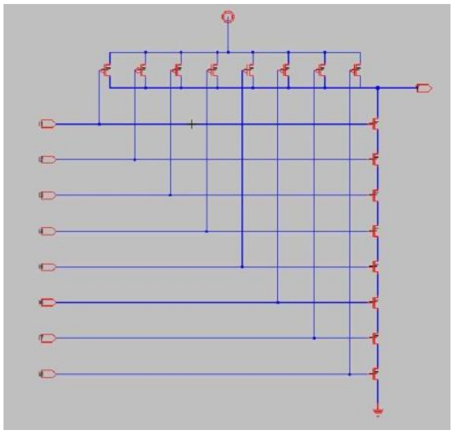


Figure 7: NAND schematic

The Layout for the shown design is as follows:

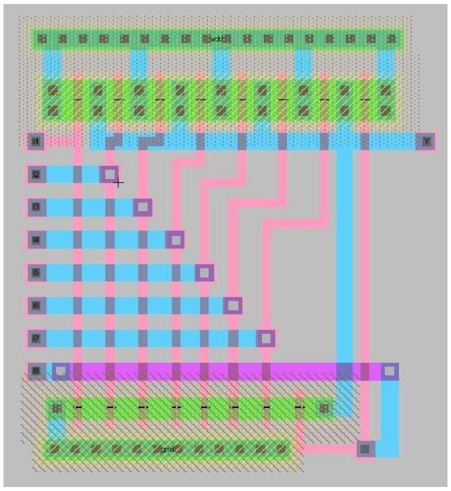


Figure 8: NAND layout

### D. NOR

A 3-bit NOR is used in the Decoder Design. The schematics for a 3 bit NOR is as shown below.

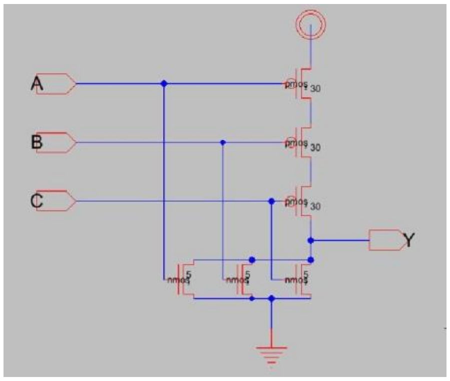


*Figure 9: NOR schematic*

The Layout for the shown design is as follows:

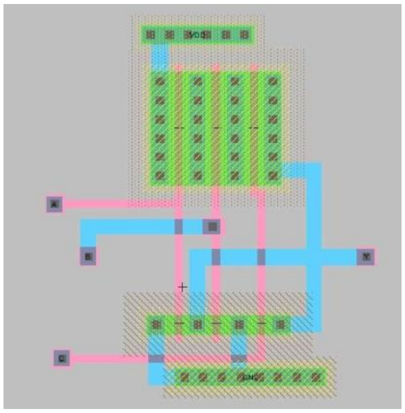


*Figure 10: NOR layout*

### E. 3x8 decoder

A CAM, offers the functionality of storing the data on the SRAMS, but we’re usually dealing with a group of bits as a single number which leads us to splitting larger memory units in rows where each row represents a number or a group of bits, an important functionality in such memory units is the ability to decide on which row we plan on writing data, in order to determine which Row should write the given input data, we need to implement a 3x8 Decoder where the outputs would work as enable lines for the CAM Row Cells.

A 3x8 Decoder can be built using 3 Inverters & 8 NOR Gates.

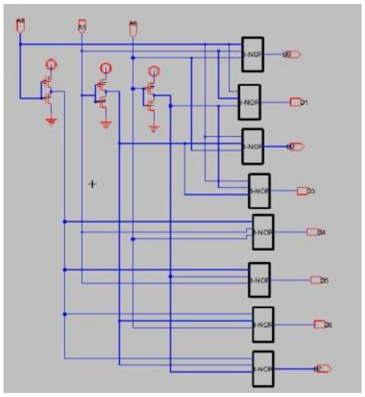


*Figure 11: 3x8 decoder schematic*

The Layout for the shown design is as follows:

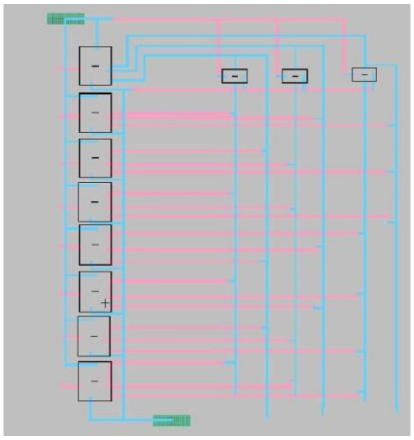


*Figure 12: 3x8 decoder layout*

### F. 8x3 encoder

After the matching process is done, we need to retrieve an address of the matched data, this can be done using an encoder. The schematics for an encoder consist of 3 OR Gates which are built using NOR Gates + Inverter for each OR. The schematic is shown in the figure below.

