Design of 8-bit CAM using 9T SRAM

Zainab Shaabneh 1182820, Amal Ziad 1192141
Department of Electrical and Computer Engineering, Birzeit University
Integrated Circuits-ENCS3330

Abstract – *Content Address Memory (CAM) is a specialized memory that functions like dedicated search engines and is used in various applications such as networking and database management systems. CAM implemented with 9T SRAM technology is popular due to its power efficiency and compactness. We aim to create an 8-bit CAM using 9T SRAM with low power consumption and minimal delays. We will outline the design steps for building a 4-bit CAM, optimizing area and power efficiency while maintaining its search functionality.*

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

I. INTRODUCTION

Computer memory is a generic term for all of the different types of data storage technology that a computer may use. Some types of computer memory are designed to be very fast, meaning that the central A Computer memory can be categorized into two main types: primary memory (such as RAM and ROM) and secondary memory (like hard drives and CDs). [1]

A content-addressable memory (CAM) is a data storage device that stores memory in cells. Whenever any part of the memory is accessed, the CAM compares the input with all the stored data. CAMs are used in applications that require fast Research, such as extracting parametric curves, performing a Hough transform, or performing Huffman encoding/decoding. However, the CAM's speed comes at the cost of increasing silicon area and power consumption, which the designers aim to minimize. CAM works by performing an exhaustive, parallel search of the entire memory space, which allows it to match specific data in a single cycle. This is in contrast to RAM modules, which require multiple clock cycles to retrieve a single memory location. A CAM cell is created using a SRAM cell and a comparison circuit to compare the CAM data to the data stored in the SRAM. [2]

RAM, multiple clock cycles are required to find the address of the input data. In contrast, CAM memory allows for parallel searching of all addresses to retrieve the corresponding address. To construct a CAM cell using a 9T SRAM, an XOR gate is used as the comparison circuitry. This XOR gate is built using two pass gates instead of two AND gates and one OR gate, resulting in reduced delay, area, and power consumption due to the use of only four transistors instead of twelve. Arranging the bits in a suitable placement, a four by four bits CAM cell can be created to achieve better optimization. Implementing CAM can be done using the SRAM schematic since it performs the same operations as SRAM. Furthermore, SRAM can be implemented using different numbers of transistors, such as 6T, 8T, 9T, and 10T, depending on the requirements and specifications of the application or project. [3]

The implementation of CAM can be done using the SRAM scheme because it performs the same operations as SRAM. In addition, SRAM can be implemented using different numbers of transistors, such as 6T, 8T, 9T, and 10T, all depending on the

requirements and specifications of the application. In our project, a CAM will be designed and implemented using 9T SRAM. The following figure (Fig. 1) shows the 9T SRAM diagram. [4]

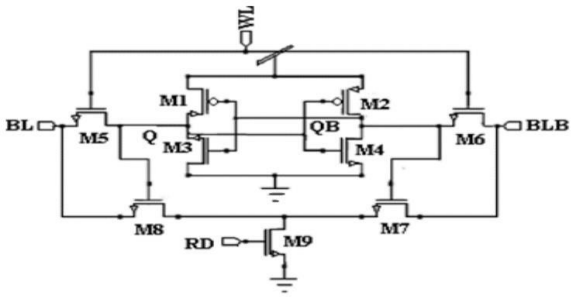

Figure 1 Schematic of 9T SRAM Circuit

II. DESIGN AND IMPLEMENTATION

In our project, we will implement 8-bit CAM using 9T SRAM; several components are required to be designed. These components are mentioned as following:

A. 9T SRAM

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

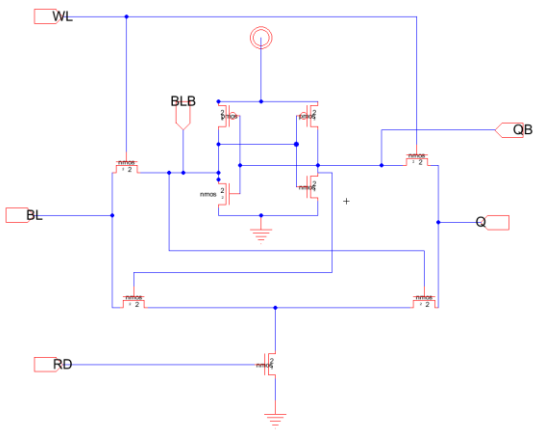

Figure 2 Schematic of 9T SRAM circuit.

