

Vietnam National University Ho Chi Minh City  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY



**LAB 5: MEMORY CIRCUIT DESIGN AND CHARACERIZATION**

**Subject: Digital IC Design**

**Class L03 --- Semester 242**

**Group: 12**

<b>Student</b>	<b>Student ID</b>	<b>Mark</b>
Ho Xuan Tuan Tai	2212970	
Chau Ngoc Thach	2014526	
Mai Gia Huy	2113493	

**Lecturer: PhD. Tran Hoang Linh**

*Ho Chi Minh City, 05-2025*

## Table of Contents

EXPERIENCE 1: SRAM .....	3
1.1. Introduction: .....	3
1.2. Schematic in Cadence and operating principle .....	4
1.3. Simulation .....	5
1.3.1. Writing mode: .....	5
1.3.2. Reading Mode: .....	6
1.4. SRAM using sense amplifier .....	11
1.5. SRAM Array 8x8.....	12
1.5.1 Writing mode .....	13
1.5.2 Reading mode .....	15
EXPERIMENT 2: TCAM.....	18
2.1 TCAM cell: .....	18
2.1.1: Simulate Writing operation of TCAM cell: .....	20
2.1.2: Simulation Compare operation of a TCAM cell: .....	22
2.2 An array of TCAM sized 8x8: .....	31
2.2.1 Simulate Writing operation of an array of TCAM sized 8x8:....	33
2.2.2 Simulate Compare operation of an array of TCAM sized 8x8: 36	

## EXPERIENCE 1: SRAM

### 1.1. Introduction:

SRAM means Static Random-Access Memory which is a crucial component in every modern digital electronics. Unlike DRAM, SRAM does not require periodic refreshing, making it faster and more reliable. The SRAM which we are going to survey is the 6T SRAM. This cell comprises two cross coupled inverters for data storage and two access transistors for controlling read or write operations. This experience aims to replicate the 6T SRAM cell and survey its functionality using Cadence Virtuoso.

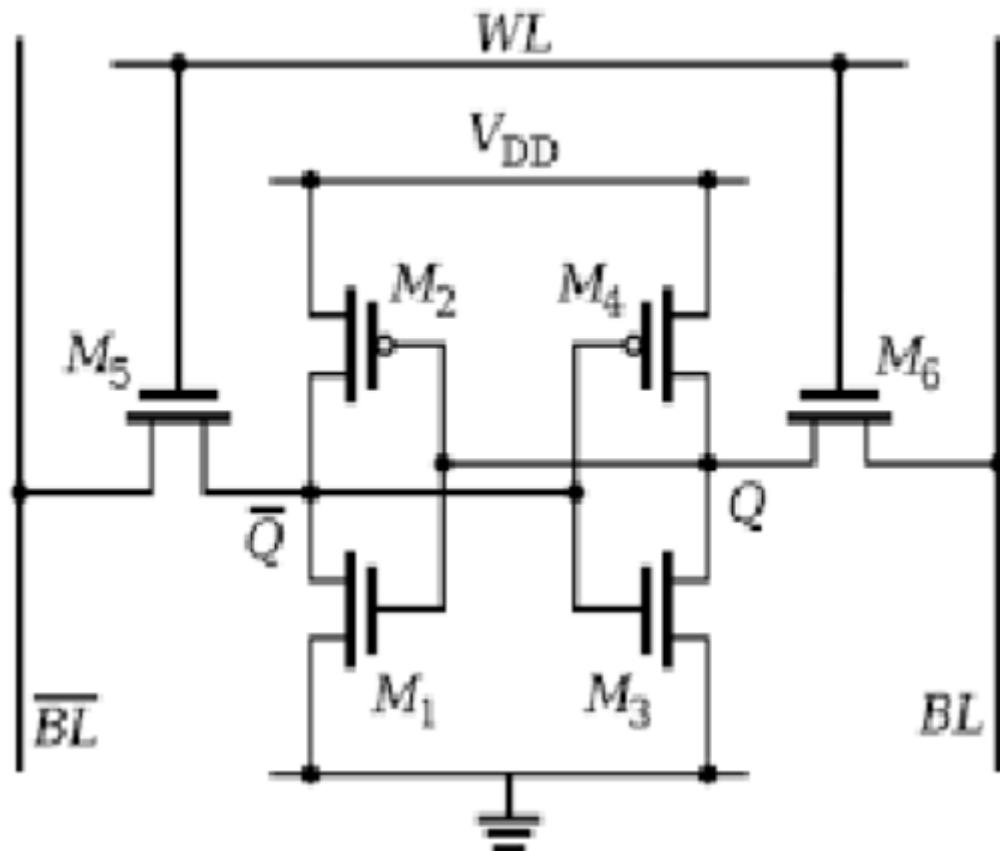


Figure 1.1.1: Schematic of 6T SRAM

## 1.2. Schematic in Cadence and operating principle

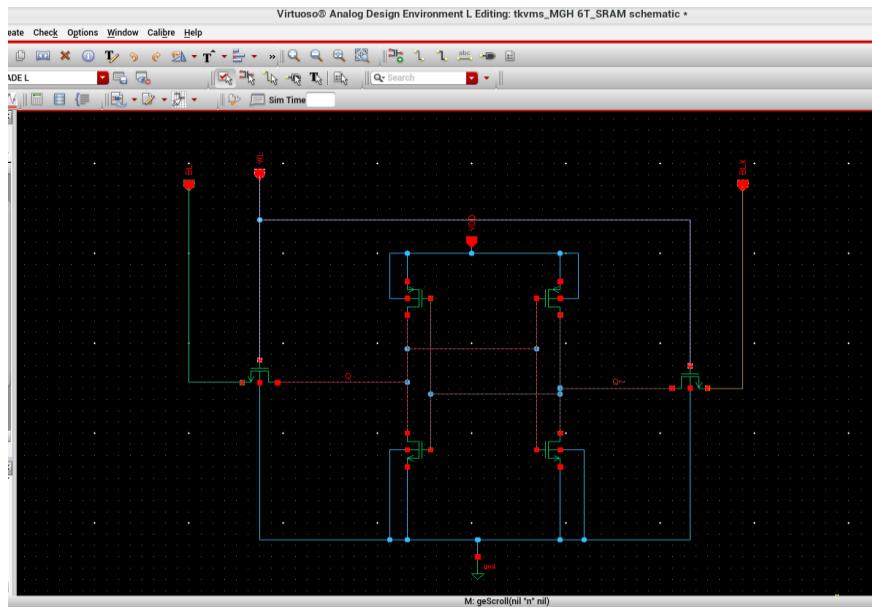


Figure 1.2.1: Schematic of 6T SRAM in Cadence

A SRAM cell has two common modes: Writing and Reading mode. The convention is that the value will be written and read at BL pin. The BLX pin will be used as an inverse reference value to BL pin

In this experience, we will do some surveys to understand how SRAM cell works in two different mode: Writing and Reading. Firstly, we need to understand the theory behind SRAM cell.

Regarding Writing mode, the pin BL and BLX have two opposite logic values, when  $WL = 0$ , the two NMOS M5 and M6 in figure 5.1.1 are off, therefore nothing happens to the cell. When we change the BL value,  $WL$  alters to 1, the old value will be replaced to make a way for the new value of BL inside the SRAM and be stored into the cell. When  $WL = 1$ , the BL value is reserved at Q. When  $WL = 0$ , the value of BL remains at Q. Whenever  $WL = 1$  again for new cycle, Q alters according to the value of BL at that time.

In terms of Reading mode, to read the value stored in the cell to pin BL we use the process of charging capacitors connected to nets BL and BLX. At first, the value stored in the two Q and Q~ point and the capacitors are charged from an external charging circuit. After charging the BL and BLX will have logic level 1. The WL pin changes to 1 that cause the 2 NMOS M5 and M6 ON. The value

in SRAM will be read to the BL and BLX pin. If the cell has the value 1, the BL pin will remain at a logic level of 1 while the BLX pins will be completely discharged to GND after the read process ends. Finally, the SRAM cell is read out exactly at the BL pin. But we can notice that the reading time can be long.

### 1.3. Simulation

#### 1.3.1. Writing mode:

Schematic for writing mode:

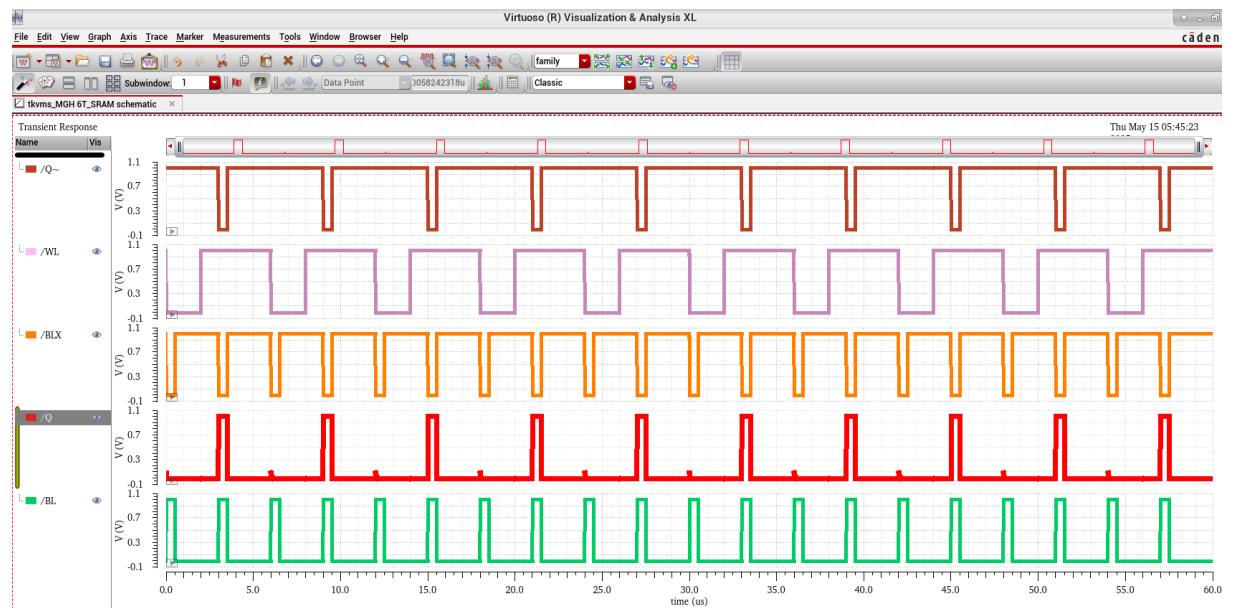


Figure 1.3.1: Simulation for SRAM writing mode

As we can see in the figure 5.3.1, when the WL is high and  $BL = 1$  from around 2ns to 3ns, the output Q equals 1 and when the WL is high and  $BL = 0$  from around 3ns to 6ns, the output Q equals 0. Therefore, we could say the result of writing operation in 6T Sram is correct.

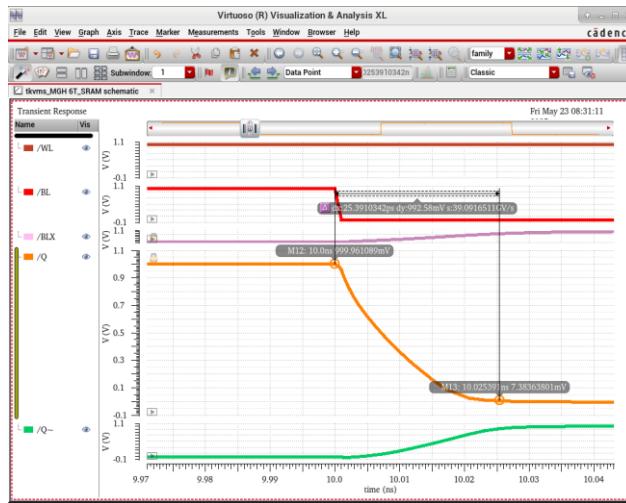


Figure 1.3.2: writing time of 6T SRAM

We also measure the writing time and the result is around 25.4 ps.

### 1.3.2. Reading Mode:

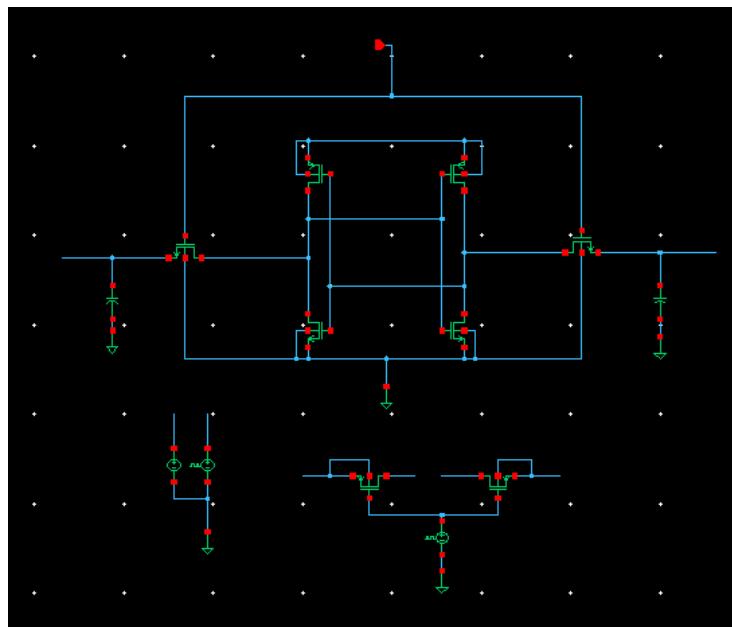


Figure 1.3.3: Schematic for Reading Operation

To Read the data stored in the SRAM, first we need to charge the capacitors connected to BL and BLX.

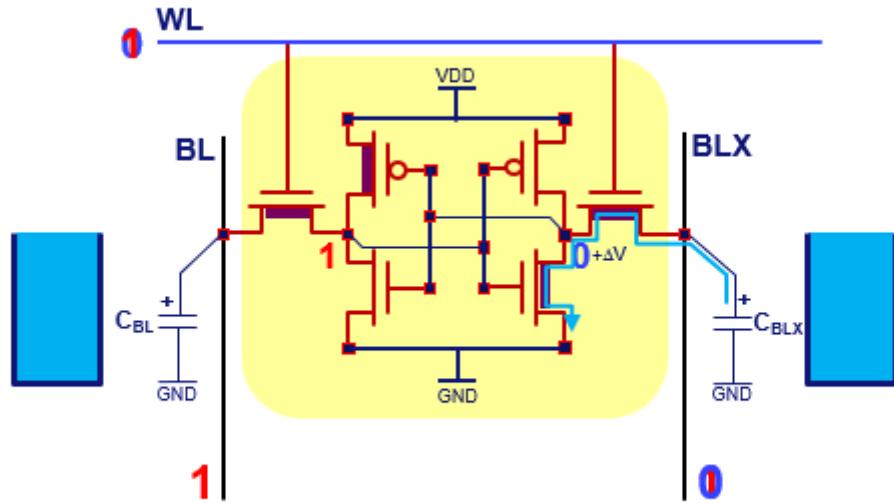


Figure 1.3.4: Reading mode explanation

In the read operation, we simulate the reading 0 and 1 respectively and then find the reading time.

