Design of a Full-Custom 8-Bit CAM using 9T SRAM

Abdullah Sami Naser[1], Ahmad Wasel Ghanem[2], Majd Riyad Abdeddein[3], Wasim Shaker Atta[4]
Electrical and Computer Engineering Department, Birzeit University
1201952[1],1201954[2],1202923[3],1200920[4]

***Abstract*— Think of Content Address Memory (CAM) as a fast search machine. Many systems like networks, routers, and databases use it. One interesting part of CAM is the 9T SRAM (Static Random Access Memory). More people are using it because it's efficient, stable, and compact. Plus, it's better than other parts called 6T, 7T, and 8T SRAM cells. When using 9T SRAM for CAM, we get both searching power and the benefits of 9T SRAM's efficiency, compactness, and cost savings. This paper's goal is to create an effective, swift 8-bit CAM with nine SRAM transistors. This relies on a 22 nm process file. We'll show you how to design the blueprints and layout for each piece needed to make the final 8-bit CAM. We'll pay close attention to keeping the area and power use low while still maintaining the powerful search ability of Content Address Memory (CAM).**

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

I. Introduction

THE Computer memory is key for all digital tech. One special part is the Random-Access Memory (RAM). a form of electronic computer memory that facilitates rapid access and modification of stored data, a capability crucial for efficient processing and execution of instructions. RAM has come a long way since the mid-1900s. It moved from mechanical counters to electronic storage. Nowadays, RAM is mostly made with special chips in a tech leap forward [1].

There are two main types of RAM, Static RAM (SRAM) and Dynamic RAM (DRAM). SRAM holds data as long as it has power. It's used where speed is top priority. DRAM needs to be refreshed often, but it's but is more cost-effective and offers higher storage density. It has a transistor and a capacitor in its memory cells [2].

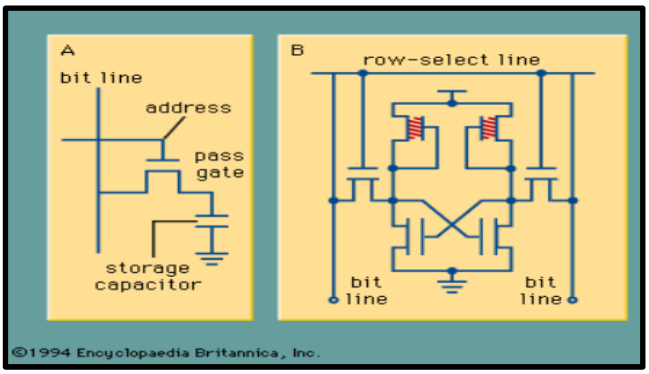

Figure 1: NMOS circuit implementation of (A) a DRAM cell, (B) an SRAM cell [2]

This paper focuses on the creative use of SRAM, specifically the 9-Transistor (9T) SRAM, in making Content Addressable Memory (CAM). CA M is a unique type of memory that operates on the principle of searching its contents to find a specific data item, rather than addressing the memory based on location. CAM works well for things that require lots of searches in a short span of time, like networking or databases [3].

The choice of 9T SRAM for building CAM systems is motivated by several factors such as advantages in terms of lower power consumption and higher stability compared to traditional 6T SRAM cells. This is crucial in CAM applications where the density of the memory array can lead to significant power usage and stability challenges. Furthermore, the 9T SRAM cell provides better noise immunity and reduced vulnerability to soft errors, which are essential attributes for reliable memory operations in high-performance computing environments.

II. CAM Functionality and its Operational Units

Think of Content Addressable Memory (CAM) as a super tool for your computer, especially handy when you need to find something fast. Unlike normal computer memory, or RAM, which requires exact memory addresses to locate data, CAM takes a different approach. In CAM, the input is the search data, and the memory system actively compares this input against all stored data. When a match is found, CAM returns the address where this matching data is located [4].

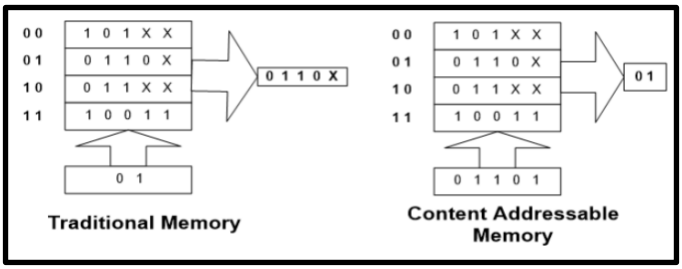

Figure 2: Read Operation in Traditional Memory and CAM [6]

This process involves the activation of a matching line when the stored data matches the incoming data. An encoder is then utilized to output an encoded representation of the match location, typically represented by log2 w bits.