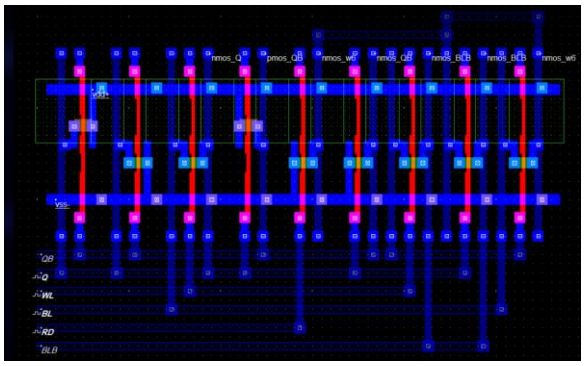

Figure 3 Layout of 9T SRAM circuit

B. 3x8 Decoder

In the 8-bit CAM implementation, the 3x8 decoder is needed, because it helps in selecting the appropriate 1-Bit CAM cell from the 8-bits for the search operation.

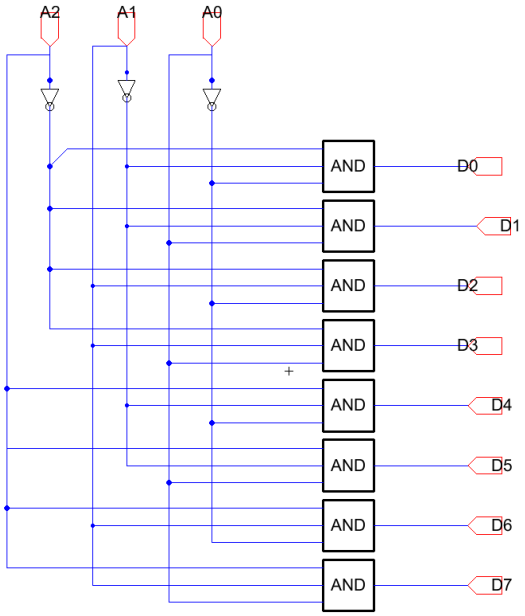

Figure 4 Schematic of *3x8 Decoder circuit*

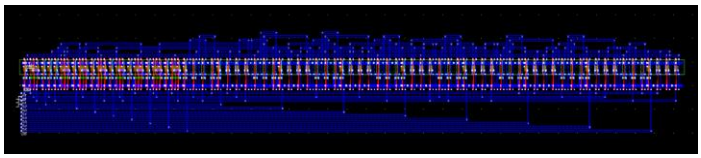

Figure 5 Layout of *3x8 Decoder circuit*

C. 1-Bit CAM

The circuit of 1-bit CAM was implemented as shown in the following circuit; two

additional invertors was added, and two pass gates for the comparison operation. And finally, the output of the circuit that represents the match signal.

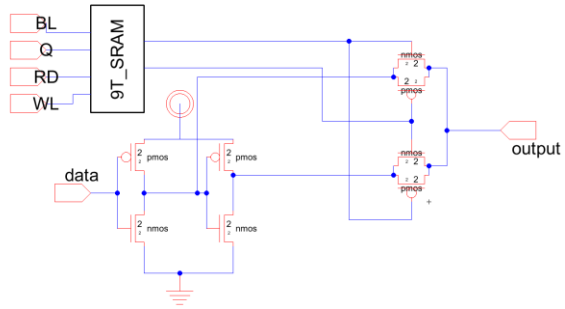

Figure 6 Schematic of 1Bit CAM circuit

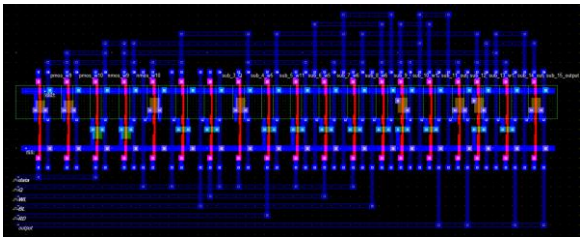

Figure 9 The Layout of 1 Bit CAM circuit

D. 8-inputs NAND gate

The 8-inputs NAND gate implementation will be used when implementing the 8-Bit CAM when connecting the 8 blocks of the 1-bit CAM to the final output.