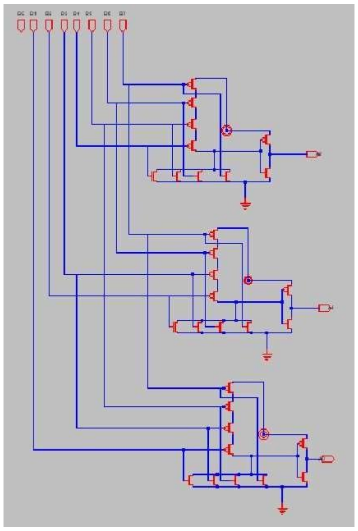


*Figure 13: 8x3 encoder schematic*

The Layout for the shown design is as follows:

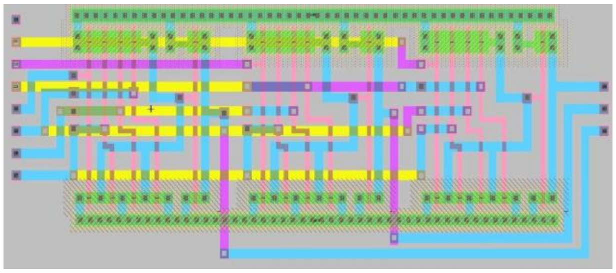


*Figure 14: 8x3 encoder layout*

### G. *One Row CAM*

A singular Row CAM consists of 8 1-bit CAM Cells, where each cell has an input data to provide the data being read a CAM data to apply comparisons, and an enable write signal (WL), the outputs of all CAM CELLS (Match) should be ANDed (NAND THEN NOT) to determine whether the entire CAM Byte matches the stored Data or not.

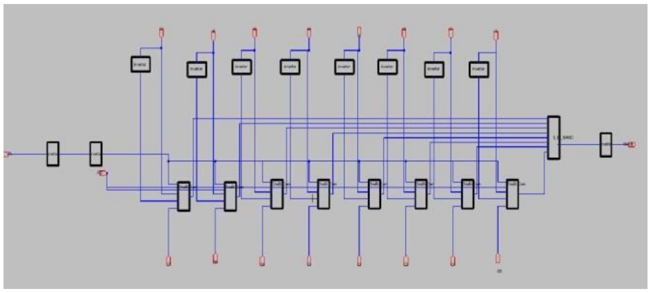


*Figure 15: One Row Cam schematic*

The Layout for the shown design is as follows

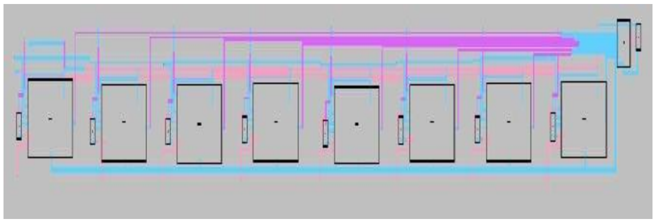


*Figure 16: One Row layout*

### H. *CAM Matrix*

The Final CAM Matrix Design consists of 8 CAM ROWS where data cam input lines are connected to all of them, a Decoder to determine where to write the data an encoder to determine the address that a (match) occurred at.