- Simulating when the cell stores 0

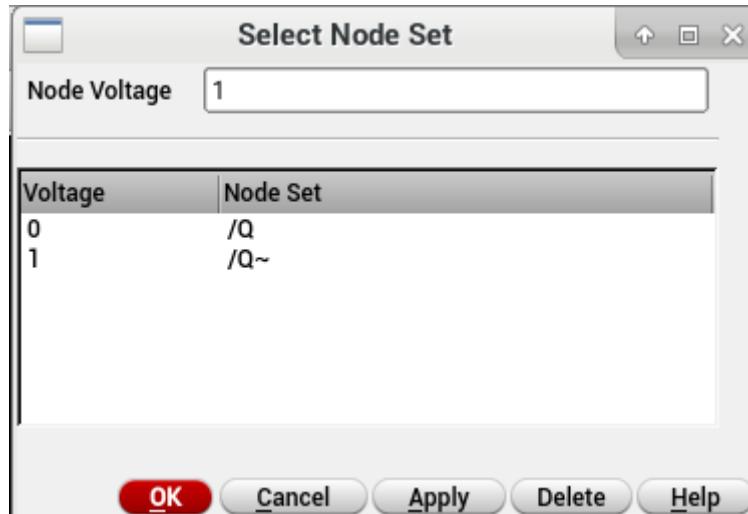


Figure 1.3.5: node set for reading 0

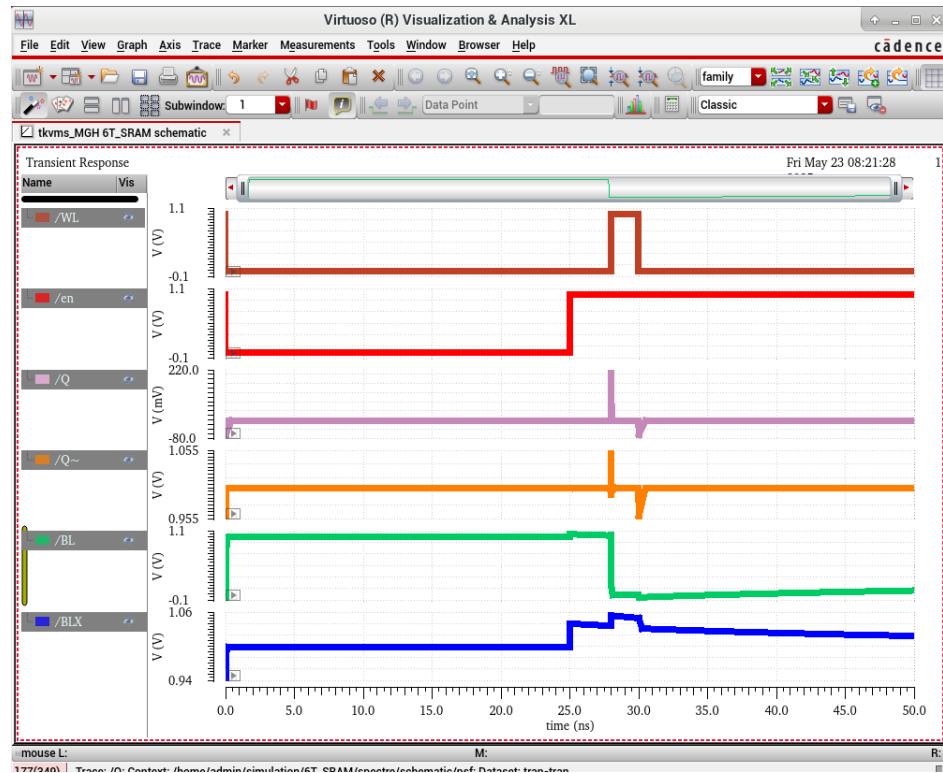


Figure 1.3.6: Result of reading 0

- Simulating when the cell stores 1

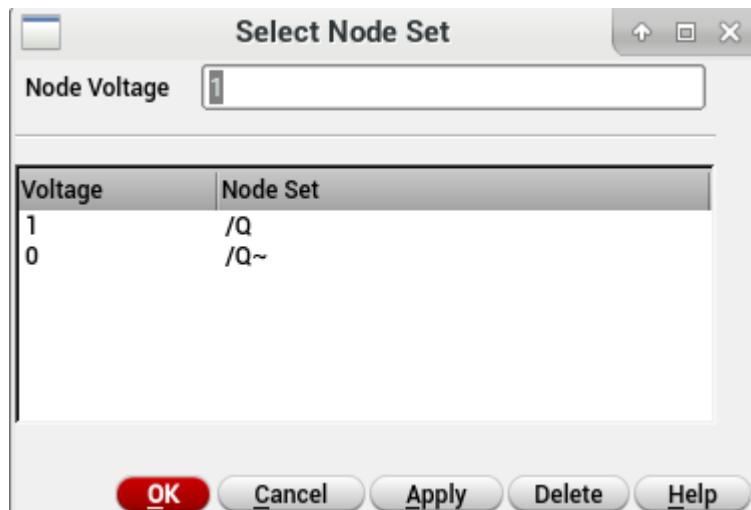


Figure 1.3.7: node set for reading 1

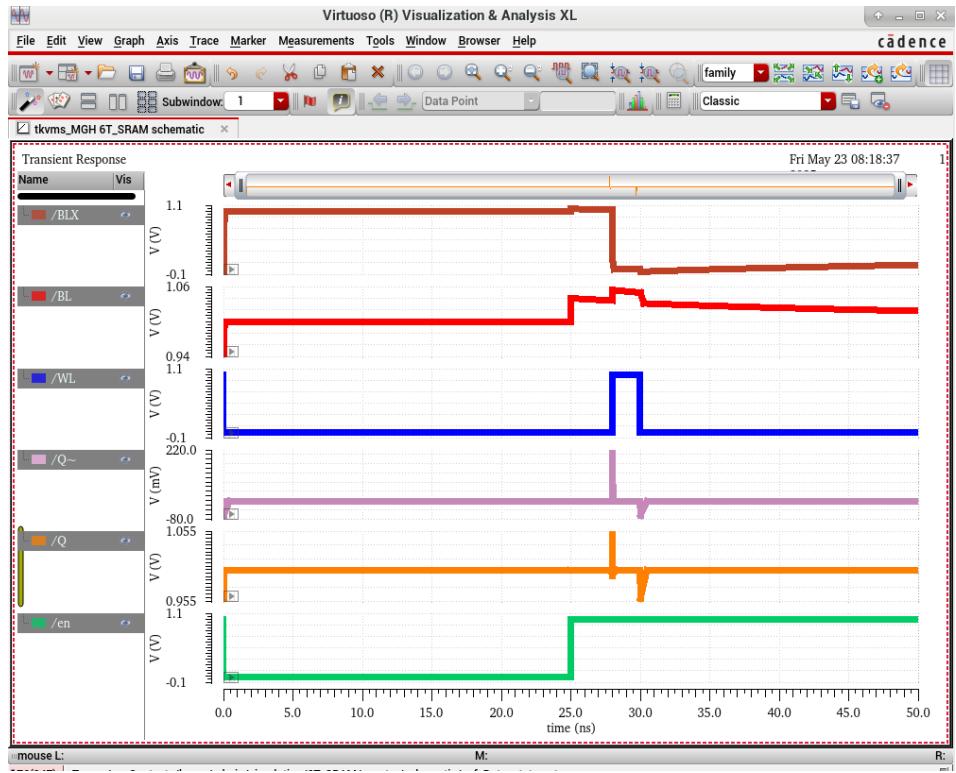


Figure 1.3.8: Result of reading 1

As we can see in figure 5.3.5 and 5.3.6, in the first 25ns we charge the capacitors and turn on the WL at around 27ns to read the data stored in the cell. Regarding the reading 0, when the WL is high the data in BL changes to low while the BLX remains unchanged. In term of the reading 1, when WL is high the BL stays at high and BLX alters to low. Therefore, we could say the simulation for reading is correct.

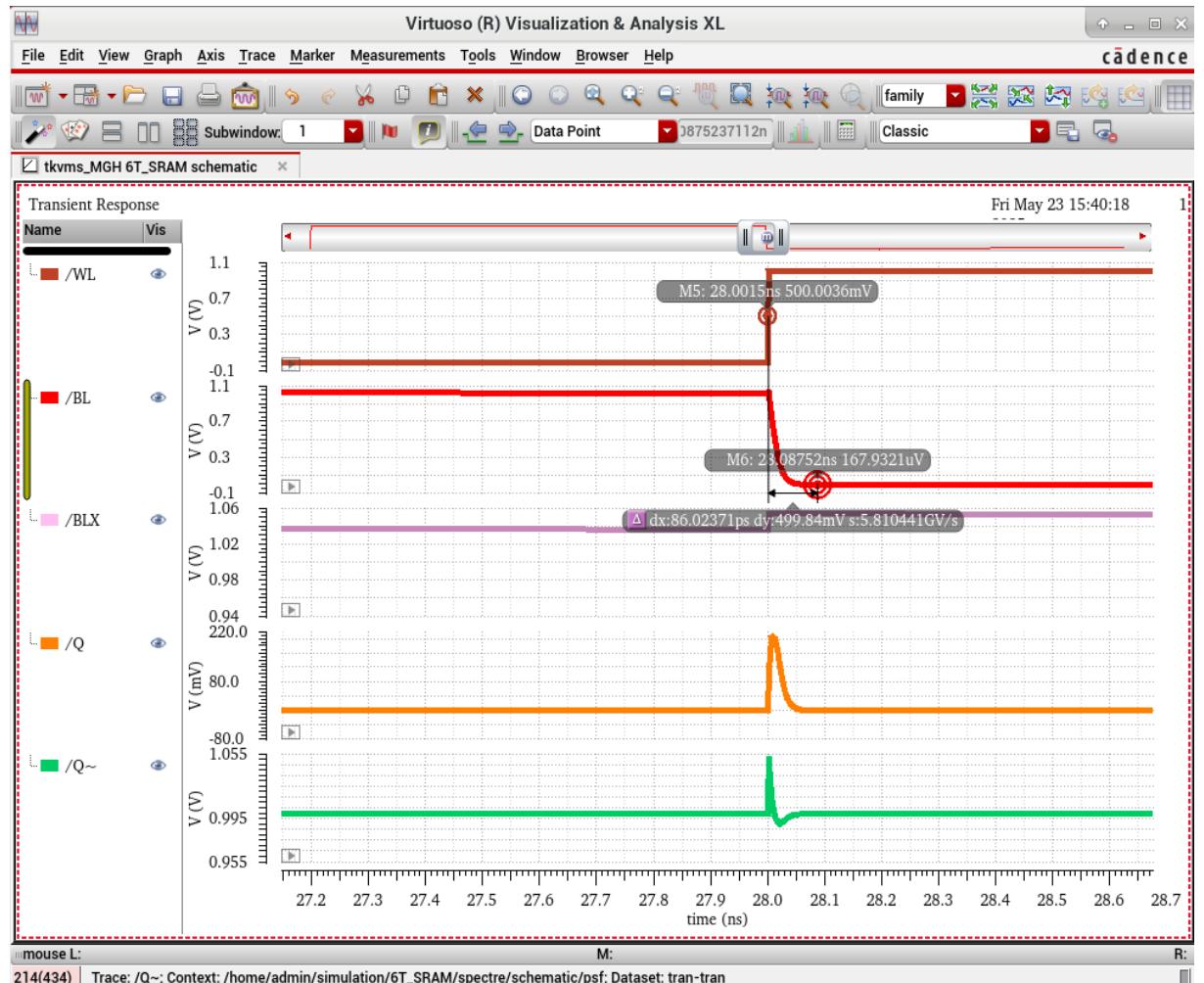


Figure 1.3.9: Reading time

After completing the simulation of reading, we turn to measure the reading time, as we can see in the figure 5.3.9, the reading time is around 86.02 ps. There are some methods that we can use to minimize the reading time such as using sense amplifier which we will examine in the next part.

## 1.4. SRAM using sense amplifier

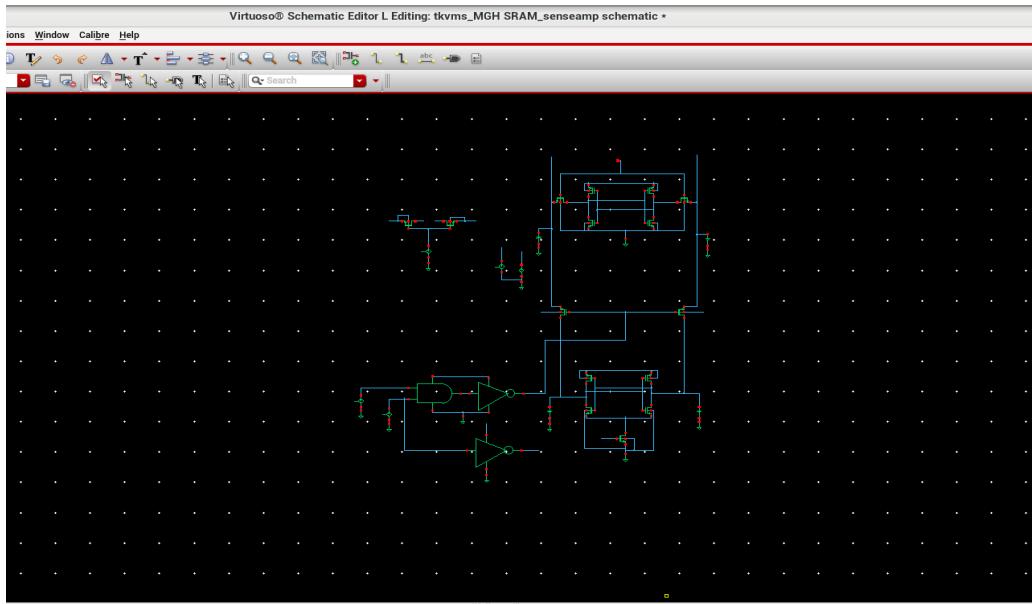


Figure 1.4.1: Schematic for SRAM using sense amplifier

In this experiment, we use the sense amplifier to reduce the reading time.