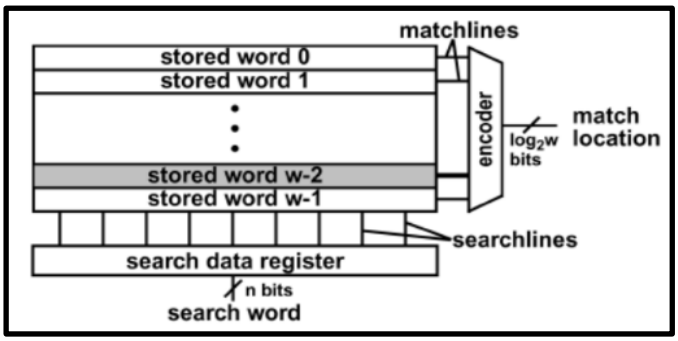

Figure 3: Content-addressable memory (CAM) conceptual view [6]

As we mentioned in the introduction. CAM can be built using SRAM with different configurations such as 6T, 8T, 9T SRAMs. Here we'll focus on the 9T SRAM since it has many advantages for a low power design. Here is some brief theory information about main operational units in designing CAM:

A- 9T-SRAM:

9T SRAM consists of Nine Transistors, including two cross-play inverters that play the Keeper's role in the Memory Unit, the other 5 transistors are used to control memory operations such as READ and WRITE. A 9T SRAM has four input ports which are BL, BLB, WL and RL. And it has two output ports also, Q and 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 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.

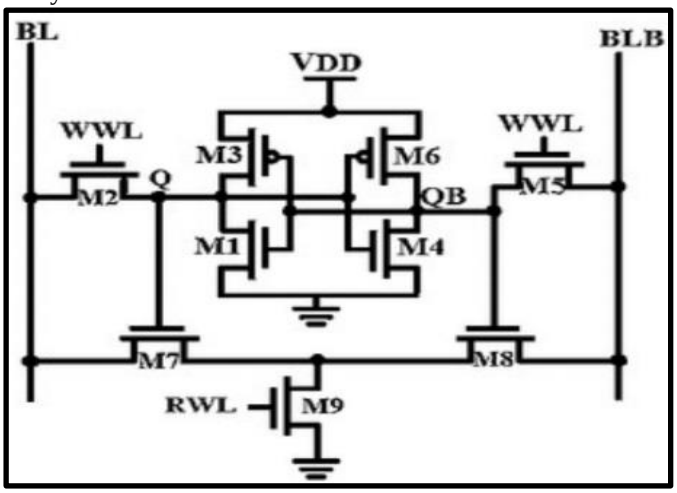

Figure 4: Schematic view of 9T SRAM

B- One Bit CAM:

One Bit Content-Addressable Memory (CAM) like a unique tool that finds stuff fast, a One Bit CAM cell has two big parts: A place where data or patterns are stored called a storage array, and a detector, or comparison circuit, that checks if the input data is the same as what's in storage. One of the key benefits of One Bit CAM is its speed. By not having to look through every piece of memory to find what it needs, it works faster and lightens the system's load.

the implementation of a one-bit CAM, each memory cell stores a single bit of data. The data storage array consists of multiple memory cells arranged in rows and columns. Each memory cell in the array holds a specific bit value, which can be either a 0 or one.

III. Design and Implementation

The design process for this project was done by using a 22nm process technology file with respect to all design rules in it. Two approaches were followed in this project, one is the hierarchal approach, and the other is flat approach. As the component needs the proper approach is used.

A- CMOS Inverter

The CMOS inverter's main functionality is to toggle binary logic. It is implemented by using NMOS and PMOS share the same input and provide the same output. The inverter is the basic element in our project since all transistors sizing in all cells depend on inverter's sizing to achieve approximately rise time equals the full time. The figures below show the schematic and layout of the CMOS inverter with proper sizing.