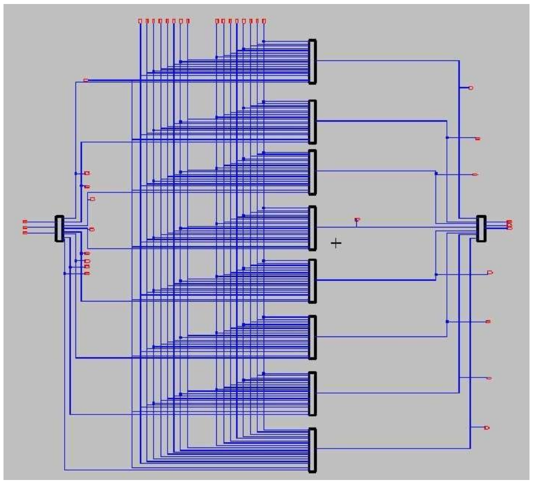


*Figure 17: CAM Matrix schematic*

The Layout for the shown design is as follows

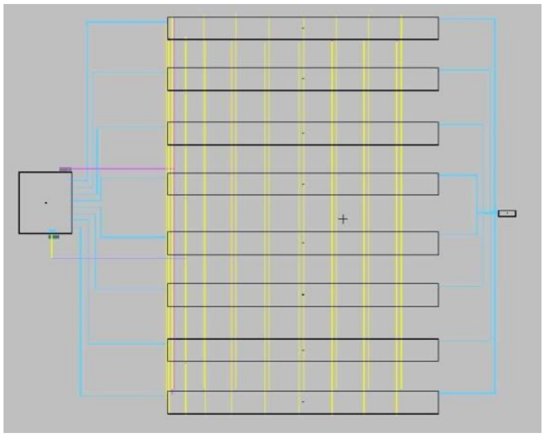


*Figure 18: CAM Matrix layout*

## IV. SIMULATION RESULT

### A. *9T SRAM*

The way SRAM works is by Storing the value & its' complement, Q & QB in the Keeper, we can write on the SRAM by providing the desired data on the Bit-Line (BL) & its' complement on the Bit-Line-Bar (BLB), while enable the write operation via (WL = HIGH) , which can be seen in the figure below as we keep the SRAM in Write Mode(WL High) & try writing High & Low on the SRAM respectively.

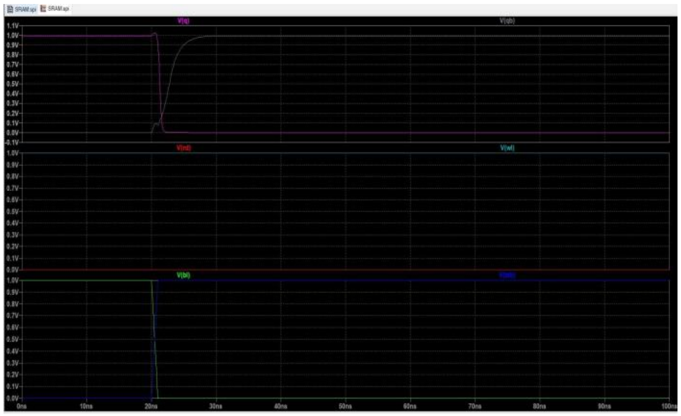


*Figure 19: output of 9T SRAM*

### B. *One bit CAM*

A One Bit Cam consists of an SRAM to store a value (Q & QB) & a comparison circuit do determine whether an input (CAM Data) is equivalent to the Stored Value or not, and output a high voltage if both Values & Equivalent and a low voltage otherwise.

We can test this by setting the CAM_Data to High & writing on the SRAM, we can see in the figure 20 that the (Match) output is high when the stored value is High & 0 when it's Low.

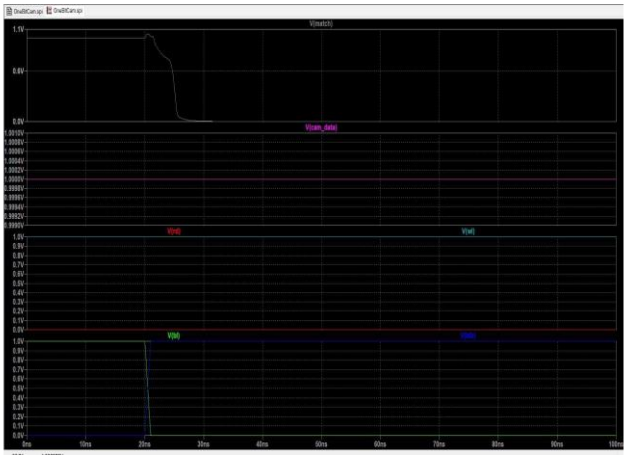


Figure 20: output of One bit CAM

*C. 8-bit NAND*

Shown in the figure below is a simulation for a 8-bit NAND; Logic Gate.