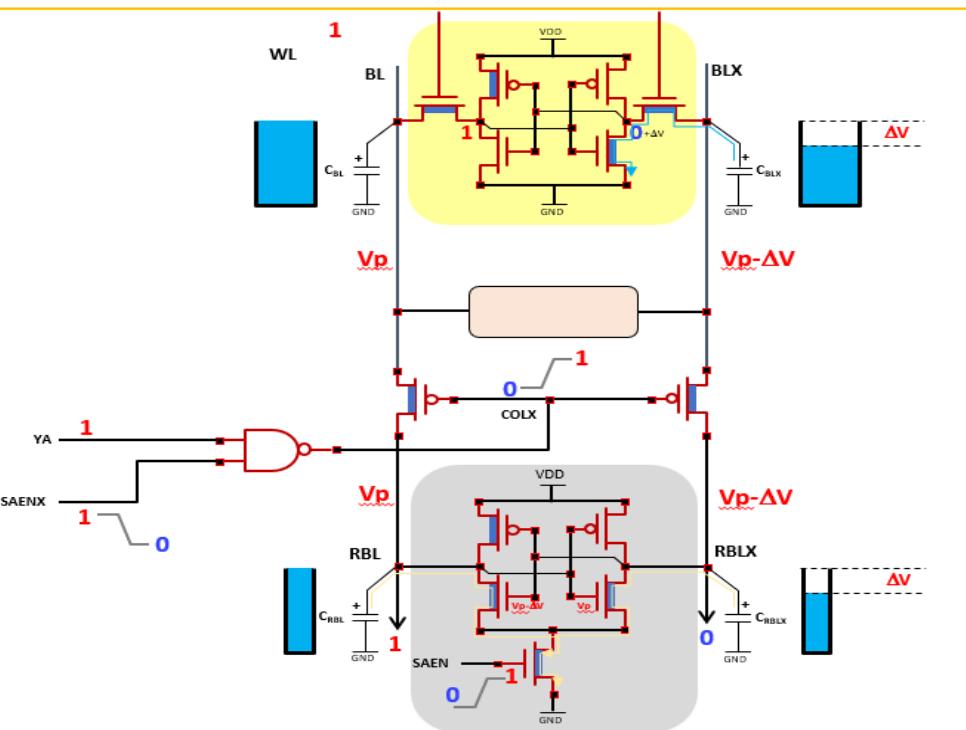


Figure 5.4.2: Sense amplifier connected to SRAM cell

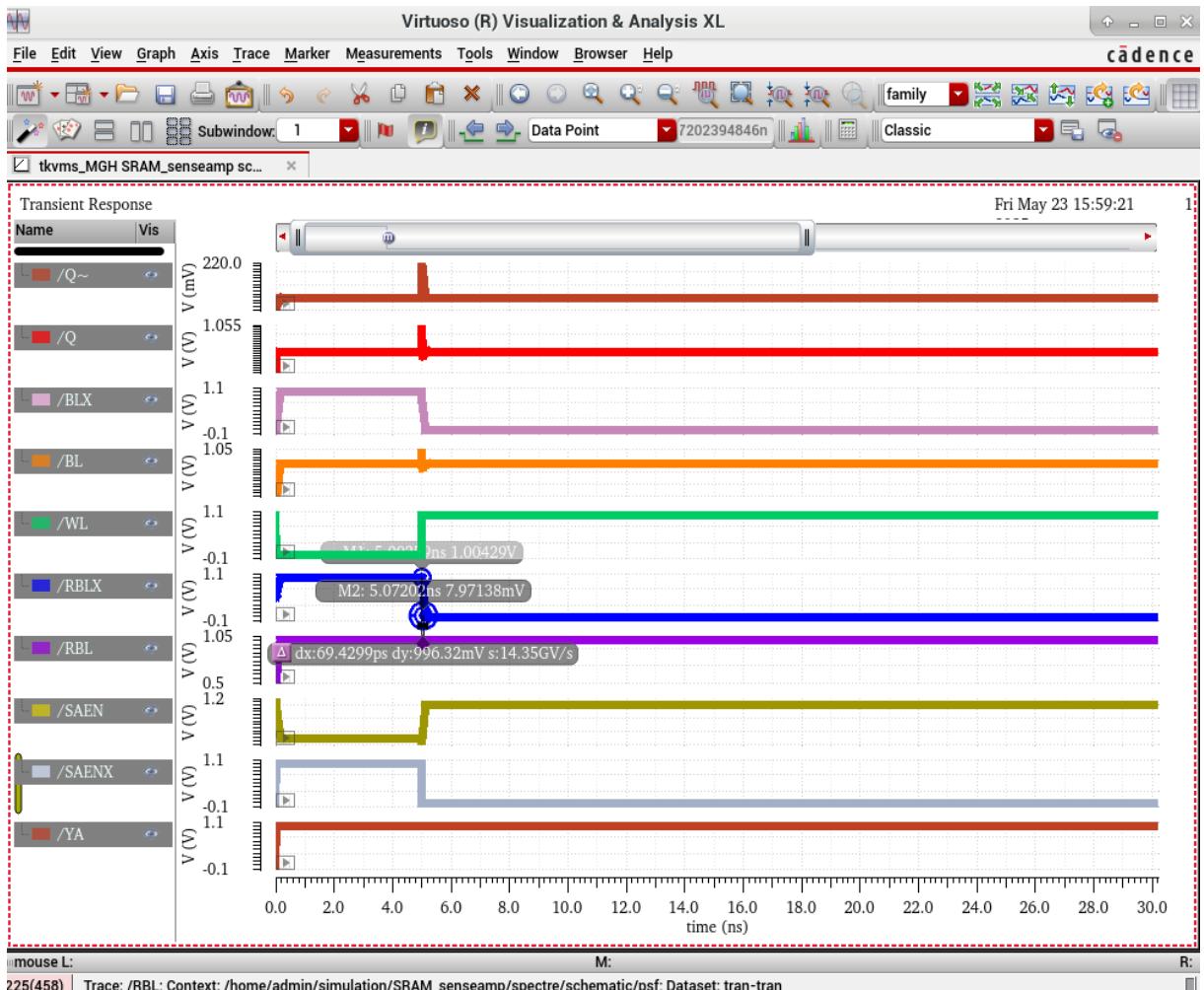


Figure 5.4.3: Transient response of SRAM using amplifier

In figure 5.4.3, we can see that the reading time has been minimized to around 69.43ps by using sense amplifier while the reading time of SRAM without using sense amplifier is 86.02ps.

## 1.5. SRAM Array 8x8

Next thing that our group is going to survey is an 8x8 SRAM array:

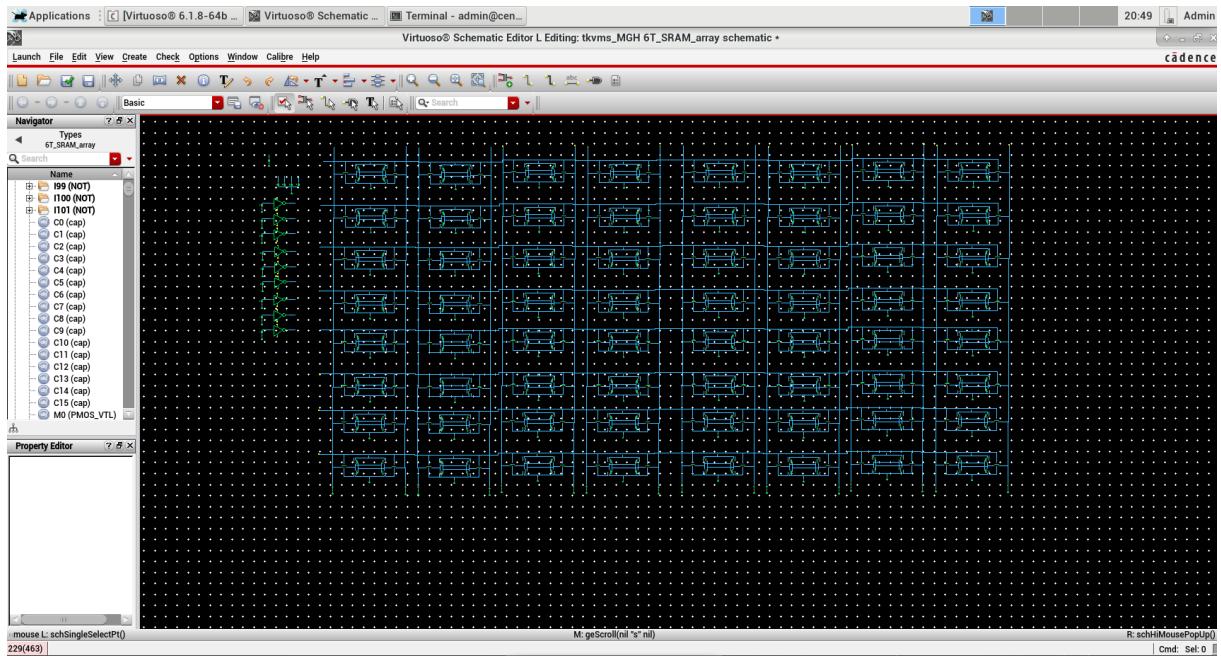


Figure 1.5.1: SRAM array 8x8

### 1.5.1 Writing mode

In writing operation, first we need to set all the BLs like below:

BL1: Voltage 1 = 0, Voltage 2 = 1V, Period = 2ns, Fall time = Rise time = 1ps, pulse width = 1ns

BL2: Voltage 1 = 0, Voltage 2 = 1V, Period = 3ns, Fall time = Rise time = 1ps, pulse width = 1.5ns, delay = 1 ns

BL3: Voltage 1 = 0, Voltage 2 = 1V, Period = 5ns, Fall time = Rise time = 1ps, pulse width = 1ns

BL4: Voltage 1 = 0, Voltage 2 = 1V, Period = 5ns, Fall time = Rise time = 1ps, pulse width = 2ns

BL5: Voltage 1 = 0, Voltage 2 = 1V, Period = 5ns, Fall time = Rise time = 1ps, pulse width = 3ns

BL6: Voltage 1 = 0, Voltage 2 = 1V, Period = 7ns, Fall time = Rise time = 1ps, pulse width = 4ns, delay = 2 ns

BL7: Voltage 1 = 0, Voltage 2 = 1V, Period = 4ns, Fall time = Rise time = 1ps, pulse width = 2ns

BL8: Voltage 1 = 0, Voltage 2 = 1V, Period = 4ns, Fall time = Rise time = 1ps, pulse width = 1ns

The result of setting all the BLs:

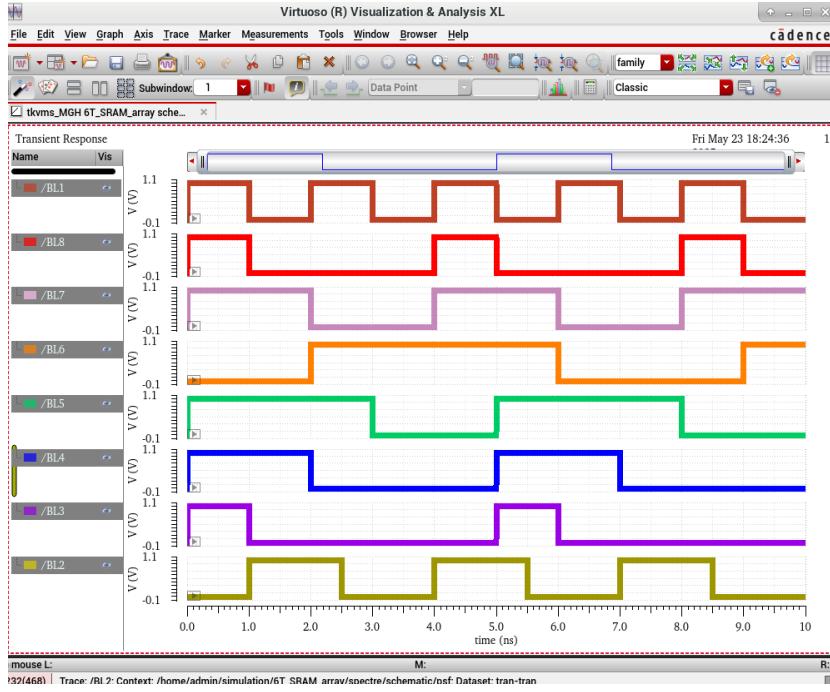


Figure 1.5.2: BLs wave

When we want to write to row 1, we need to set the WL1 high to store the data. WL1 is high from 0ns to 1ns.

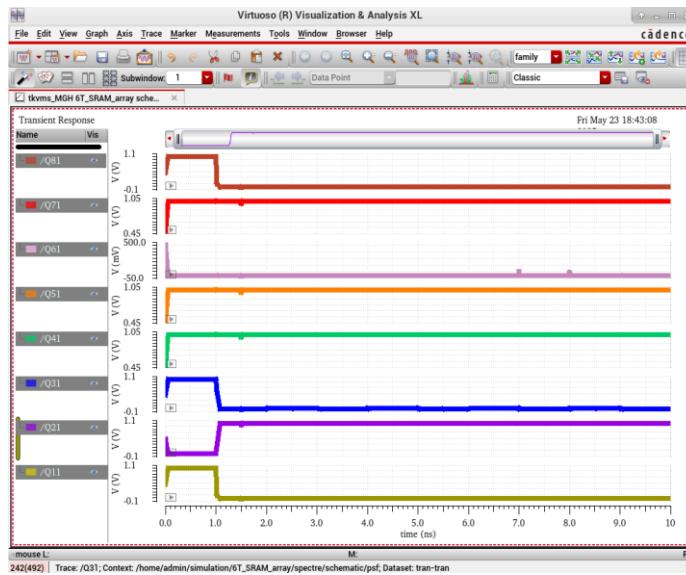


Figure 1.5.3: Result of row 1

Result Explanation: From 0ns to near 1ns, the data of BL lines is 10111011, and the result in figure 1.5.3 shows the same. In 1ns, the data of BL lines is 01011010, and the result is also correct. After 1ns, the WL1 is low so

the SRAM currently store 01011010. As a result, writing mode works correctly as expected.

### 1.5.2 Reading mode

Turning into reading operation, we will have to set some value Q in row 1 and row 6 in figures below:

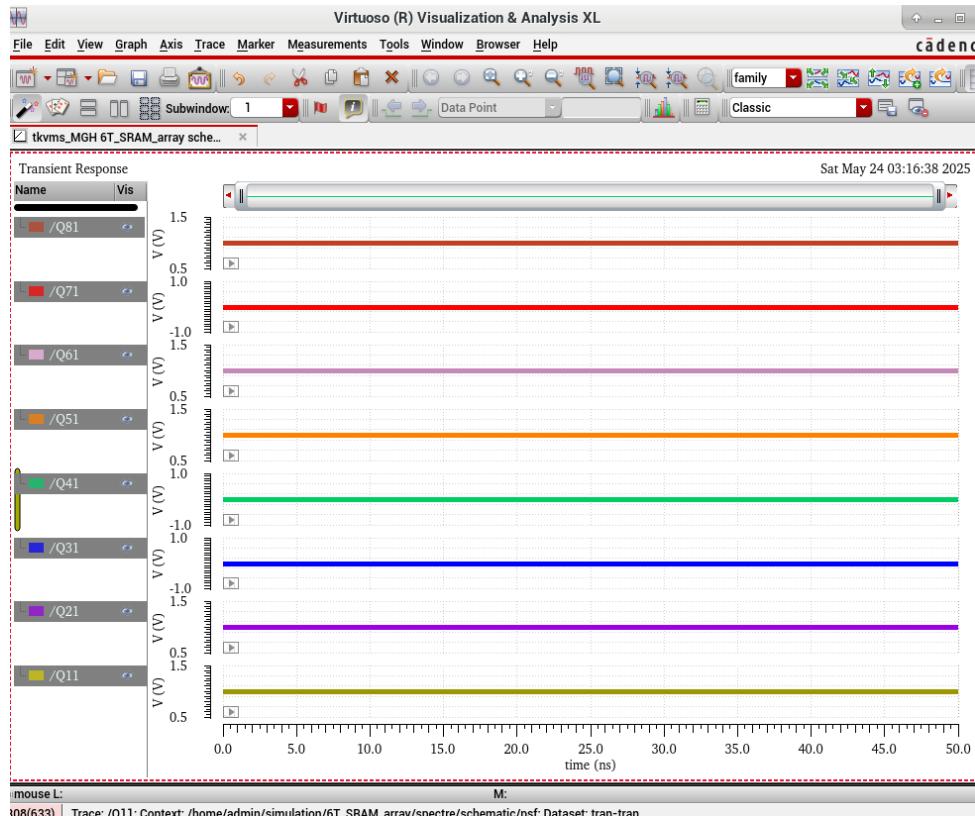


Figure 1.5.4: Setting data for Q in row 1

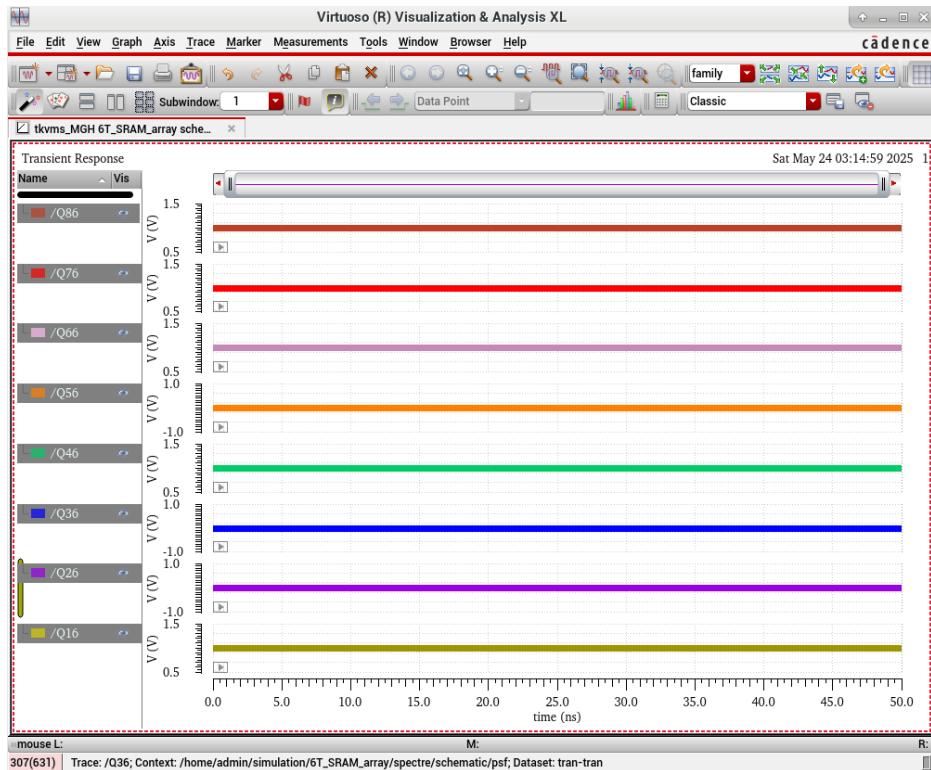


Figure 1.5.5: Setting data for Q in row 6

The data that is set in row 1 is 11001101 and in row 6 is 10010111. Therefore, when the WL1 is high, the expected results in BL lines should be 11001101 and when WL6 is high, the expected result should be 10010111.