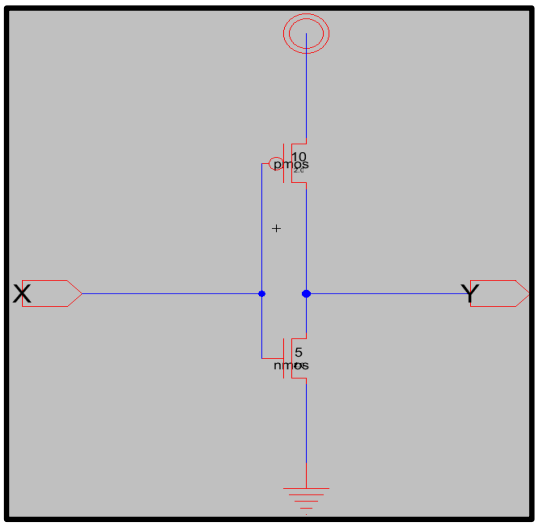

Figure 5: CMOS Inverter Schematic and Symbol view

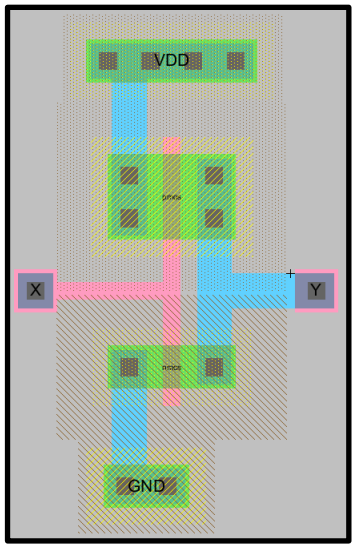

Figure 6: CMOS Inverter Layout

B- 3-bit NOR

The 3-bit NOR gate was used in this project in the implementation of DCDR. CMOS NOR consists of three parallel NMOS and three series PMOS Transistors. The figures below show the schematic and layout view for this gate.

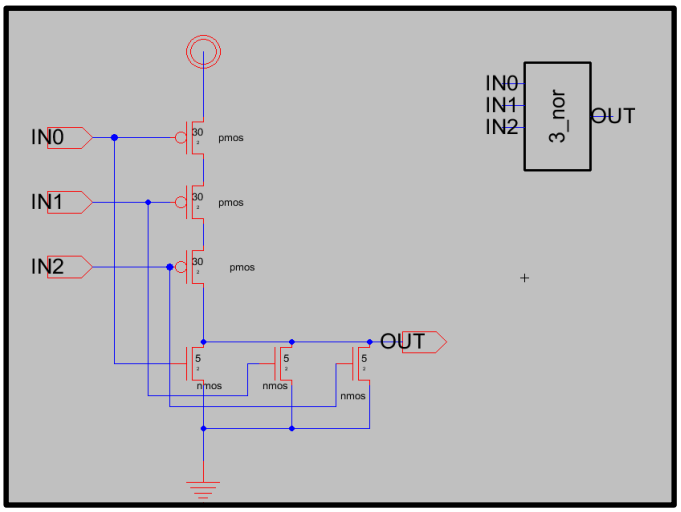

Figure 7: 3-input NOR Schematic and Symbol view

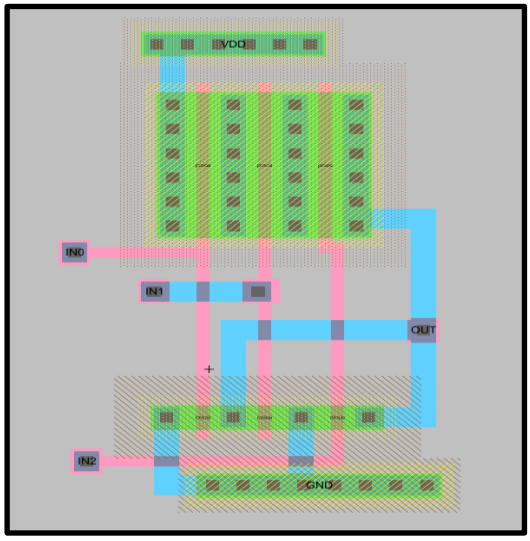

Figure 8: 3-input NOR Layout view

C- 3x8 Decoder

In the CAM 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 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 as shown in the figures below.