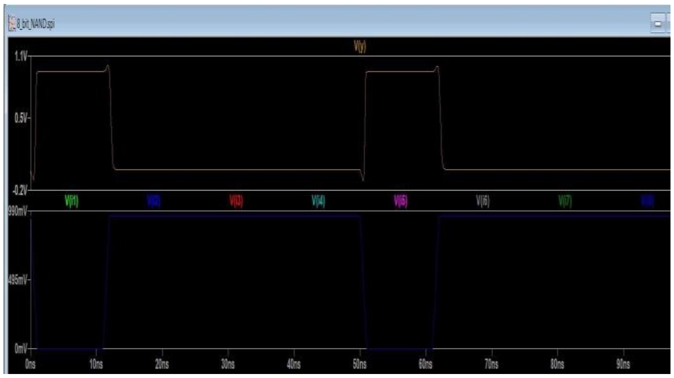


Figure 21: output of NAND

*D. 4-bit NOR*

Shown in the figure below is a simulation for a NOR Logic Gate.

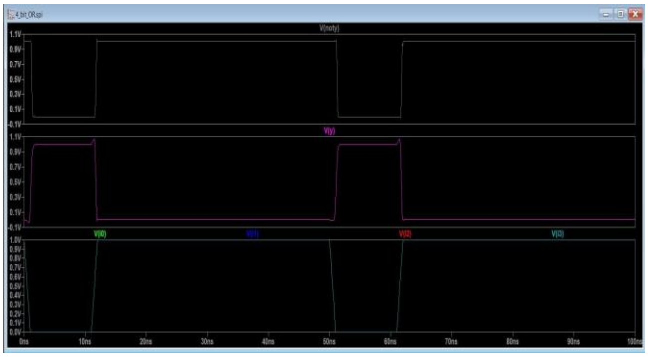


Figure 22: output of NOR

.

*E. 3x8 decoder*

A Decoder is a logic circuit that takes (n inputs) & has (2^n outputs), in our case, a 3x8 Decoder has 3 inputs(A2-0) & 8 outputs (D7-0) , only one output should be high at any time, and it is decided based off the decimal number that the inputs converge to. A simulation for a 3x8 Decoder is shown in the figure below.

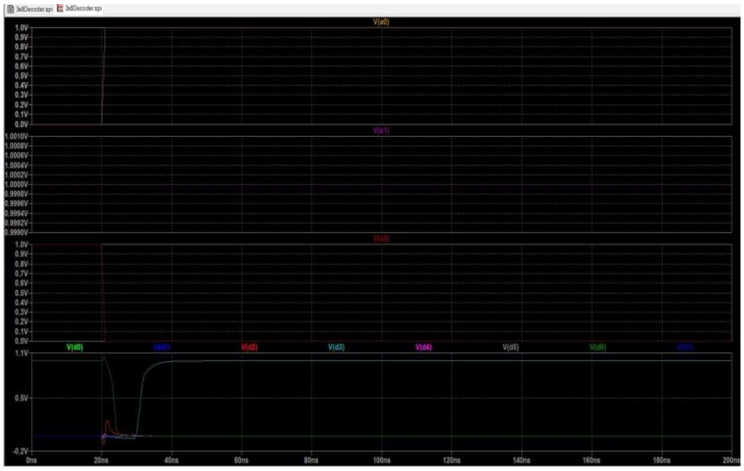


Figure 23: output of 3x8 decoder

*F. 8x3 encoder*

An Encoder does the complete opposite function of a Decoder, it transforms 2^n input lines into n output lines, where only one input signal can be high at a time. Simulation for a few Encoder cases can be shown in fig below.

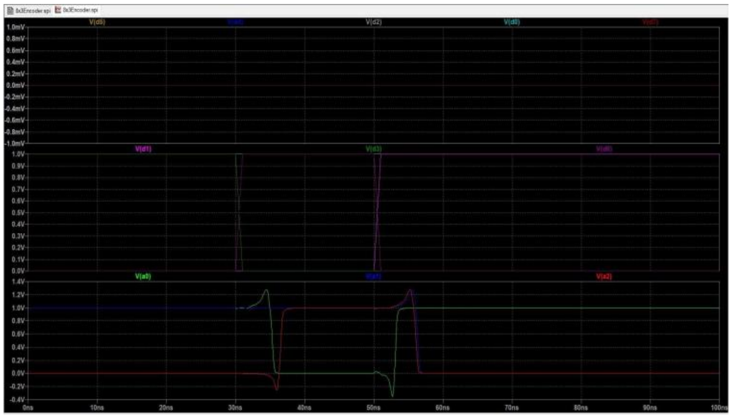


Figure 24: output of 8x3 encoder

*G. One Row CAM*

One Row CAM is a combination of 8 single bit CAMs that are treated like a group together (Byte).

This can be tested by writing a Byte of data into the 8 SRAMs included in the CAMs of the row, then try providing the same Byte & then a different Byte & notice the Match bit (Determines whether the CAM Byte is equivalent to the stored Byte). This test is shown in the figure below.