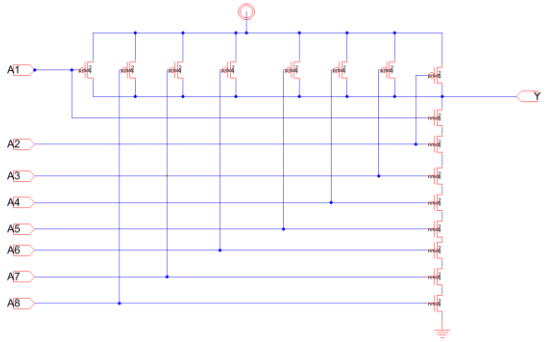

Figure 11 The Schematic of 8-NAND circuit

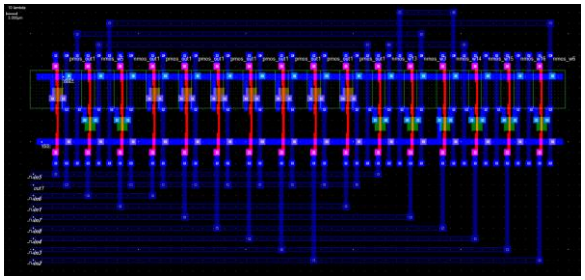

Figure 7 Layout of 8-NAND circuit

E. Invertor

An important component needed for the 8-bit CAM circuit is the inverter circuit are in the following figures:

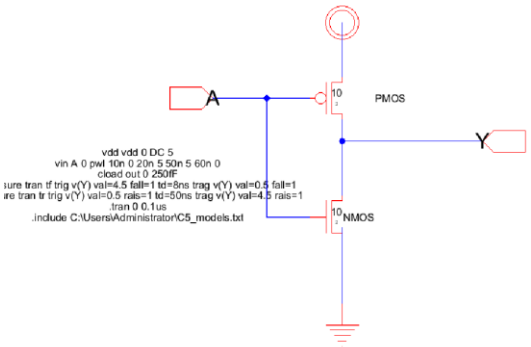

Figure 8 Schematic of invertor circuit

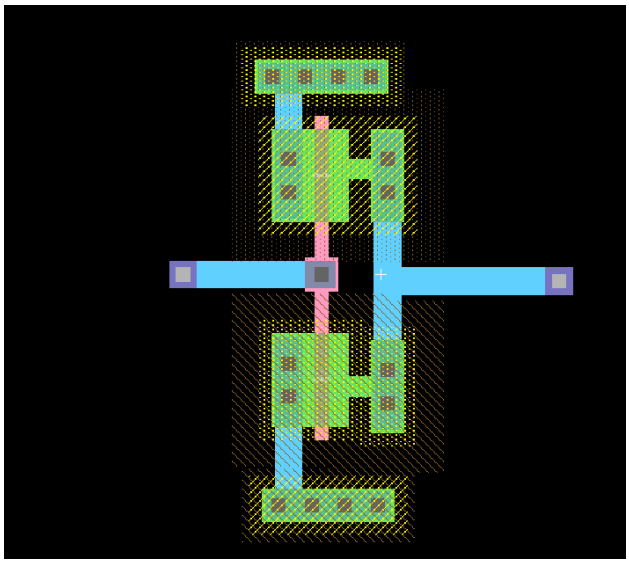

Figure 9 Layout of invertor circuit

F. 3-Input AND

Another important component needed for 8-bit CAM circuit is the 3-AND as shown:

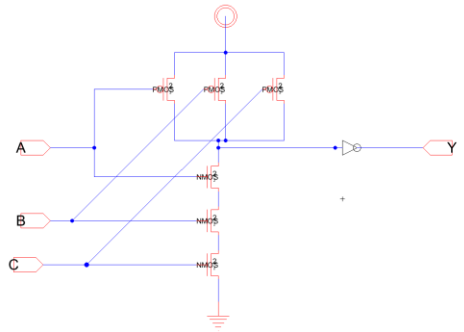

Figure 10 Schematic of 3-input AND circuit

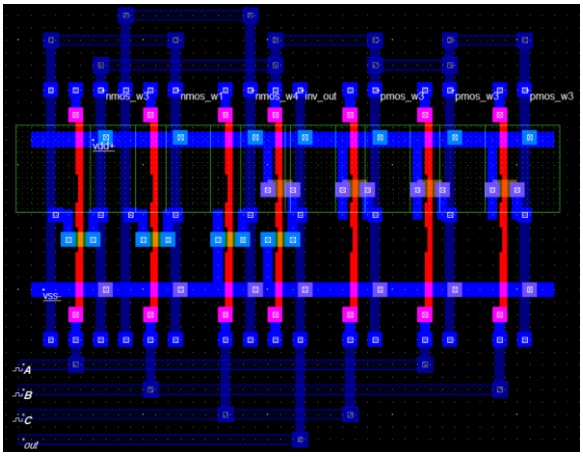

Figure 11 Layout of 3-input AND circuit

G. 4-Bit CAM

The circuit of the 4-Bit CAM using 9T SRAM was implemented as shown in the following figure, the decoder, and 1-bit CAM, NAND gates, and the invertor were used to design and implement to find the output of the 4-Bit CAM.

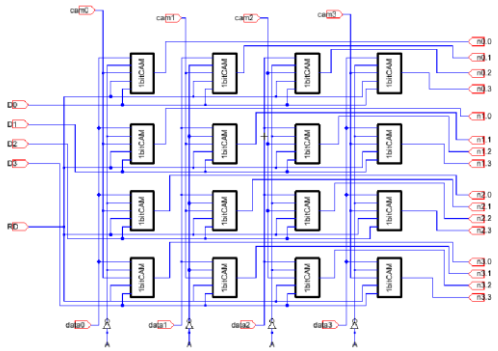

Figure 12 Schematic of 4-bit CAM circuit

H. 2 of 4-Bit CAM

This circuit consists of 2 uses of 4-bit CAM, which are sharing the same main inputs, and their outputs are the inputs of 8-input NAND. This whole block will be used to make the deal of 8-bit CAM easier to implement and connect.

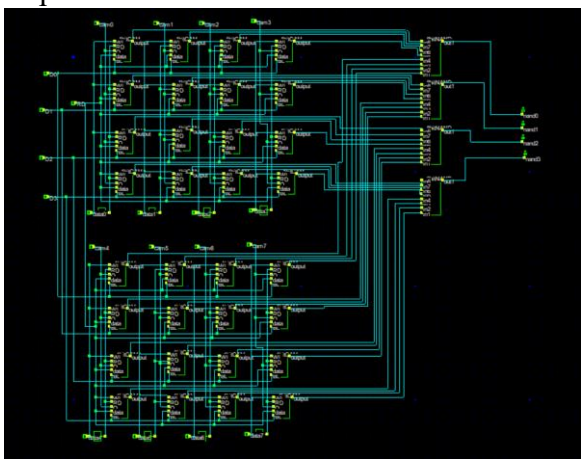

Figure 13 Schematic of 2 of 4-bit CAM circuit

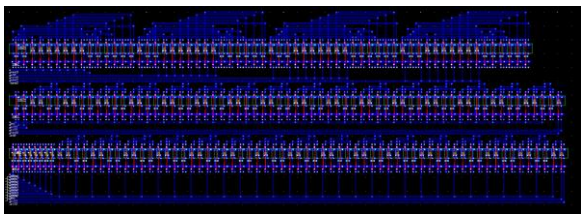

Figure 14 Layout of 2 of 4 bit CAM circuit

I. 8-bit CAM

Finally, the circuit of the 8-bit CAM using 9T-SRAM was implemented from 2 blocks each one containing 2 of 4-bit CAM with NAND gates for their outputs, including 3x8 decoder, NAND gates, and the invertor were used to design and implement to find the output of the 8-bit CAM.