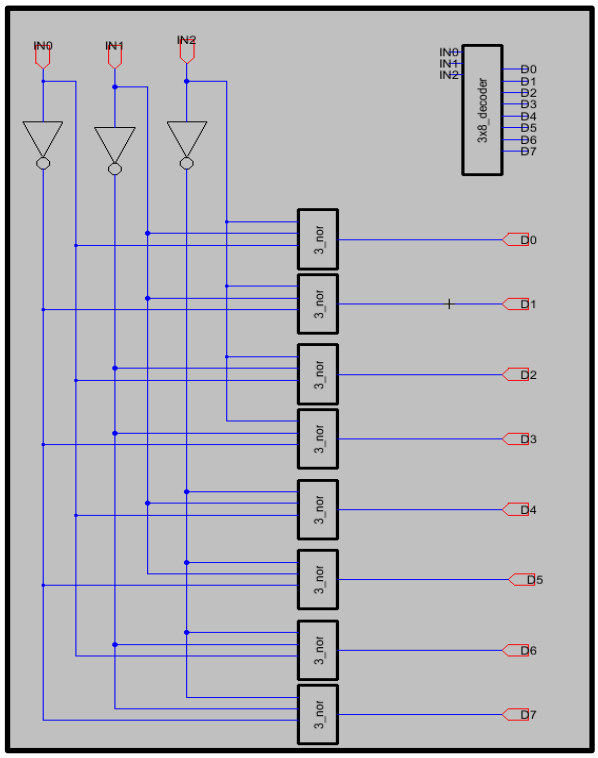

Figure 9: 3x8 Decoder Schematic and Symbol View

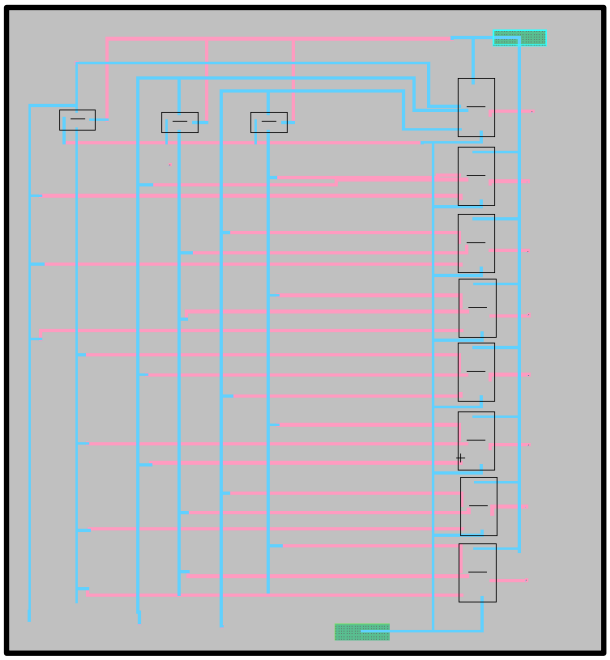

Figure 10: Figure 9: 3x8 Decoder Layout

D- 9T-SRAM:

As described in the theory part about 9T SRAM. It was built to match the schematic provided there with proper transistors sizing. The figures below show 9T SRAM schematic and layout.

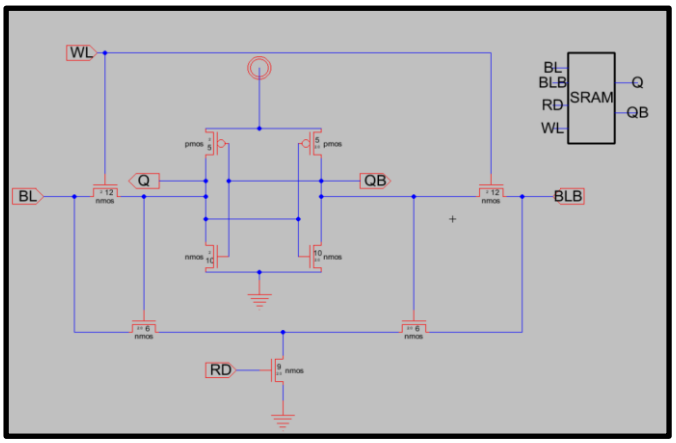

Figure 11: 9T-SRAM Schematic and Symbol View

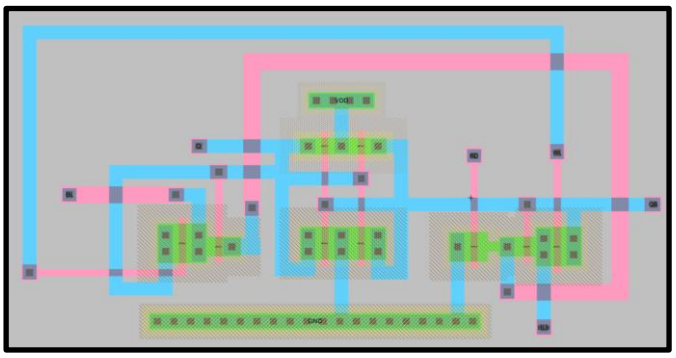

Figure 12: 9T-SRAM Layout

E- One-bit CAM:

One-bit CAM was implemented as described in the theory part by using 9T SRAM. and then it is converted to a cell in order to use it in hierarchical approach for the final connections of the full system. The figures below show the schematic and the layout view.