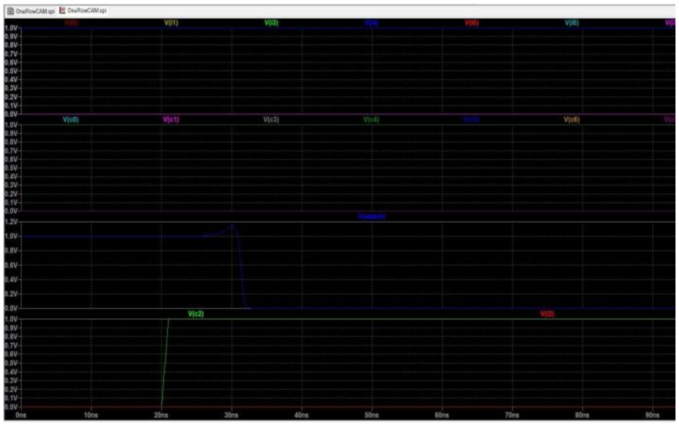


*Figure 25: output of one Row CAM*

### *H. CAM Matrix*

Finally, we can test the final Circuit which is a CAM Matrix of size 8x8, which uses a Decoder to determine which CAM Row should be written on. And the result obtained from this CAM's goes as input to Encoder to determine the index (Address) of the Match Row (if there is a match). This can be tested by writing a Byte on a specific Row, we can specify which row that is by controlling the Decoder's input (Address) & if we provide the same Byte on the CAM, the output of the Encoder should be equivalent to the Address on the Decoder input, we can also try adding a different Byte than the stored one which should result in a (000) output (automatic for no match).

This test is shown in the figure below.

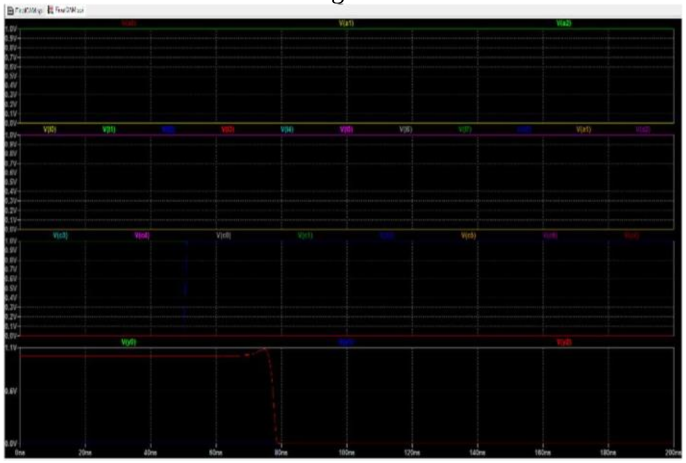
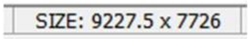


*Figure 26: output of CAM matrix*

## V. Over all Area, Delay and Power

The area for the full circuit is obtained from the tool as bellow figure:



SIZE: 9227.5 x 7726

*Figure 27. the area for circuit*

The power can be calculated as Voltage multiply by Current and the result for the whole circuit is:

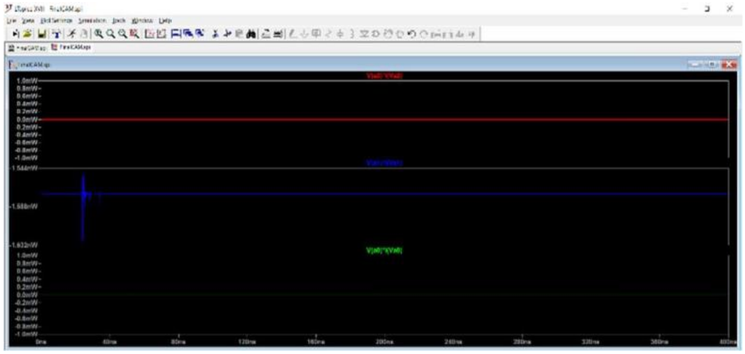


*Figure 28. the power for design*

Falling & Rising Time as well as Falling & Rising Propagation Delay can be estimated by inspection of simulations.

We use figure 29 to estimate Rise Time & Rise Propagation Delay.