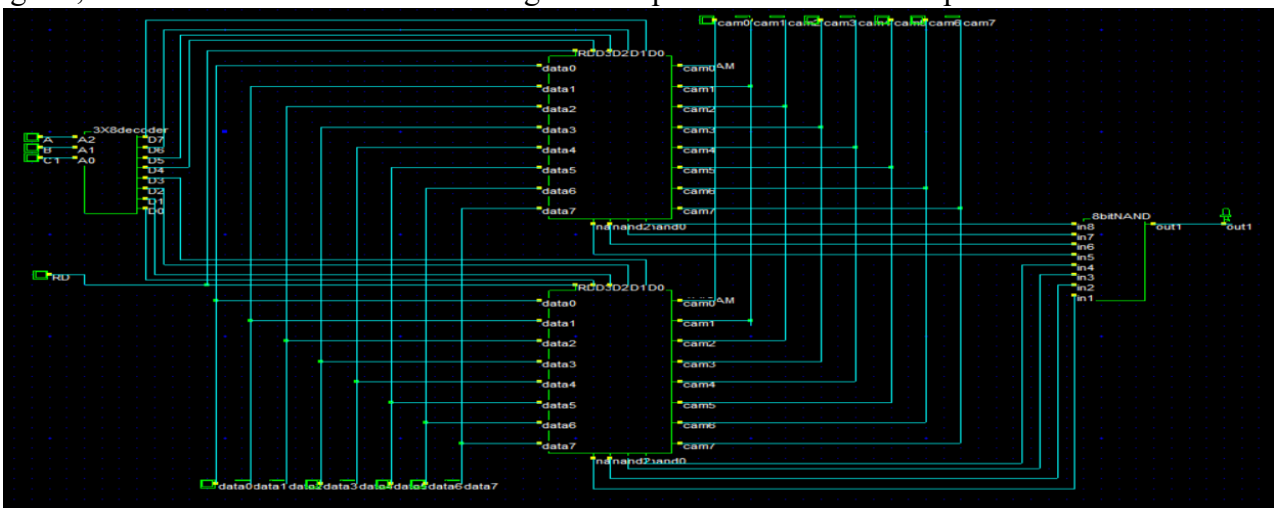

Figure 15 Schematic of 8-bit CAM circuit

III. AREA, POWER, AND DELAY OPTIMIZATION

To optimize the design of a 9T SRAM memory using a 14nm process technology and an 8-bit (CAM), we need to consider the size of the CMOS transistors.

Transistor	Delay(ns)	Power Consumption(uw)
9	0.054	1.23
12	0.27	1.23
12	0.054	1.23
6	0.581	1.39
6	0.054	1.23
14	0.367	2.56
2	0.533	0.12

Typically measured in units of lambda (λ). The optimal transistor sizes for power efficiency and performance are determined through extensive research and experimentation.

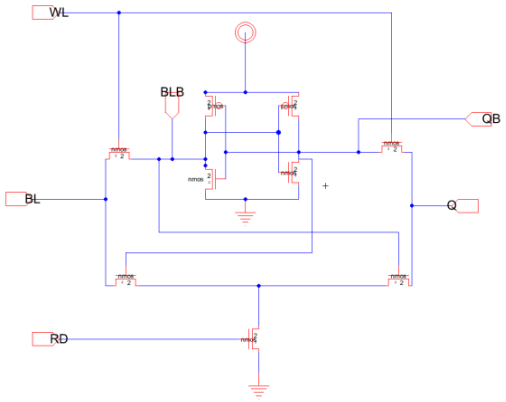

Figure 16 Sizing, and power optimization of SRAM

In our pursuit of area and cost optimization, we have implemented various strategies to minimize the number of transistors and silicon area while maintaining efficient output values and circuit performance. One notable approach involves utilizing a 3x8 decoder circuit. This decoder configuration is composed of 8 AND gates and 3 inverters, resulting in a total of 28 gates, as depicted in the accompanying figure. By adopting this design, we have successfully reduced the overall resource utilization without compromising the effectiveness of the output or the circuit's performance.