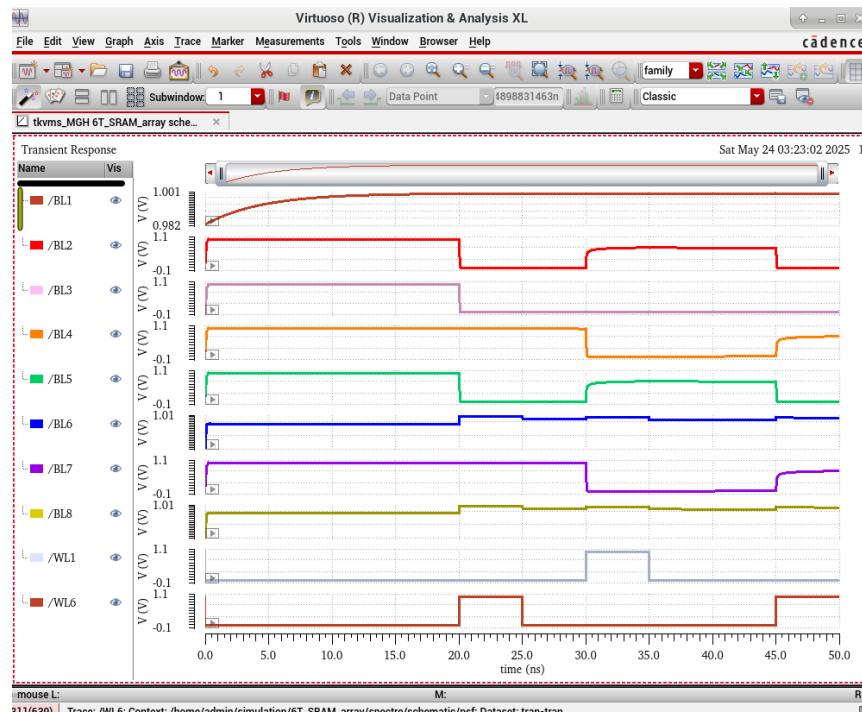


Figure 1.5.6: The result of SRAM 8x8 reading mode

Result: As we can see in figure 5.5.6, WL6 is high from 20ns to 25ns and the data in BL lines is 10010111, so the result is correct as expected. Regarding the result when WL1 is high from 30ns to 35ns, the data in BL lines is 11001101, so the result is correct as well. Therefore, we could say the SRAM array 8x8 works correctly.

## **EXPERIMENT 2: TCAM**

**Objective:** Known a specific structure of CAM (Content Addressable Memory).

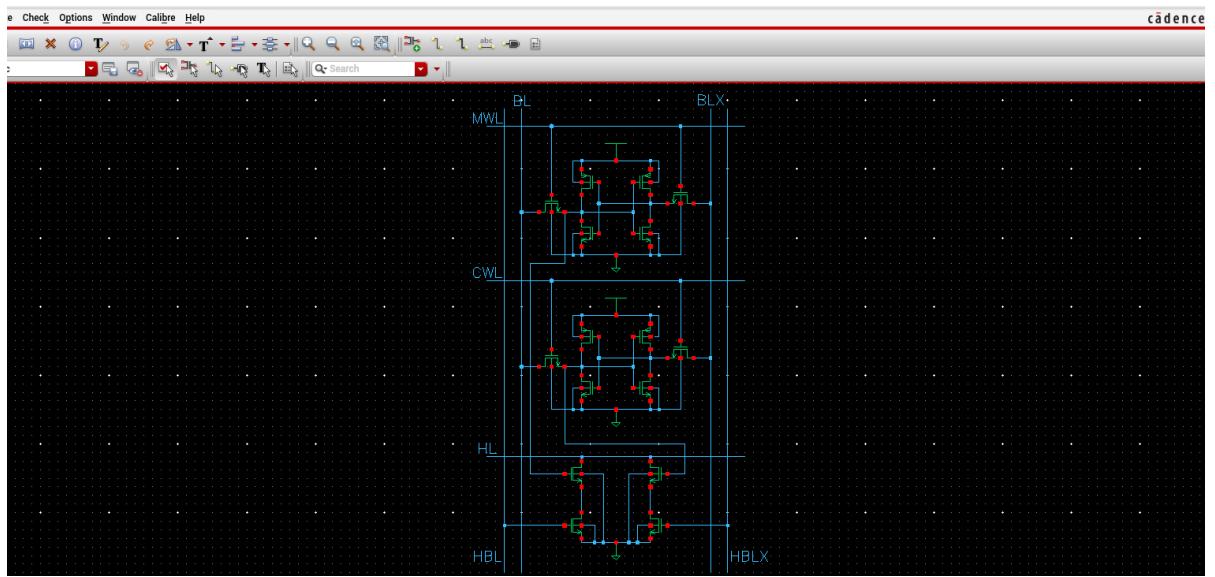
**Requirements:**

- Simulate Writing and Compare operations.
- Simulate an array of TCAM sized  $8 \times 8$ .

### **2.1 TCAM cell:**

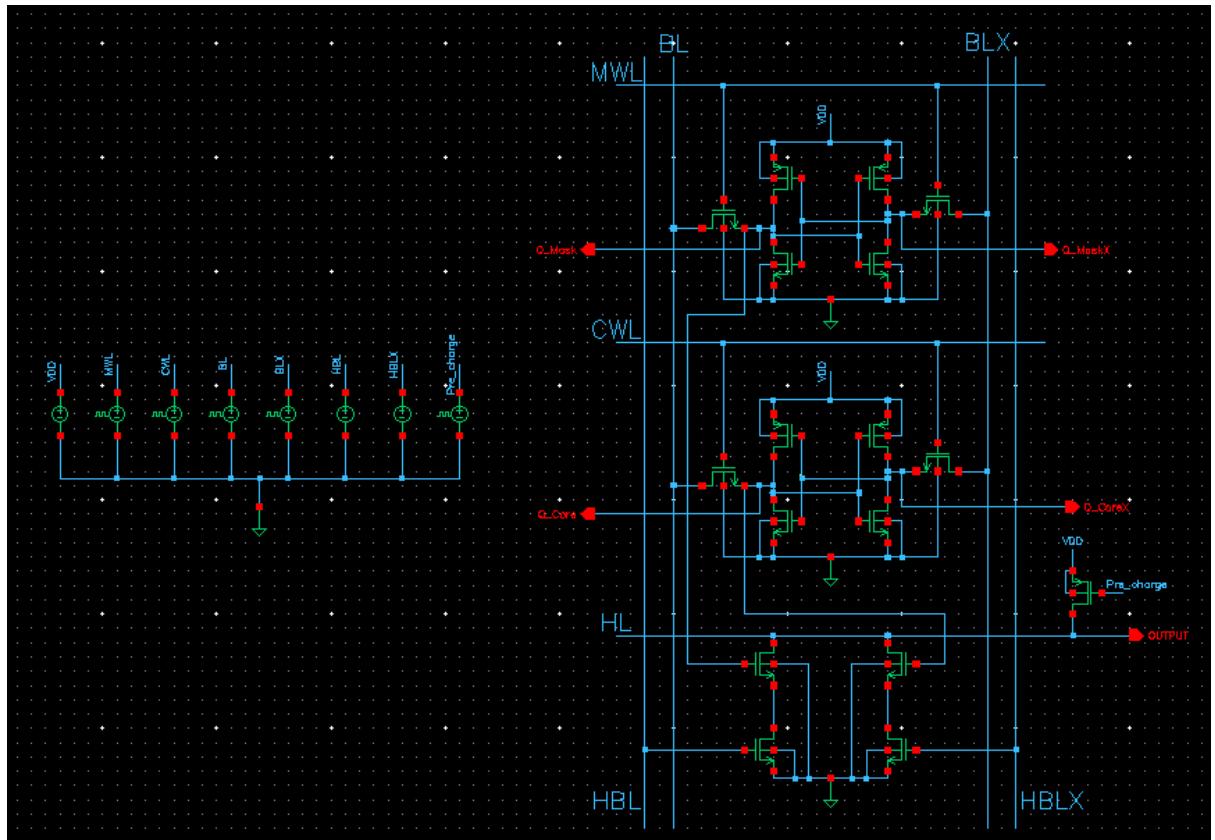
TCAM (ternary content-addressable memory) is a specialized type of high-speed memory that searches its entire contents in a single clock cycle. The term “ternary” refers to the memory's ability to store and query data using three different inputs: 0, 1 and X. TCAM can perform one search each clock cycle. This is much faster than searching traditional RAM, which requires many clock cycles to query and check each memory location.

The structure of a TCAM cell can consists of 3 parts: Mask cell, Core cell and Compare circuit. Mask cell and Core cell have the same structure as SRAM cell: 2 NOT gates are connected one after another to form a bi-stable circuit, used to store data in one cell, and 2 pass-gates. transistors, used to connect BL and BLX pins to cells. Compare is made from 4 NMOS, the connection as shown above is controlled by pins HBL, HBLX, ICBL, IMBL connected from pin HL to GND.



**Figure 2.1:** TCAM cell's schematic

Provide the sources for the signals in order to perform waveform simulation and verify the operation of the TCAM cell:



**Figure 2.2:** TCAM cell's testbench

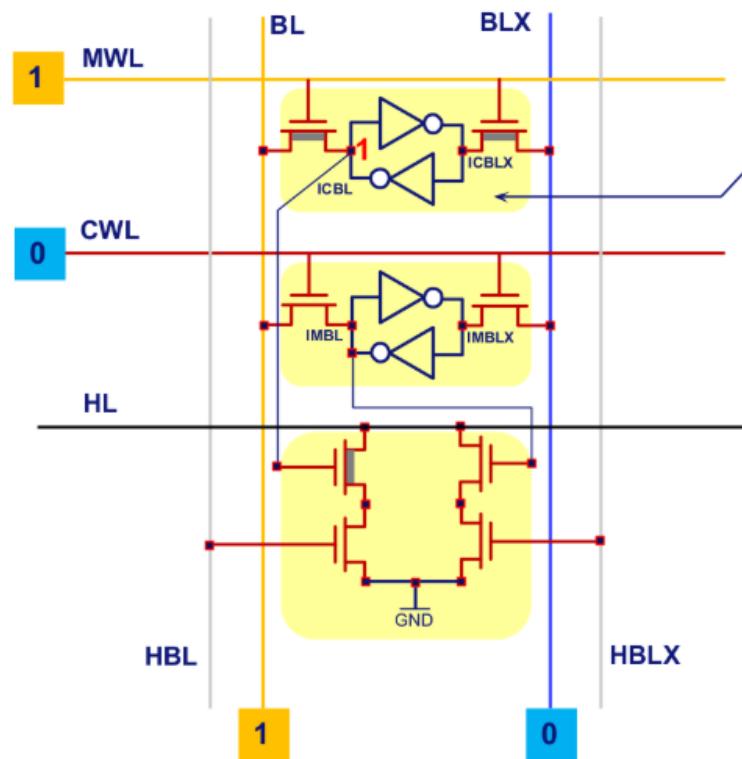
### 2.1.1: Simulate Writing operation of TCAM cell:

In writing mode, TCAM cell data will be written to the mask cell and core cell. The convention is that the value of TCAM will be written to the core cell, and the reverse reference value will be written to the mask cell. Because the mask cell and core cell are structured like a SRAM cell, the operation of the writing function is similar.

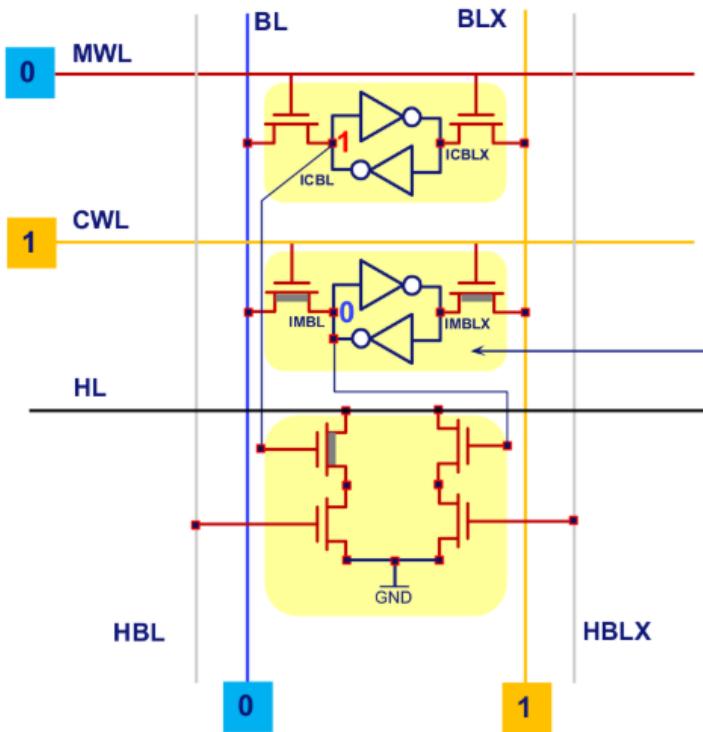
Perform a write operation of 1 into the mask cell and 0 into the core cell of the TCAM using the control signals MWL and CWL, along with the input data signals BL and BLX.

Specifically, when  $MWL = 1$ , the data on BL and BLX will be written into the mask cell, and the cell will be in read mode when  $MWL = 0$ . Similarly, for the core cell, the write and read operations are controlled by the CWL signal.

The following figures illustrate how the TCAM writes data into the mask cell and the core cell (In addition, it is possible to write other values into the TCAM cell. However, this report specifically demonstrates the process of writing a 1 into the mask cell and a 0 into the core cell. Writing other values can be carried out in a similar manner).