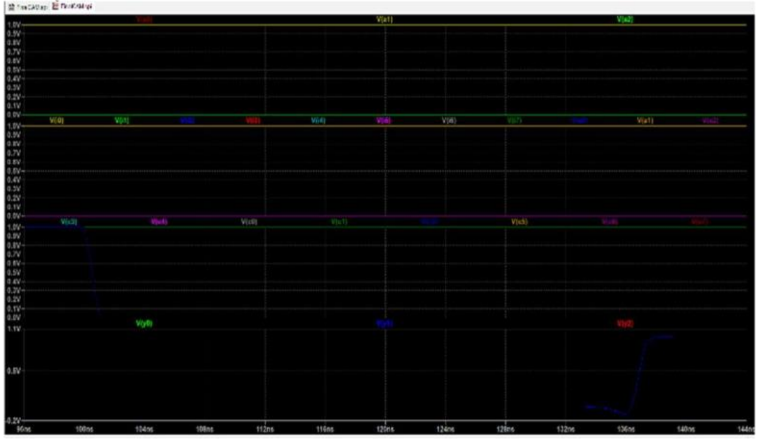


*Figure 29. Rising Time*

The Estimated Values are:

- Rising Time: 3.5ns
- Rising propagation delay: 37ns

We can use figure 30 to estimate the Falling Time as well as the Falling Propagation Delay.

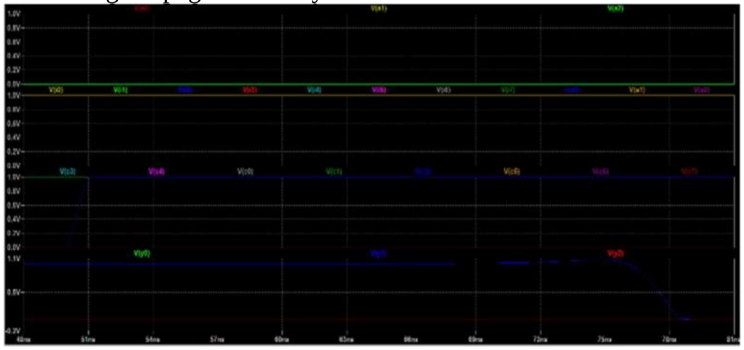


*Figure 30. Falling Time*

The Estimated Results.

- Falling Time: 3ns
- Falling propagation delay: 27ns

## VI. Conclusion

In conclusion, this project focused on the design and implementation of a cutting-edge Content-Addressable Memory (CAM) cell using a 9T Static Random Access Memory (SRAM) configuration. The objective was to explore the advantages and challenges associated with integrating 9T SRAM into CAM architecture, while optimizing for speed, area, and power consumption. In conclusion, the successful implementation of the CAM cell using 9T SRAM demonstrates the potential for improved performance and efficiency in content-addressable memory designs. This project not only deepened our understanding of CAM cells and SRAM but also provided valuable hands-on experience in digital integrated circuit design. The knowledge gained and the outcomes achieved will serve as a foundation for future advancements in memory design and contribute to the continuous evolution of high-performance computing systems. Also, we build 8-bit SRAM not 4-bit so the circuit is bigger and can save data as duple as 4-bit can do. For future use we can use our final block in complicated design such that CPU and build from this basic circuit our daily computer components.

## References

1. *https://www.semanticscholar.org/paper/Full-Swing-8x8-XOR-Content-Addressable-Memory-Adithya-Basha/efacb4a06aa10ccb65c69e96c1cb10c19bf4b6ff*

2. *https://www.electronics-notes.com/articles/electronic_components/semiconductor-ic-memory/static-ram-sram.php*

3. *https://www.geeksforgeeks.org/what-is-content-addressable-memory/*

4. *https://www.geeksforgeeks.org/encoders-and-decoders-in-digital-logic/*

5. Low-power Content Addressable Memory (CAM) Array for Mobile Devices Khader Mohammad ∗, Aziz Qaroush ∗, Mahdi Washha ∗, Baker Mohammad† ∗ Department of Electrical and Computer Engineering, Birzeit University, Ramallah, Palestine †Khalifa University, Abu Dhabi, United Arab Emirates

6. Design of Efficient Low Power 9t SRAM Cell K. Gavaskar1 S. Priya2 1P.G Scholar/VLSI Design, 2Assistant professor/ECE, Bannari Amman Institute of Technology Sathyamangalam, India.

7. https://www.sciencedirect.com/science/article/abs/pii/S0167926022001080

8. https://www.sciencedirect.com/science/article/abs/pii/S2214785321046046

9. Chat GPT to gain good information and know the differences between components.

10. https://www.tutorialspoint.com/difference-between-ram-and-cam

11. https://link.springer.com/article/10.1007/s00034-015-0119-0