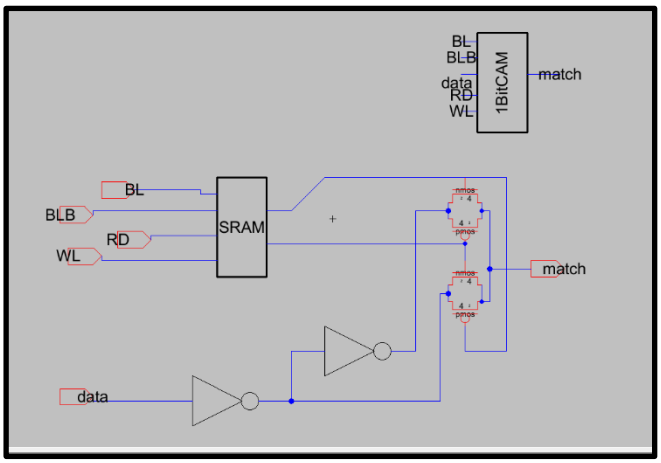

Figure 13: One-bit CAM Schematic and Symbol View

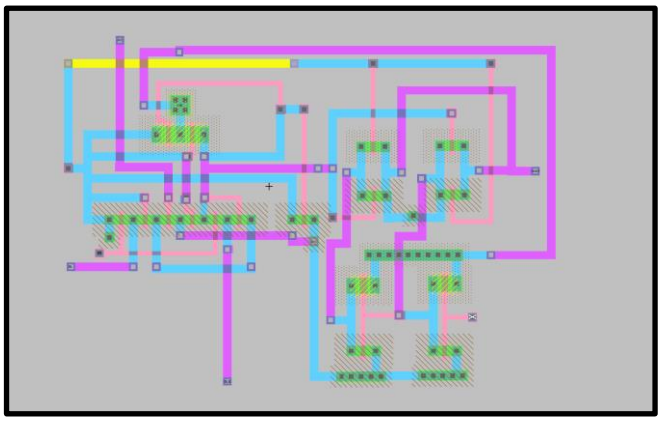

Figure 14: One-bit CAM Layout view

F- 8-input NAND:

An 8-Input NAND Gate. It's a type of logic gate that takes eight input signals. It has eight lines for input signals and just one for its output, The gate gives a low signal only when all eight inputs are high. The figures below show the schematic and the layout view of the 8-input NAND.

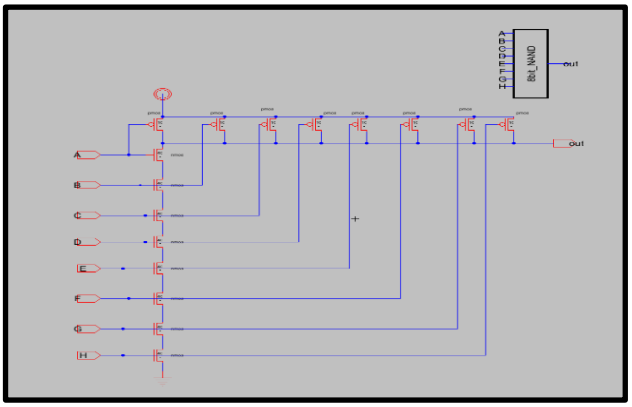

Figure 15: 8-input NAND Schematic and Symbol View

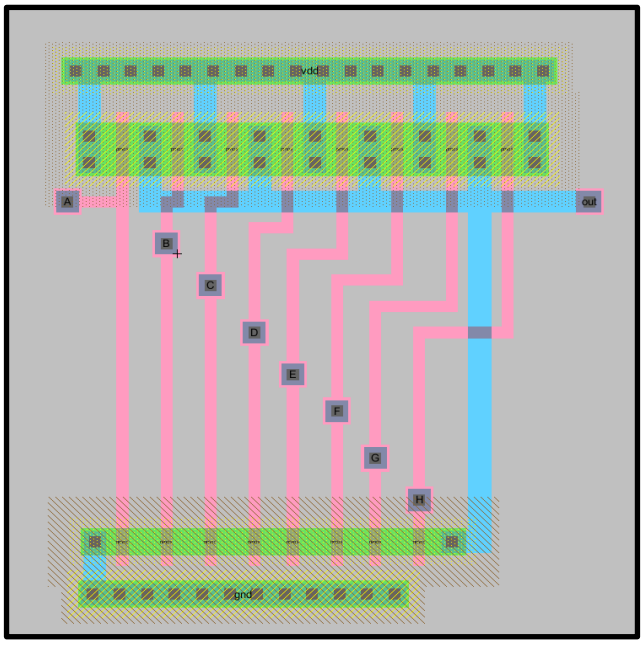

Figure 16: 8-input NAND Layout

G- 8-bit CAM Full System

As mentioned in theory about CAMs fast retrieval of data and speed, building an 8-bit CAM means making a setup of multiple CAMs. It needs to check an input data word and the stored data words together. This means quick and efficient data finding and getting. A big challenge in making an 8-bit CAM is balancing speed and storage capacity, The figures below show the schematic and the layout view of the full system where all components were combined together.

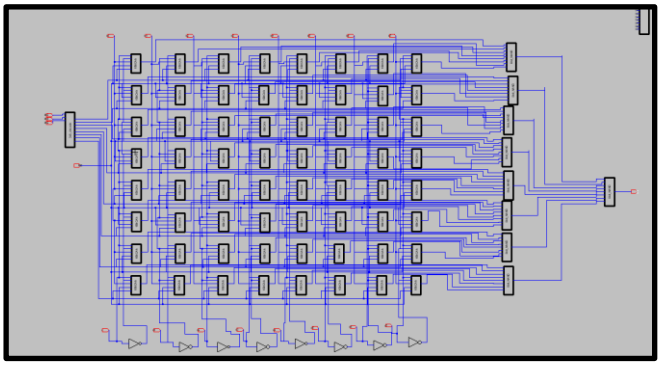

Figure 17: 8-bit CAM Schematics (full design)