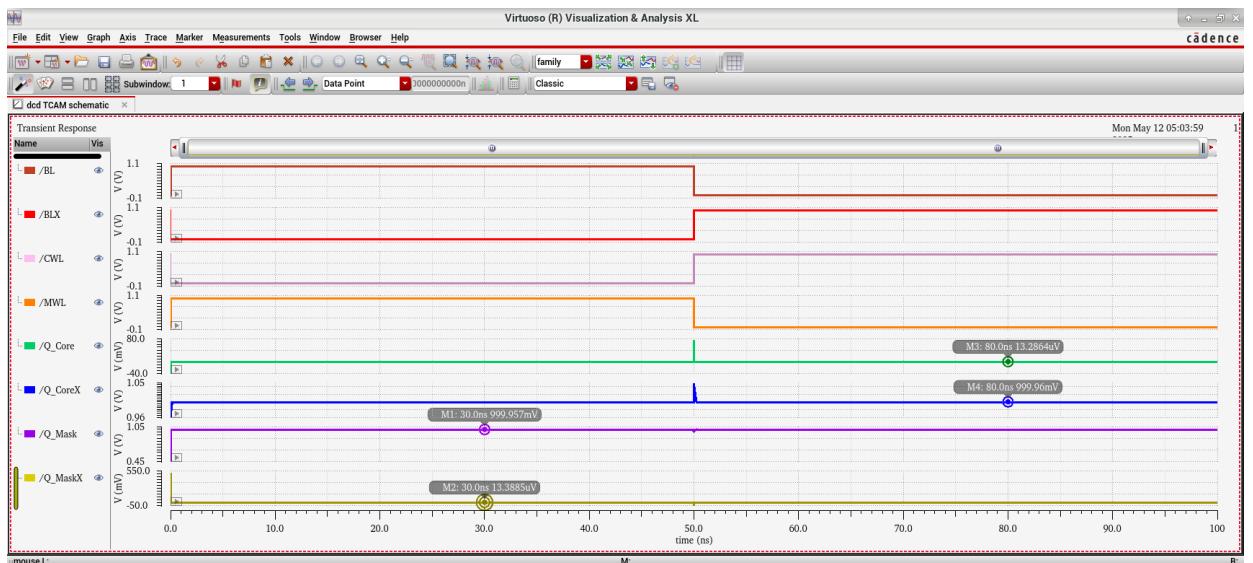


**Figure 2.3.1:** Write 1 to mask cell



**Figure 2.3.2:** Write 0 to core cell

The waveform below is generated from the TCAM testbench and illustrates the signals and data within the TCAM:



**Figure 2.3.3:** Waveform of TCAM cell

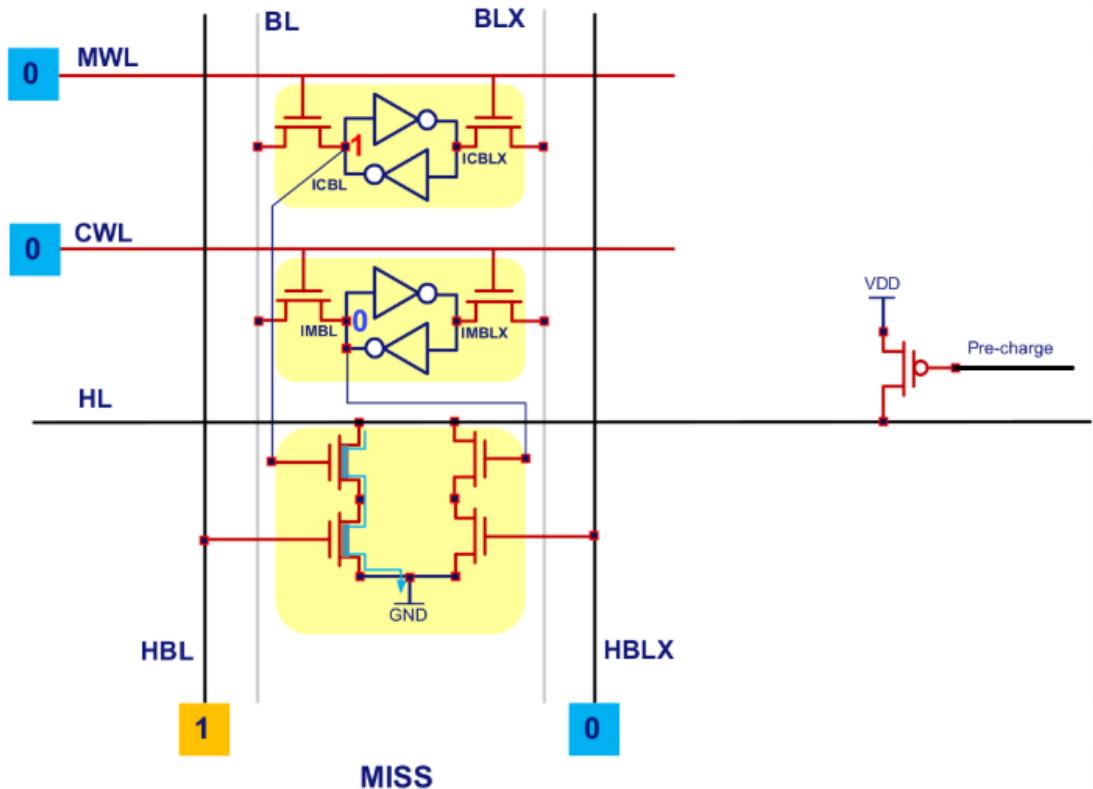
=> By examining the signals Q\_Core and Q\_Mask, it can be observed that the value 1 has been successfully written into the mask cell, and 0 has been written into the core cell.

### **2.1.2: Simulation Compare operation of a TCAM cell:**

In comparison mode, when data has been written to the core cell and mask cell. First, reset the signal at the Pre-charge pin to “0” to charge the HL pin (set the signal level to 1). After a certain time, set the signal at the Pre-charge pin to “1” to stop the charging process. Then, transmit the signal to be compared to pin HBL and the signal with inverted logic level to HBLX. If the value to be compared is the same as the value in the core cell, the comparator circuit does not create a path to discharge to GND, so logic level at pin HL is maintained as “1”, also known as MATCH state. If the value to be compared is different from the value in the core cell, the comparator circuit creates a path to discharge to GND, so the logic level at the HL pin changes to "0", also known as the MISS state. In some special cases, it is possible to set the value for the core cell and mask cell to the same "1" or "0". If the core cell and mask cell both store the value "0", that TCAM memory cell always has the ALWAYS MATCH state. On the contrary, if the core cell and mask cell both store the value "1", that TCAM memory cell will always have the ALWAYS MISS state.

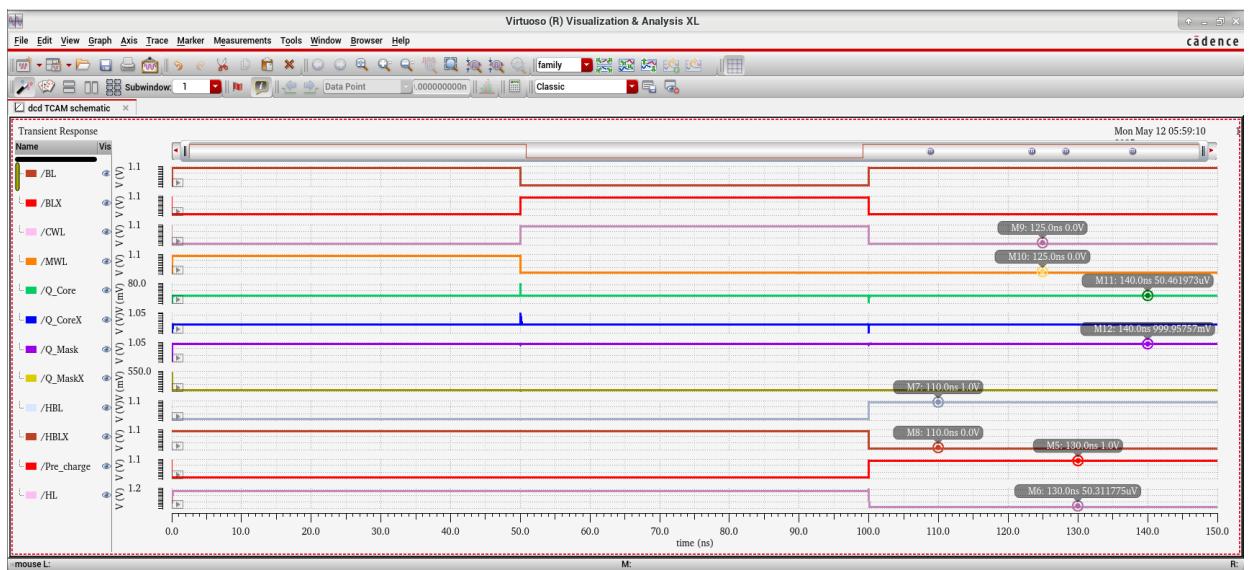
The following figures illustrate how the states are generated during the compare operation, along with waveform-based verification from the TCAM testbench:

- **MISS state of TCAM cell:**



**Figure 2.4.1:** MISS state of TCAM cell

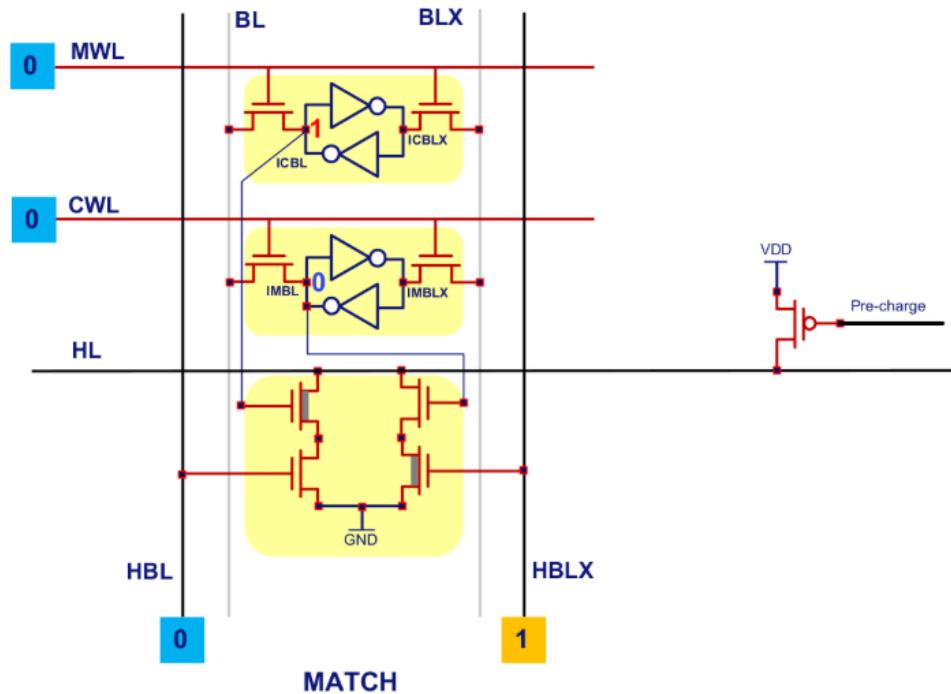
After writing the desired values into the cells, the pre\_charge signal is pulled low (set to 0), and the comparison is performed using the values on HBL and HBLX:



**Figure 2.4.2:** Waveform of TCAM cell's MISS state

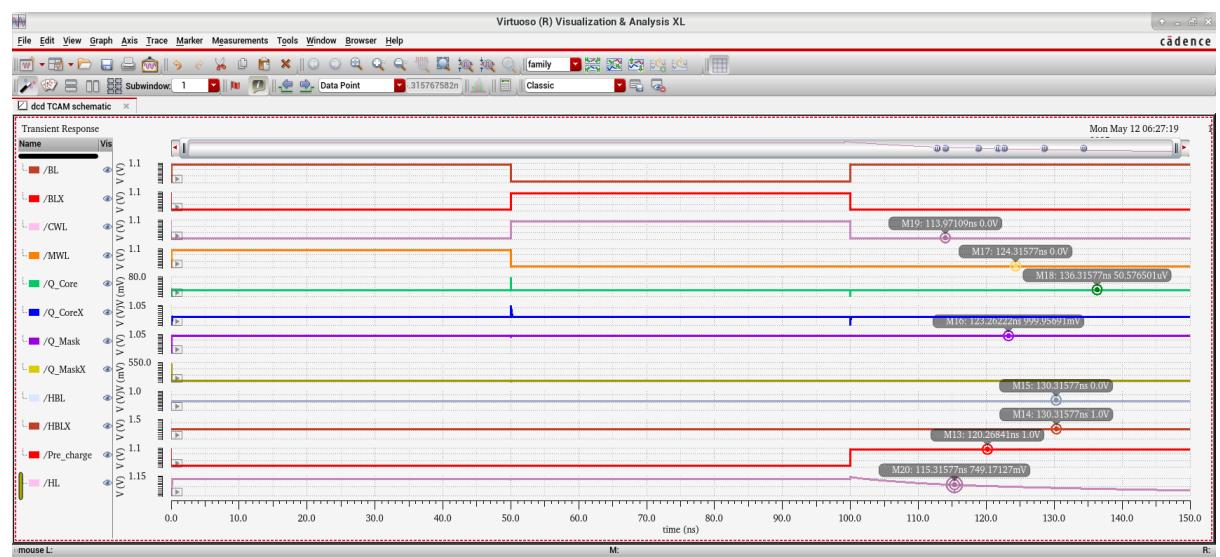
=> The waveform demonstrates that the TCAM operates correctly in the MISS state, where the HL signal is 0 when the values of HBL and the core cell do not match.

- **MATCH** state of TCAM cell:



**Figure 2.4.3:** MATCH state of TCAM cell

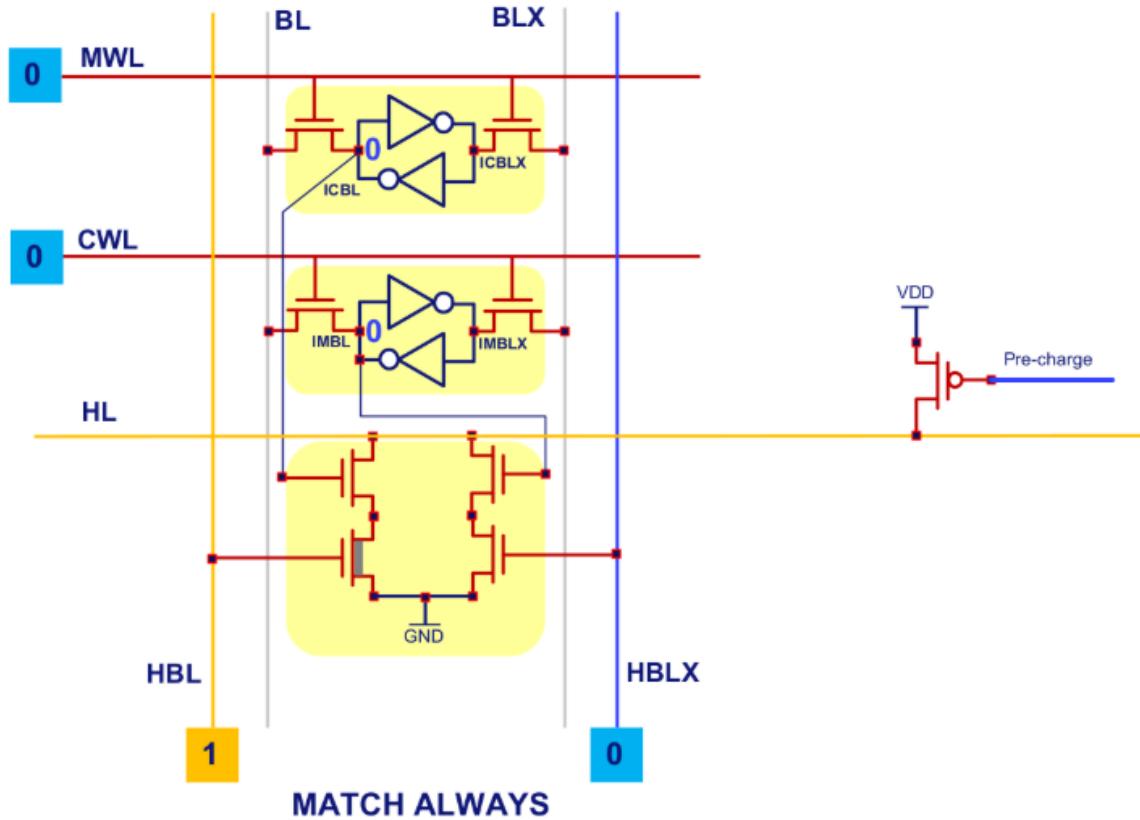
After writing the desired values into the cells, the pre\_charge signal is pulled low (set to 0), and the comparison is performed using the values on HBL and HBLX:



**Figure 2.4.4:** Waveform of TCAM cell's MATCH state

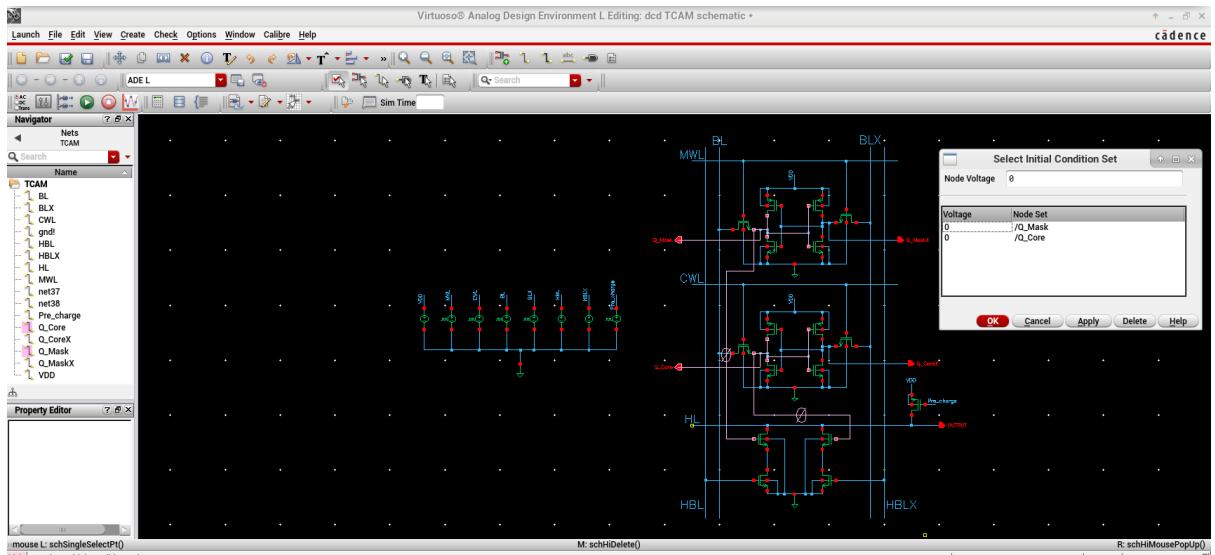
=> The waveform demonstrates that the TCAM operates correctly in the MATCH state, where the HL signal is 1 when the values of HBL and the core cell match.

- **MATCH ALWAYS** state of TCAM cell:



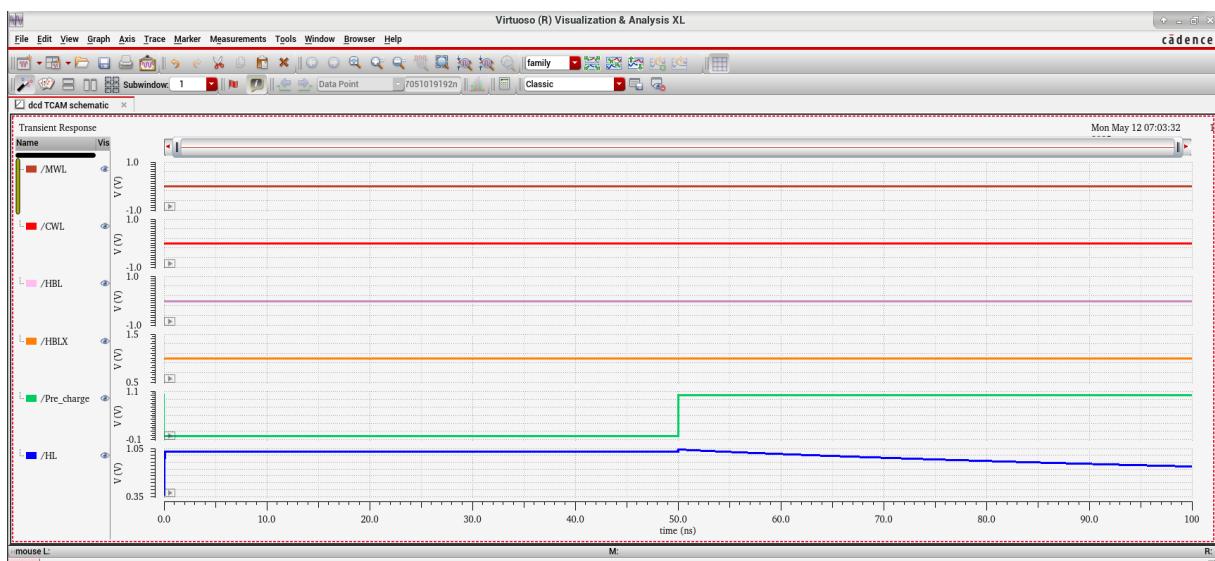
**Figure 2.4.5:** MATCH ALWAYS state of TCAM cell

Since we are only focusing on the TCAM cell operation during the compare operation, using **Initial Condition** allows us to set the stored bits in the mask cell and core cell. This approach facilitates a smoother and more efficient simulation process.



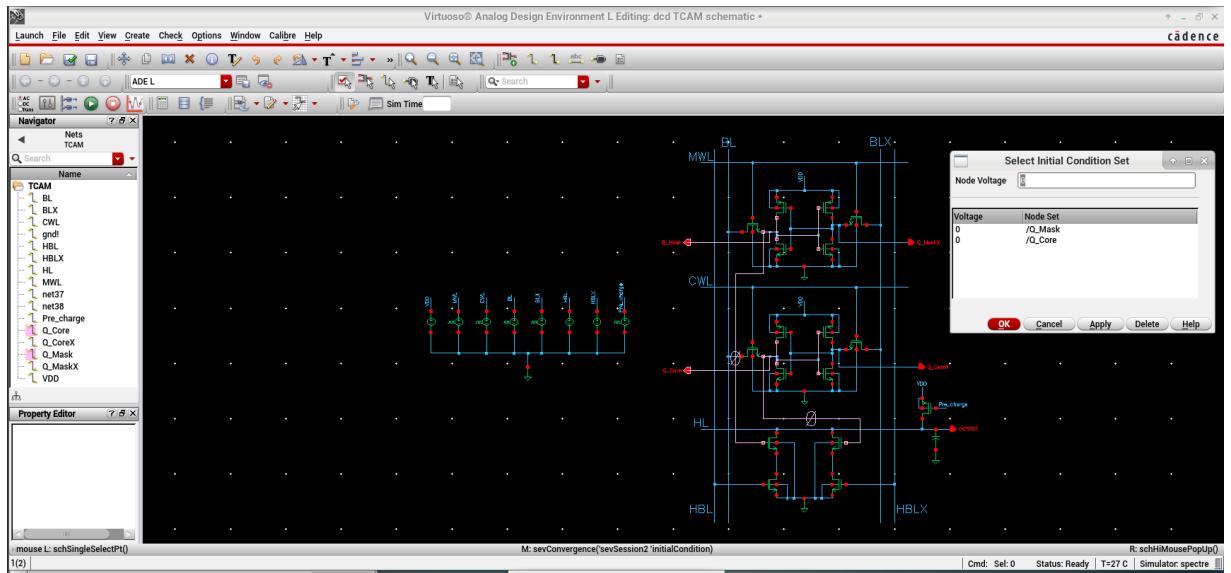
**Figure 2.4.6:** Set initial values in core cell and mask cell

Perform waveform simulation for the MATCH ALWAYS state of the TCAM cell:



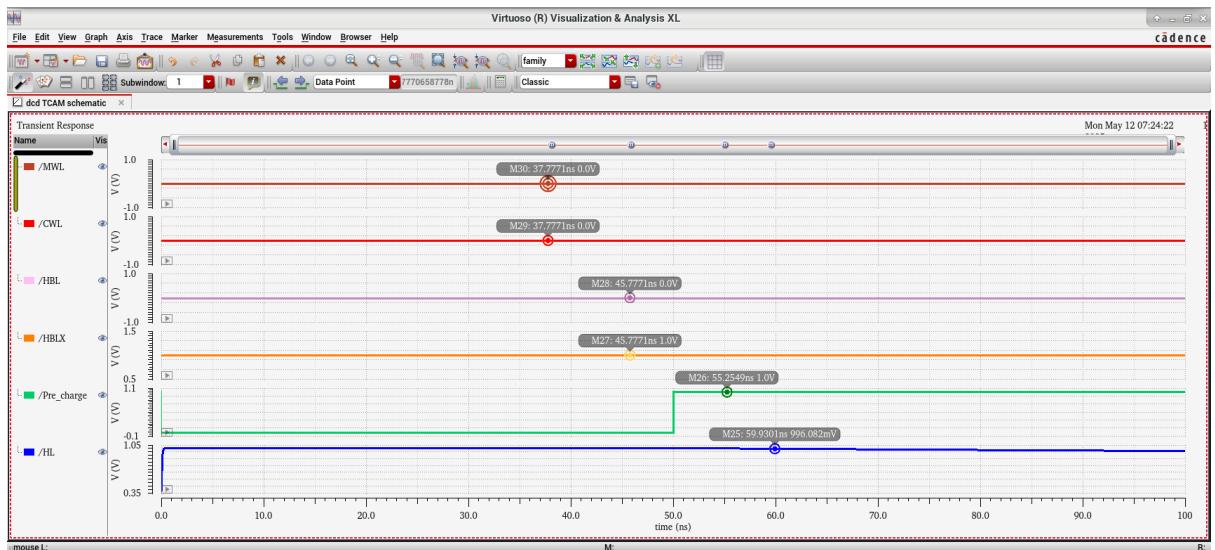
**Figure 2.4.7:** Waveform of TCAM cell's MATCH ALWAYS state

=> The output signal is clipped due to the presence of parasitic capacitances. To address this issue, I found that adding a bypass capacitor helps stabilize the HL signal and reduces clipping.



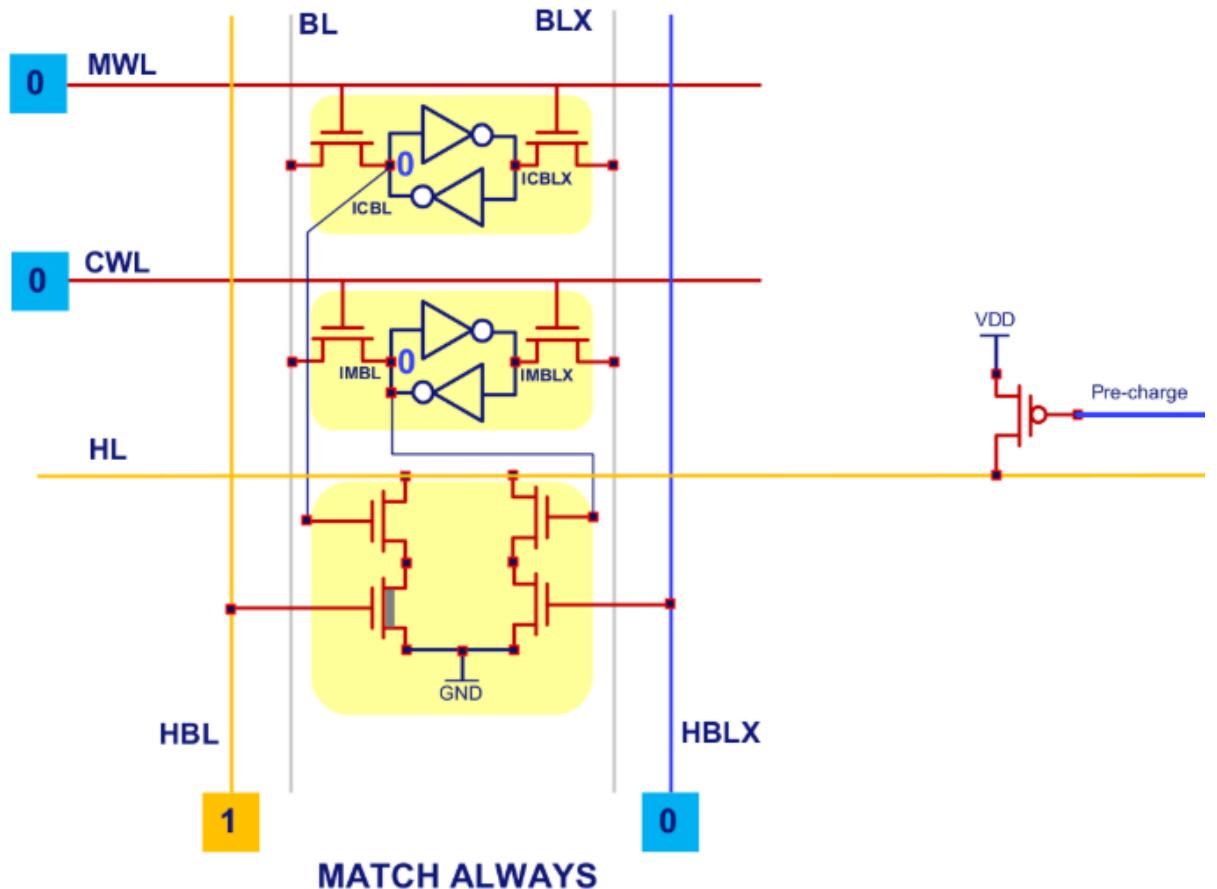
**Figure 2.4.8:** Add bypass capacitor to HL signal

A 0.01 pF bypass capacitor was used, and the waveform was regenerated to verify the improvement.