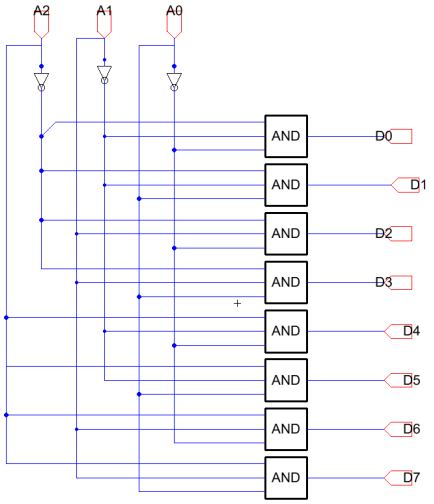

Figure 17 Schematic 3x8 Decoder

IV. Simulations and Results:

A. 9T SRAM Simulation

In SRAM memory, it is the write operation performed when writing a line, also known as writing line, empowerment, or WE, is in high case. This line serves as a memory control signal, Determine whether memory cells are in a file Read-only or write mode. When the line of writing High, memory cells can receive and store the new data. Conversely, when the write line is low, the Memory cells are in any "read only" state attempts to write to memory will be ignored. This mechanism ensures stability and integrity of data stored with dynamic allow updates as needed.

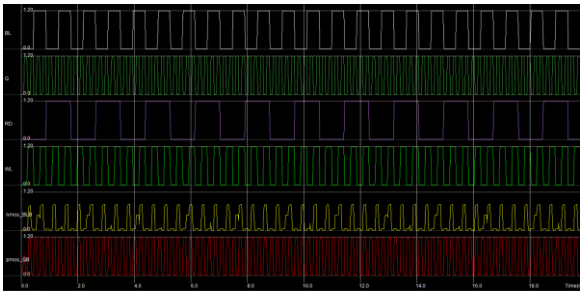

Figure 18 Simulation of 9T SRAM circuit

B. 3x8 Decoder Simulation

3x8 decoder is a combinational logic circuit that takes three binary inputs and produces eight outputs corresponding to each possible combination of the input bits. When the input bits are "low, low, low", the decoder outputs a logical "high" signal at the first output, referred to as D0, while the remaining seven outputs, namely D1, D2, D3, D4, D5, D6, and D7, are all set to logical "low". This specific behavior of the 3x8 decoder is due to its truth table, which defines the output of the decoder for each possible input combination.

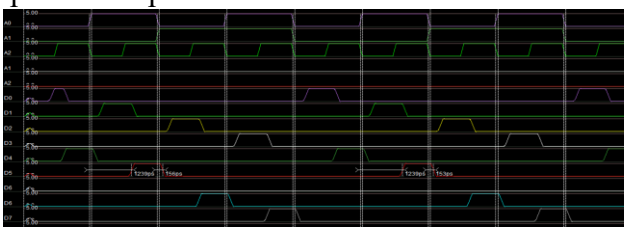

Figure 19 Simulation of 3x8 Decoder circuit

C. 1-bit CAM

In 1-bit CAM, the output demonstrated the ability of the 1-bit CAM to quickly retrieve stored information, making it suitable for various applications requiring fast and accurate data searching. As for example, when BL and CAM data are the same the match is 0 and when they are different the match is 1(active low signal), and this is exactly an XOR gate behavior that works as a comparator between BL and CAM data.