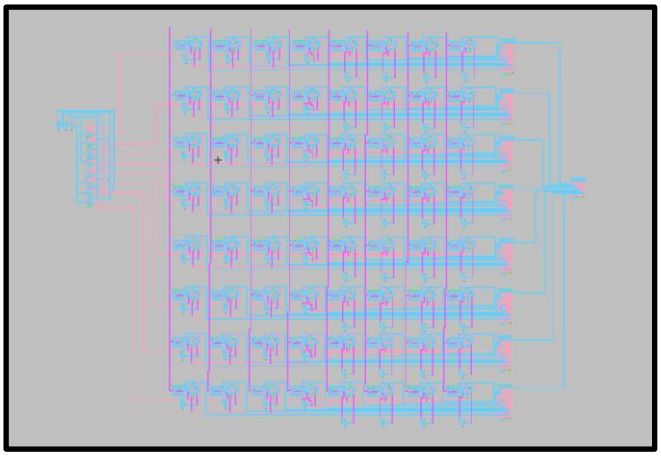

Figure 18: 8-bit CAM Layout (full design)

IV. Simulation Results

After making schematics and layout for each component in our project, we made simulation to check the functionality and timing analysis.

A- CMOS Inverter Simulation

The figure below shows the simulation results for the CMOS inverter that was presented at design and implementation part. It shows how the inverter toggles the binary logic. Moreover, it shows the rising and falling time that are equal, all of this was done by adjusting the size of the transistors.

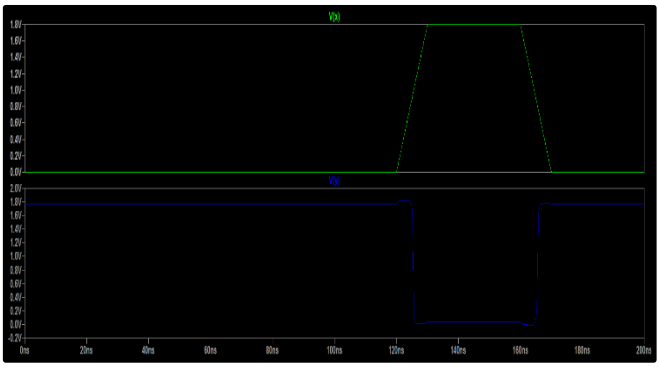

Figure 19: CMOS Invertor Simulation

B- 3-bit NOR

The simulation figure below shows the simulation of 3-input NOR gate with three inputs V(a), V(b) and V(c) and the output V(out) remains low as long as any of the inputs is high, reflecting the NOR logic where the output is high only if all inputs are low.

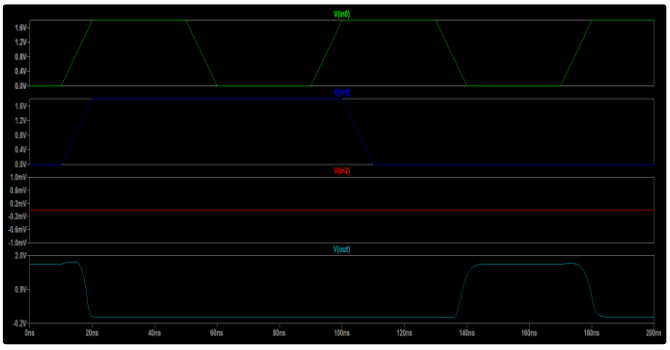

Figure 20: 3-input NOR Simulation

C- 3x8 Decoder

The simulation below shows 3x8 decoder with three inputs V(a), V(b), and V(c) and eight outputs (V(0) to V(7)) , these outputs represent the possible combinations of the three input bits, these outputs are LOW and only one line is HIGH to represent the binary value of the input bits.

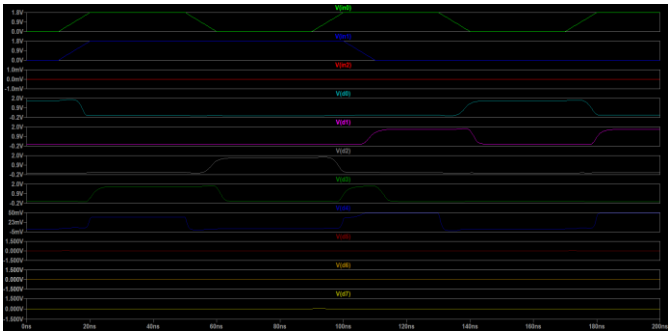

Figure 21: 3x8 Decoder Simulation

D- 9T-SRAM:

Simulation below shows the SRAM simulation, the top two lines are the bit lines (BL) and the complement of it (BLB), V(wl) is the word line, which when activated allows access to a particular row of cells in the SRAM array, and the next line V(q) being the stored data and V(qb) its complement.