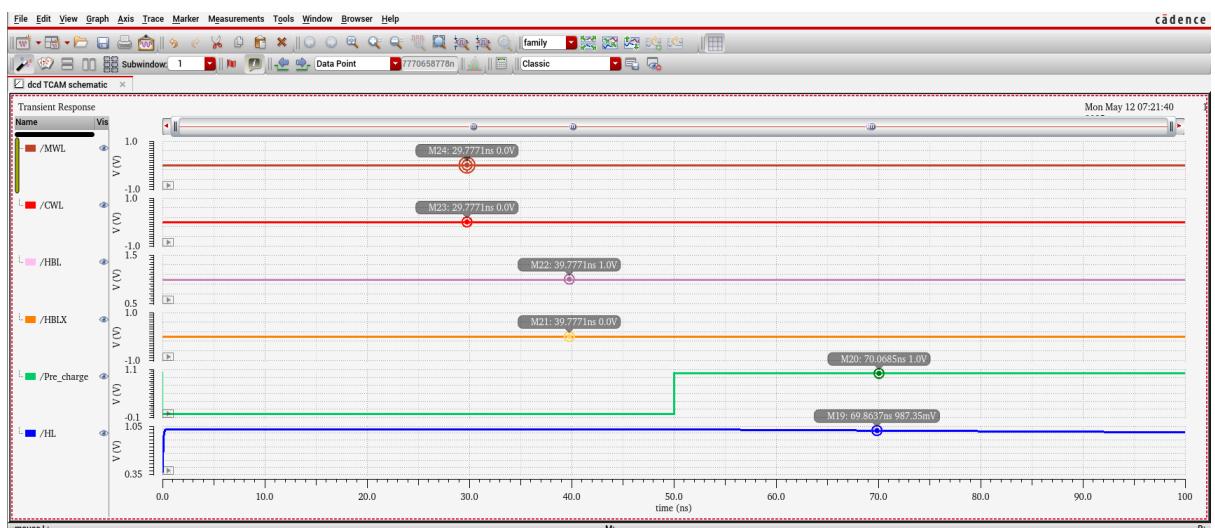


**Figure 2.4.9:** Waveform of TCAM cell after adding capacitor

With different values of HBL and HBLX, the result and operation of the TCAM cell will vary:



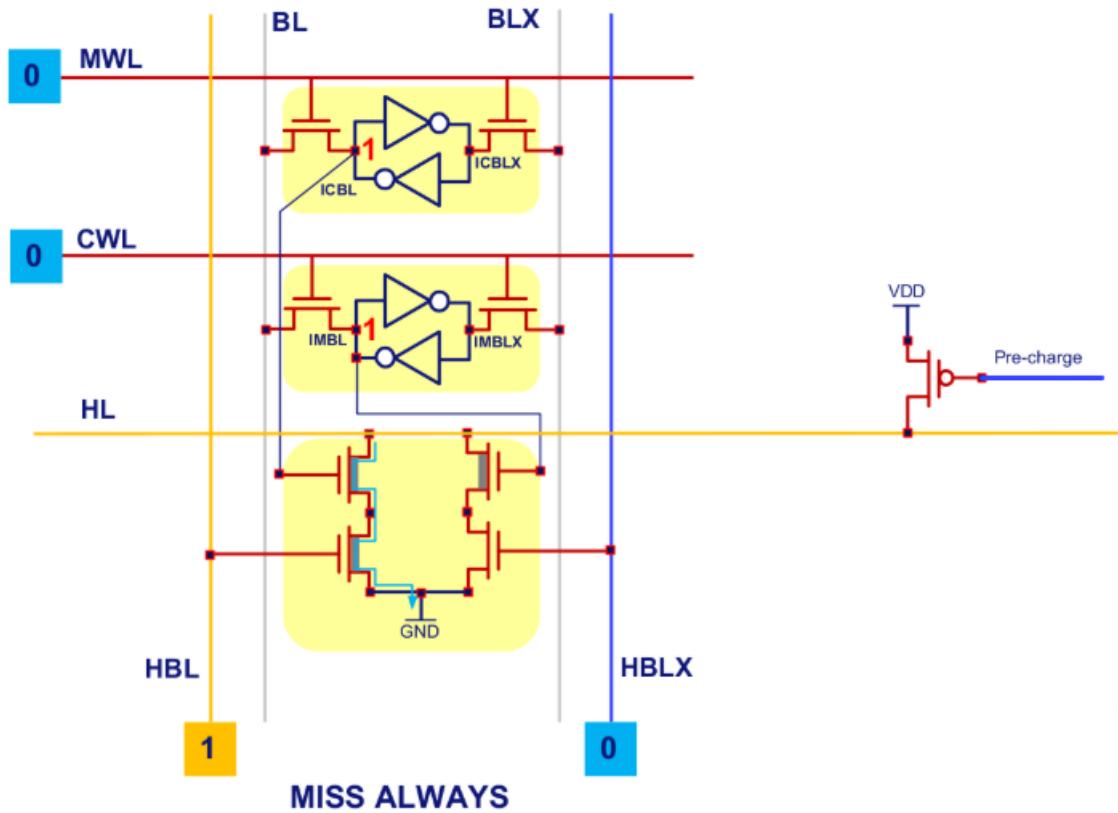
**Figure 2.4.10:** Another HBL, HBLX value of MATCH ALWAYS state



**Figure 2.4.11:** Waveform of TCAM cell's MATCH ALWAYS with another HBL, HBLX

=> With a value of 0 in both the core cell and the mask cell, the output HL will be at logic level 1, regardless of the values of HBL and HBLX.

- MISS ALWAYS state of TCAM cell:



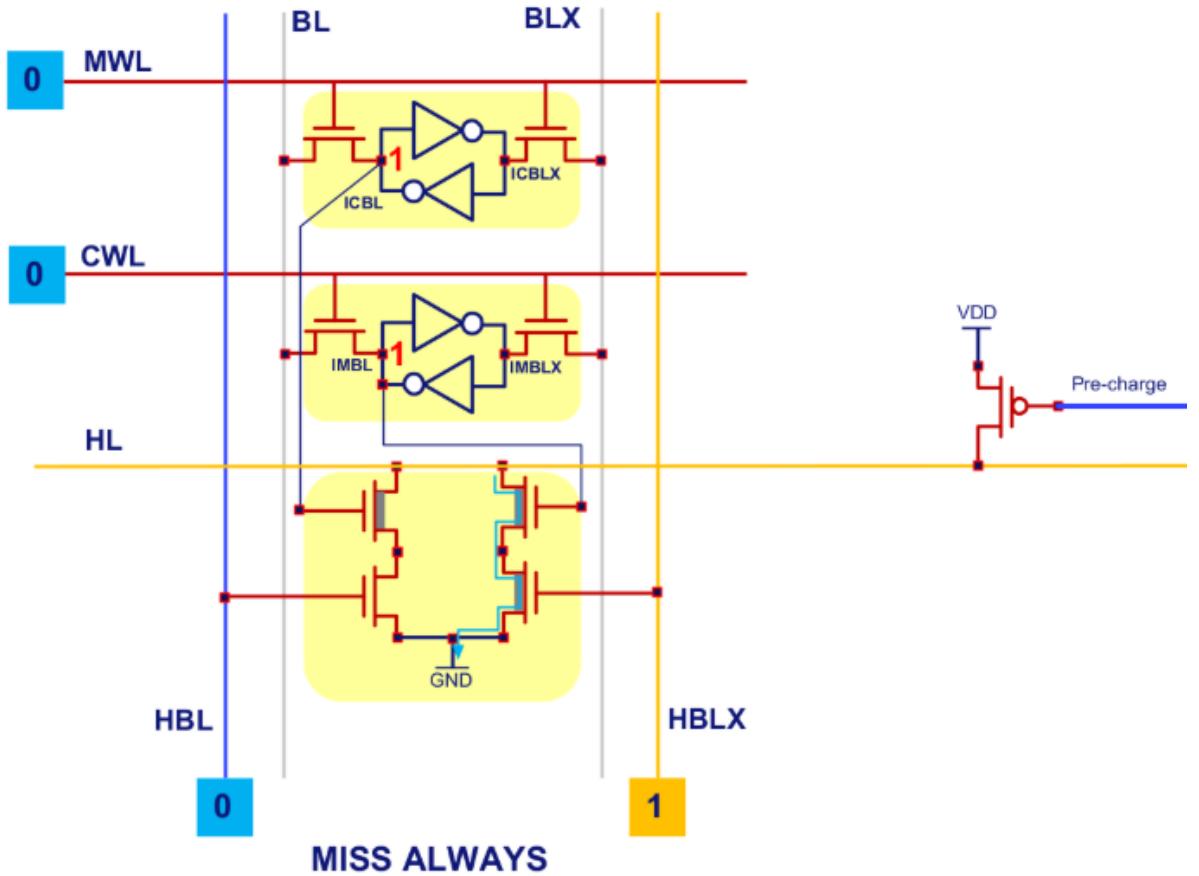
**Figure 2.4.12:** MISS ALWAYS state of TCAM cell

The waveform illustrates the signals in the MISS ALWAYS state of the TCAM cell:

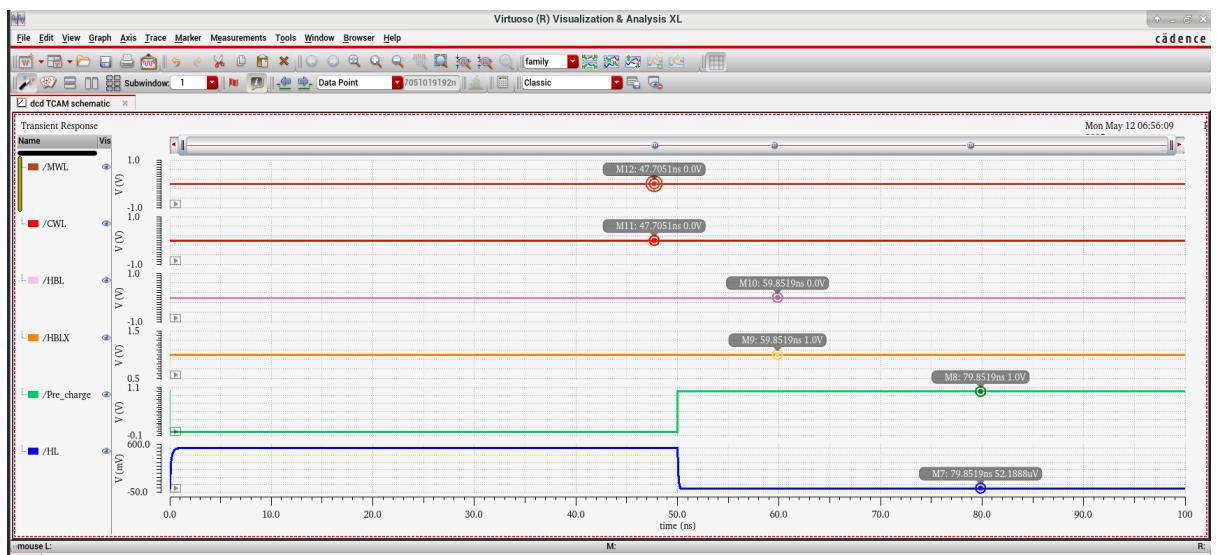


**Figure 2.4.13:** Waveform of TCAM cell's MISS ALWAYS state

Another HBL, HBLX value:



**Figure 2.4.14:** Another HBL, HBLX of TCAM cell's MISS ALWAYS state

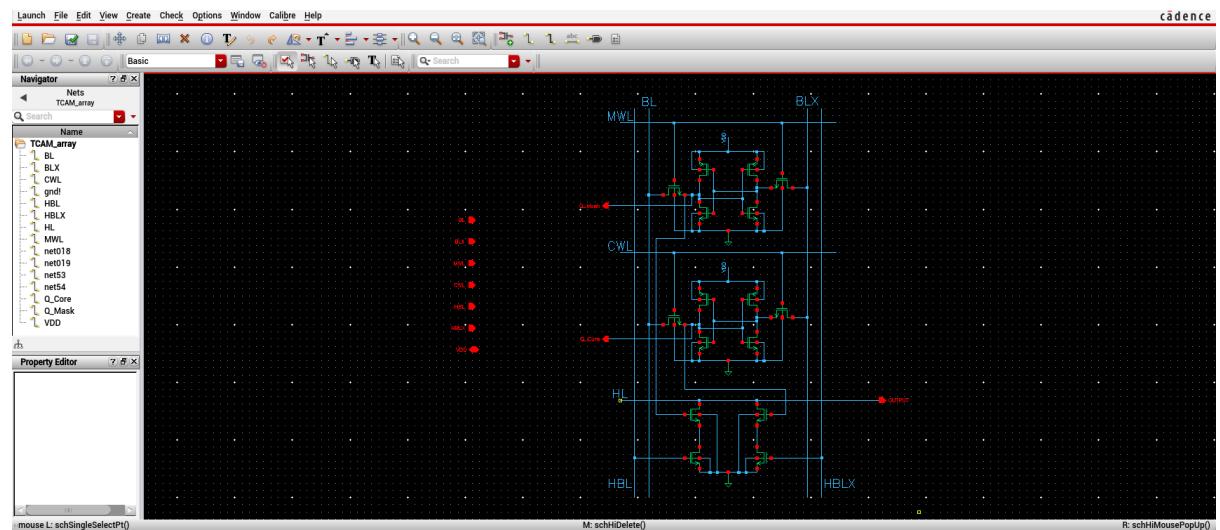


**Figure 2.4.15:** Waveform of TCAM cell's MISS ALWAYS state with another HBL, HBLX

=> With both the core cell and the mask cell of the TCAM set to logic level 1, the output HL during the compare operation will always be at logic level 0, as it is connected to GND

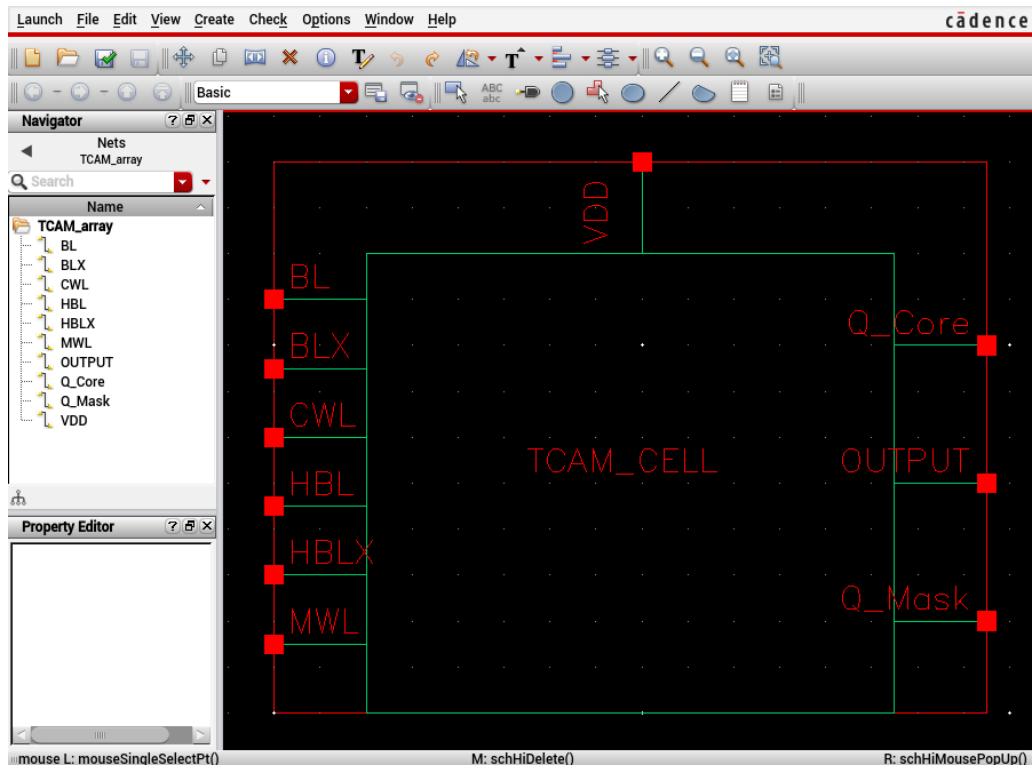
## 2.2 An array of TCAM sized 8x8:

Design a TCAM cell that can be used to build an array of TCAM



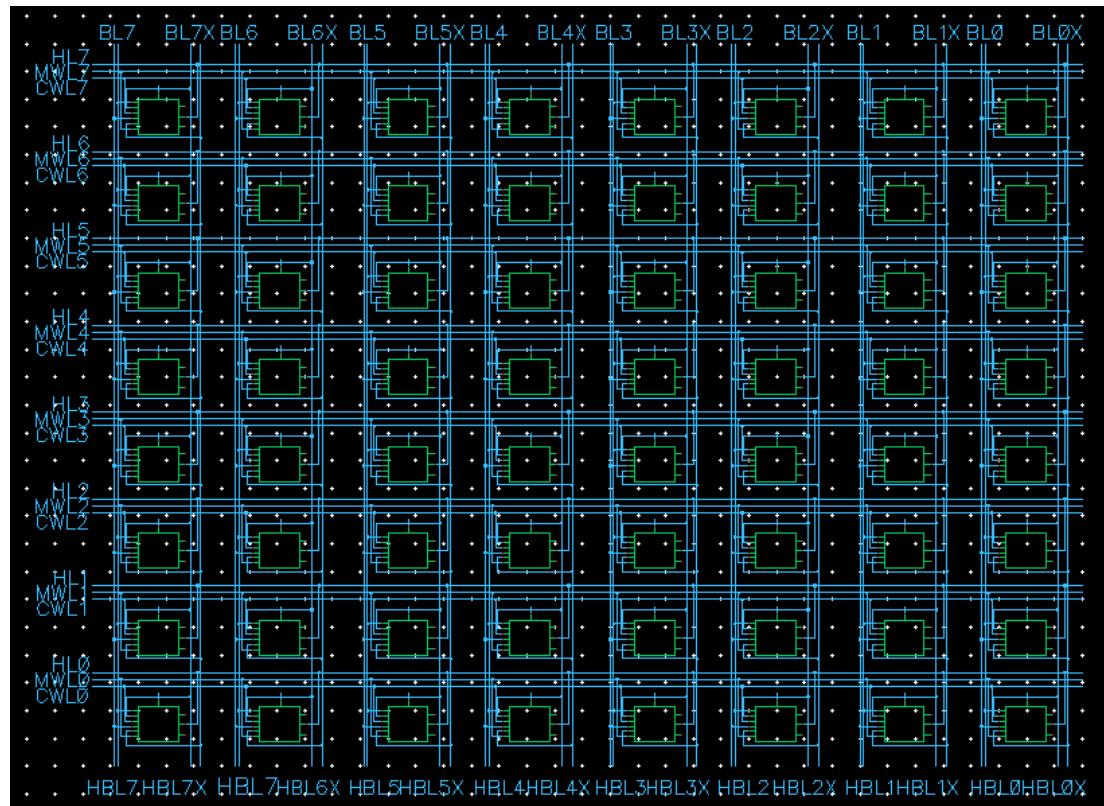
**Figure 2.5.1:** Schematic of TCAM cell

Create a symbol for the TCAM cell to facilitate its use in schematic designs and to enable the construction of larger TCAM arrays.