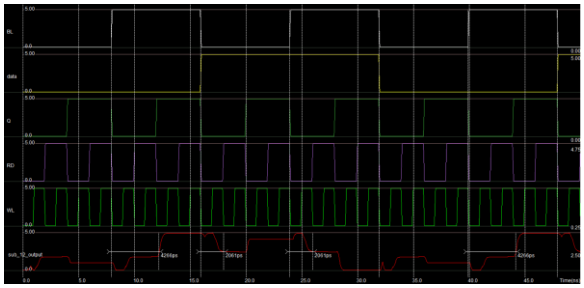

Figure 20 Simulation of 1 Bit CAM circuit

D. 8-input NAND gate

By combining eight input signals using the NAND gate, we obtained a single output signal that reflected the logical function of negating the conjunction of all the inputs.

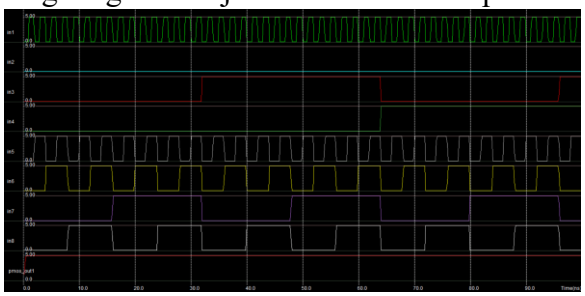

Figure 21 Simulation of 8-NAND circuit

E. 3-input AND

By combining 3 input signals using the AND gate, we obtained a single output signal that reflected the logical function of the conjunction of all the inputs.

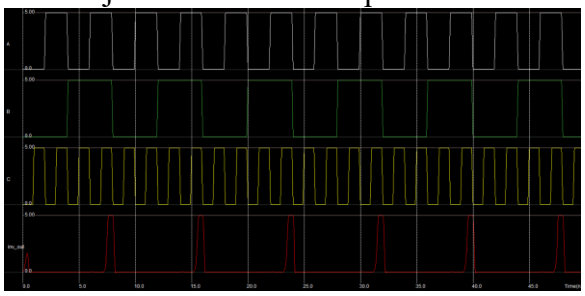

Figure 22 Simulation of 3input-AND circuit

F. 2 of 4-bit CAM

The result shows the behavior of the 4 outputs of this block while getting outputs of each 1-bit CAM, then combining all of them through 8-bit NAND.

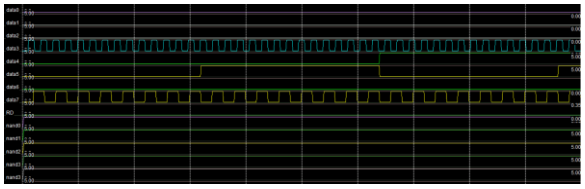

Figure 23 Simulation of 2 of 4 bit CAM circuit

V. CONCLUSION

The design and implementation steps of the 8-bit CAM circuit were thoroughly discussed and presented, providing clear illustrations of the layout and schematic circuit for each component used in the final circuit. Additionally, various techniques were employed to optimize power consumption, area, and delay for the 8-bit CAM while ensuring that the performance and searching functionality of the Content Address Memory (CAM) remained unaffected. Furthermore, the simulation results of several circuits were comprehensively represented and discussed to provide a comprehensive understanding.

VI. References

[Online]. Available: https://www.techtarget.com/whatis/definition/memory.

[Online]. Available: https://www.geeksforgeeks.org/what-is-content-addressable-memory/.

[Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiehtTq4IGAAxVhUqQEHZx1DqUQFnoECA4QAQ&url=https%3A%2F%2Farxiv.org%2Fpdf%2F1712.05096&usg=AOvVaw2cLvmcdp6nKlRFg21G3KkE&opi=89978449.

[Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj7kZKU4YGAAxUEbKQEHRWHCwEQFnoECA8QAQ&url=https%3A%2F%2Fwww.ncbi.nlm.nih.gov%2Fpmc%2Farticles%2FPMC9416021%2F&usg=AOvVaw05SsRvcG0Ntp0A5ykQwYzQ&opi=89978449.

[Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj6q6_E4YGAAxWZRKQEHRn_Ar4QFnoECA4QAQ&url=https%3A%2F%2Fwww.techtarget.com%2Fwhatis%2Fdefinition%2FSRAM-static-random-access-memory&usg=AOvVaw27D_JN06hbfDAfsF4X2zAI&op.