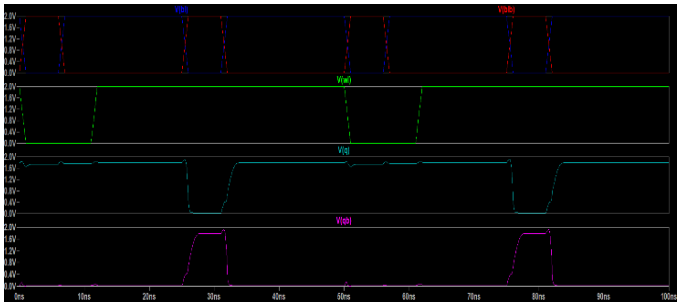

Figure 22: SRAM Simulation

E- One-bit CAM:

The simulation below shows a 1-bit CAM , and shows V(wl) is the word line it enables the access to the cell, V(bl) and V(blb) are the bit lines used for writing to or reading from the cell, they go contrary to each other during the write operation they are equal during the read operation V(match) indicates the data matches expected data, and V(data) is the data read or written dem the cell, when the V(bl), V(blb) while stable the change in V(data.

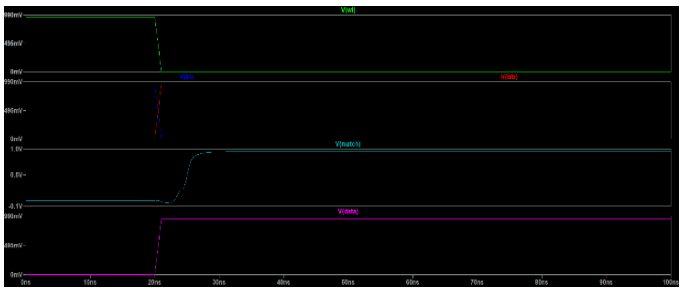

Figure 23: One-bit CAM Simulation

F- 8-input NAND:

The figure below shows the simulation if 8-bit NAND Every trace, V (0) through V(7), indicates an input of the gate We see the output, V(out), in the bottom trace This matches the NAND gate logic where a low output only occurs if all inputs are high. The subtle change in V(out) towards the end suggests that the input states shift. But as it doesn't hit 0V

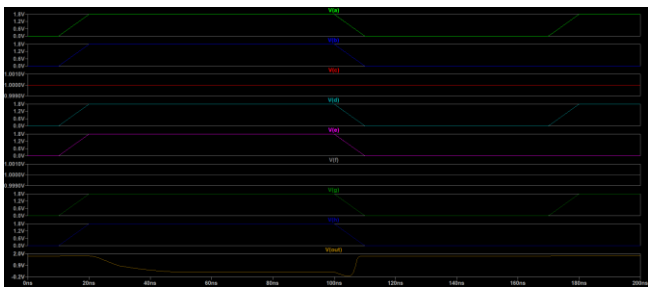

Figure 24: 8- input NAND Simulation

G- 8-bit CAM Full System:

Simulation below shows the 8-bit CAM simulation, we can see the data lines, match line, address line, control signal and the timing, with the data lines being the data being searched for and they are 8 bits in this case, match line this signal would go high if the input data matches any data stored within the CAM cells, control signal are such as Write Enable (WE), Read Enable (RE), or Search Enable might be present to control the operation of the CAM and address lines in CAM when a match occurs the corresponding address lines will indicate the location of the matching data within the memory array.

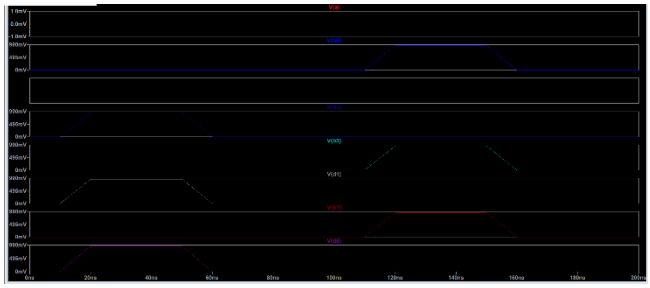

Figure 25: 8-bit CAM Simulation

V. POWER AND AREA:

Power plays a crucial role in modern integrated circuits chips. To calculate the power consumption in the gates we should have the following parameters like the supply voltage **(Vdd)** and the current **(I).** Using this formula:

$$\boldsymbol{P} = V_{dd} . I \,..(1)$$

Where:

- P is the power consumption in watts (W).
- V_{dd} is the supply voltage in volts (V).
- I is the current in amperes (A).

We can calculate the approximate dynamic and static power consumption:

$$P_{Dynamic} = C_{load} * V_{dd}^2 * f \,...(2)$$

$$P_{Static} = V_{dd} * I_{leak} \,...(3)$$

A- 6T SRAM Vs. 9T SRAM

We used 9T SRAM for the CAM system which has various advantages, for example in terms of power consumption when compared to 6T SRAM cells.[5]

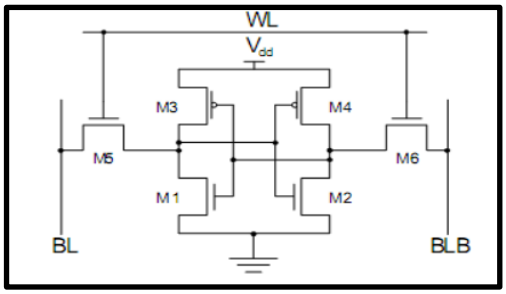

Figure 26: 6T SRAM Cell [5]

To compare the power consumption of the circuit elements, we took approximate values of the values of dynamic and static power as follows:

- $V_{dd} = 1V$
- $C_{load} = very\ small\ value$
- $F = 1GHz$
- $I_{leak} = 10\ nA$
- $22nm\ Process$

B- Power Consumption Estimation Table

Table 1: Approximate Power Estimation for basic elements

Gate	Dynamic Power	Static Power
Invertor	$P_{dynamic}$ = 1 µW	$P_{static:}$ =10nW.
3-input NOR	$P_{dynamic}$ = 2 µW	$P_{static:}$=30nW
8-input NAND	$P_{dynamic}$ = 4 µW	Pstatic: =80nW

C- Total Power Approximation in Full Design

We have 88 1- bit CAM to create the full design and each. 1-bit CAM has an SRAM which has 2 invertors and 2 external invertors so in total in the full system we have 352 invertors in the Full design Which adds up to in terms of $P_{dynamic}$ = 352 µW, Pstatic:=3520nW.

For the 3-input NOR, we have 8 3-input NOR gates in the full system which adds up to $P_{dynamic}$ = 16 µW, $P_{static:}$=240nW.And for the 8-input NAND we have 9 8-input NAND gates which adds up to $P_{dynamic}$ = 36 µW, $P_{static:}$=720nW.

We have calculated the power consumption for the Components and calculated the total power consumption for the entire system which is adds up to $P_{dynamic}$= 404 µW, $P_{static:}$ = 4480 nW.

Our team carried out a thorough careful studying of the full system that contained 352 inverters and 8 3-input NOR gates as well as 9 8-input NAND gates. This setup worked under a 22 nm process, with the specifications mentioned above.

D- Optimizing power consumption in our circuit

For Optimizing the power in our design, we can suggest these techniques:

- **Clock Gating:** Clock gating its selectively disabling the clock of certain components when they are disabled.
- **Power Gating:** Power gating it is shutting off the power of the un active components in the circuit.

Using the methods mentioned above we can make improvements to the power consumption of our design.

E- Area optimization

For Optimizing the area in our design, we can suggest these techniques, for minimizing the silicon footprint while maintaining or enhancing power efficiency.

1. Efficient Layout Design

- **Compact Cell Design:** Making sure that the 9T SRAM and other gates have minimal area and still meet the performance requirements.
- **Optimal Routing:** Minimize the routing area by using efficient wiring placement like using multi-layer routing to reduce the footprint.

2. Hierarchical Design

- **Modular Design:** Use a hierarchical design approach by creating reusable modules or blocks (SRAM, 3-imput NOR, 8-input NAND). This strategy makes the design work easier and covers the way for smoothly fitting these chunks into the bigger CAM system.

3. Optimization Techniques for Area Reduction

- **Gate and Circuit Level Optimization:** Analyze the design for opportunities to simplify logic gates and circuits. For instance, combining functions or reusing parts of the circuit for multiple operations can reduce the number of components required.
- Dynamic Reconfiguration: Implementing dynamic reconfiguration mechanisms to allow parts of the CAM to be powered down or reconfigured based on the workload can not only save power but also reduce the need for larger, more complex structures.

REFERENCES

.

[1] "Random Access Memory", Wikipedia, 21 Jan. 2024 https://en.wikipedia.org/wiki/Random-access_memory

[2] "RAM | Definition, Types, & Facts." Britannica., 21 Jan. 2024 https://www.techtarget.com/searchstorage/definition/RAM-random-access-memory.

[3] Difference between Random Access Memory (RAM) and Content Addressable Memory (CAM) – GeeksforGeeks, 21 Jan. 2024 https://www.youtube.com/watch?v=GaMR6vYeHoY&t=185s.

[4] Introduction to Content-Addressable Memory (CAM) | Kostas Pagiamtzis 21 Jan. 2024

[5]https://www.researchgate.net/publication/271304374_Speed_and_Leakage_Power_Trade-off_in_Various_SRAM_Circuits 26 Jan. 2024

[6] 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 27 Jan. 2024