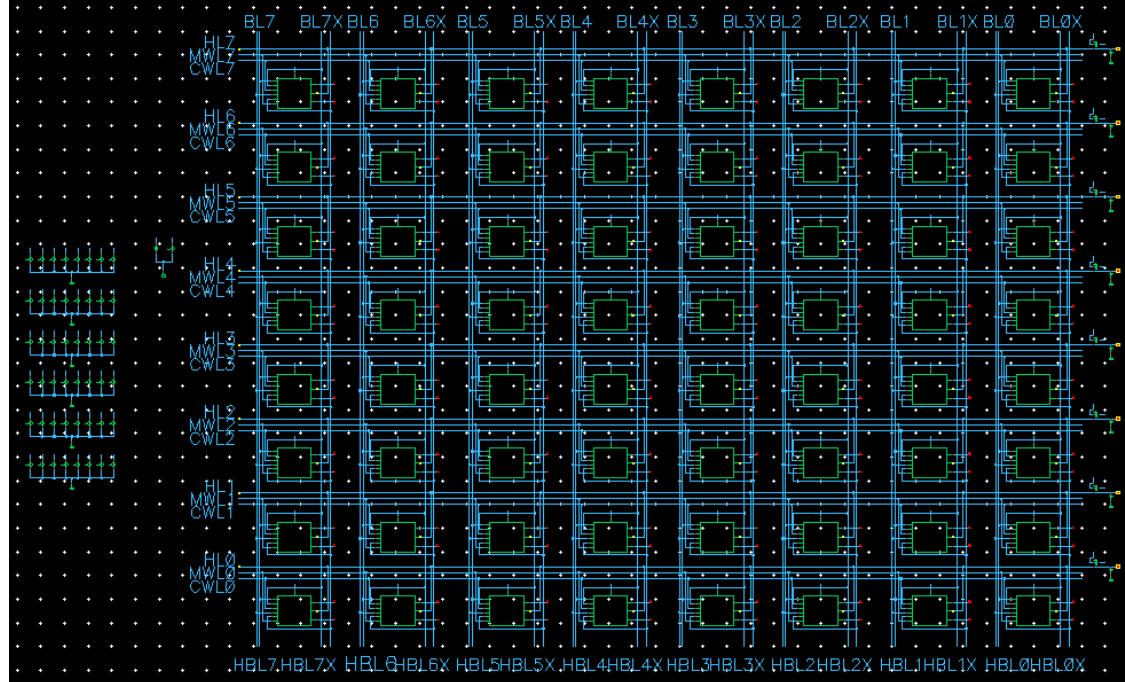
**Figure 2.5.2:** Symbol of TCAM cell

Connect the TCAM blocks together and establish the shared signals, including: HBL, HBLX, BL, BLX, MWL, CWL, and HL.



**Figure 2.5.3:** An array of TCAM sized 8x8's schematic

Provide the necessary signals to perform write operations into the TCAM cells and to carry out data search operations by row in an array of TCAM:

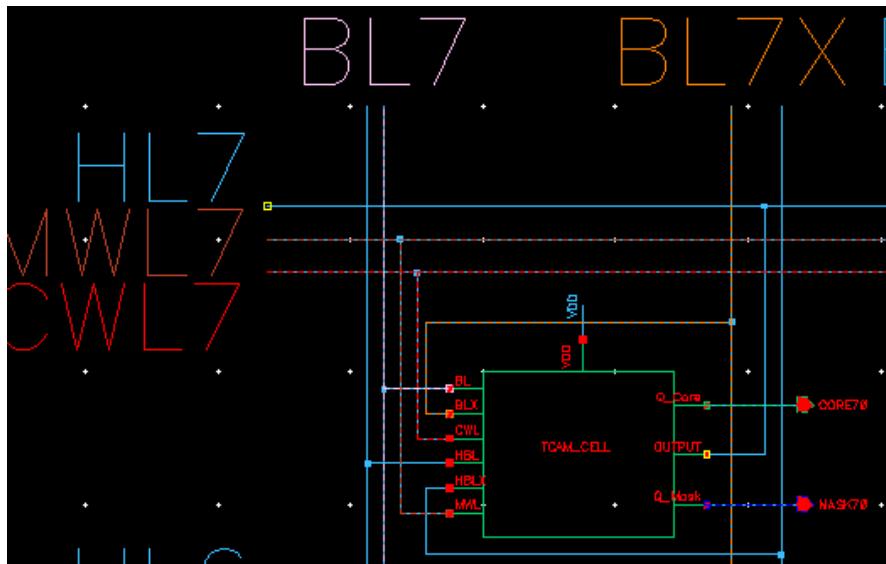


**Figure 2.5.4:** An array of TCAM testbench

### 2.2.1 Simulate Writing operation of an array of TCAM sized 8x8:

Specifically, to write a value into one TCAM cell among the 64 TCAM cells, the process is as follows:

Activate the MWL and CWL signals of the target row one at a time (in this example, row 7: MWL7, CWL7), and provide the desired value to the TCAM cell through the corresponding column bitlines (in this case, column 0: BL0, BL0x).



**Figure 2.6.1:** Schematic of a TCAM cell

Set the values of the signals in the same manner as when writing to a standalone TCAM cell.



**Figure 2.6.2:** Waveform of writing operation to 1 TCAM cell

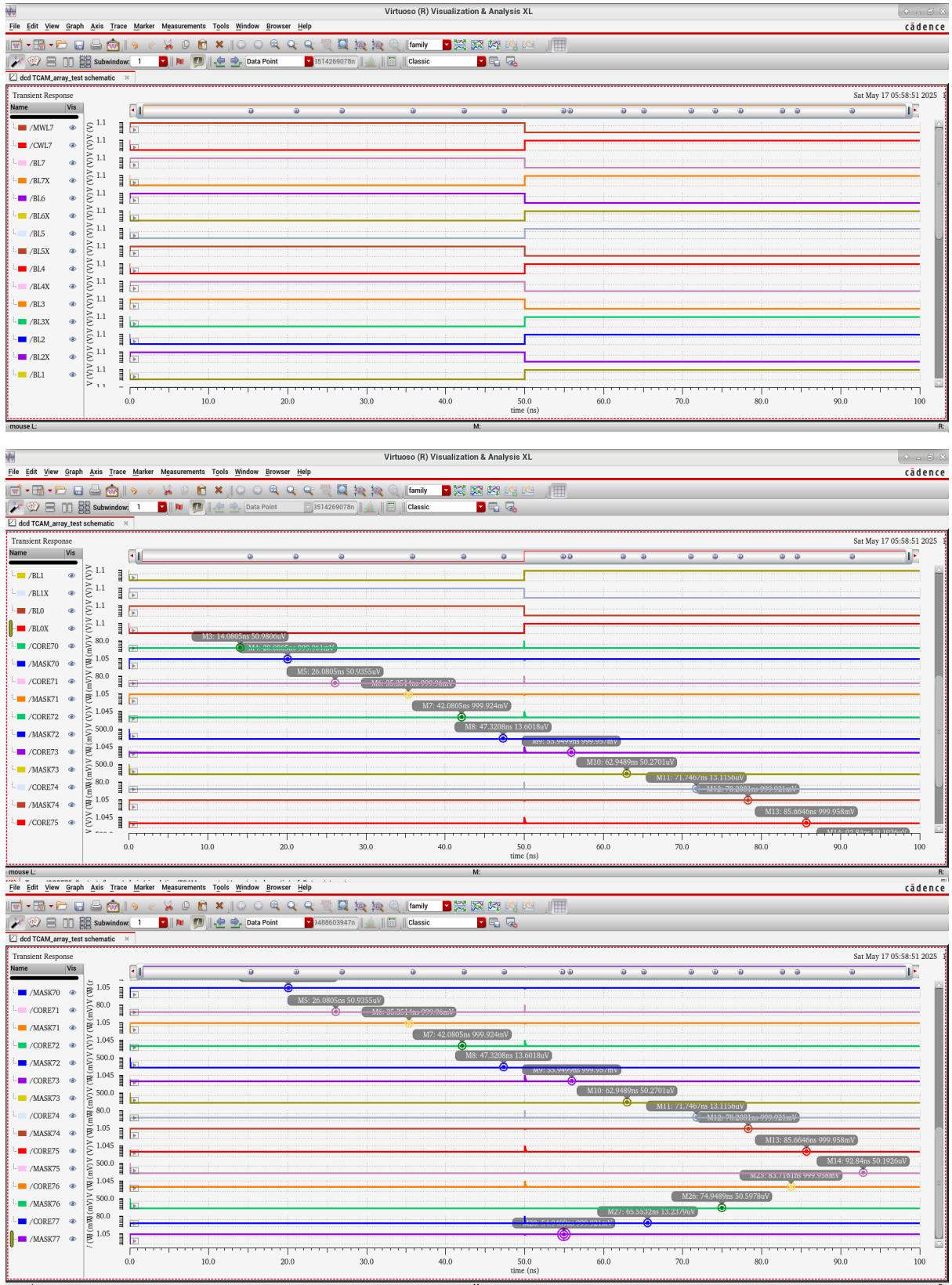
=> By reading the Core and Mask signals, it is confirmed that the write operation into the TCAM was successful.

Each write operation to the TCAM array writes an entire row consisting of 8 cells.

For example, we will perform a write operation with the bit sequence 0, 0, 1, 1, 0, 1, 1, 0 into the TCAM cells from left to right.

This includes writing into both the mask cell and the core cell of each TCAM cell in a single row of the TCAM array. The write process is carried out

in parallel, with the values for each TCAM cell in the row provided by the corresponding signals: BL7, BL7X; BL6, BL6X; ... and so on.



**Figure 2.6.3+2.6.4+2.6.5:** Waveform of writing operation a row of an TCAM array

=> The values have been successfully written into the TCAM cells according to the desired bit sequence.

Repeat the same procedure for the remaining rows of the TCAM array. After completing this process, we will have an  $8 \times 8$  TCAM array containing the desired data, which can then be used for search operations.

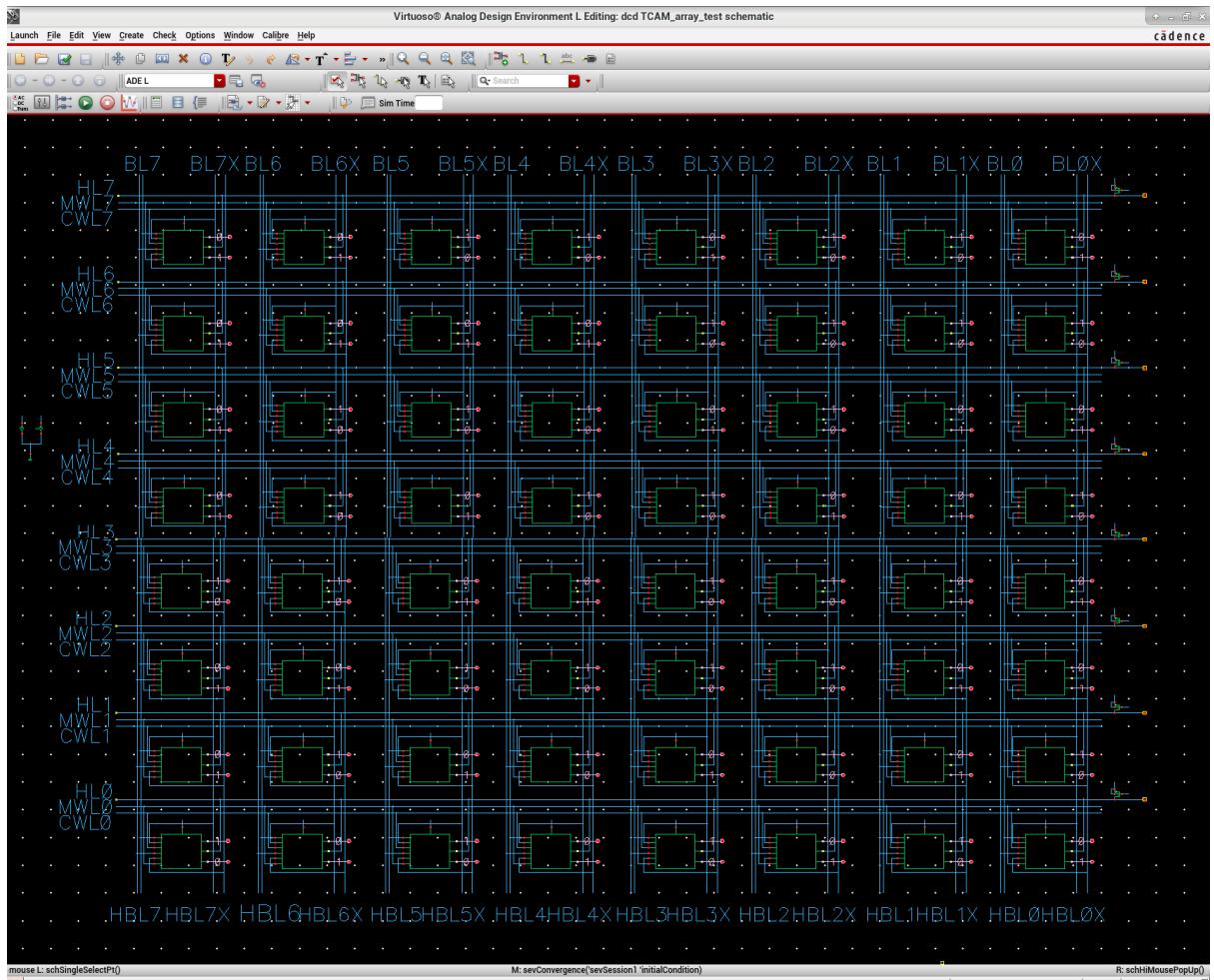
### 2.2.2 Simulate Compare operation of an array of TCAM sized 8x8:

To facilitate the reliability check of the compare operation in an array of TCAMs, we will configure the values of the 64 TCAM cells as follows:

0	0	1	1	0	1	1	0
0	0	0	0	1	1	1	1
0	1	1	1	0	1	0	0
0	1	0	1	1	0	0	1
1	1	0	0	1	1	0	0
0	0	1	1	1	1	0	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0

**Table 1:** Initial values of TCAM cells

Using Initial Conditions simplifies the verification of the compare operation. We will assign initial values to each mask cell and core cell of every TCAM cell according to the predefined table.



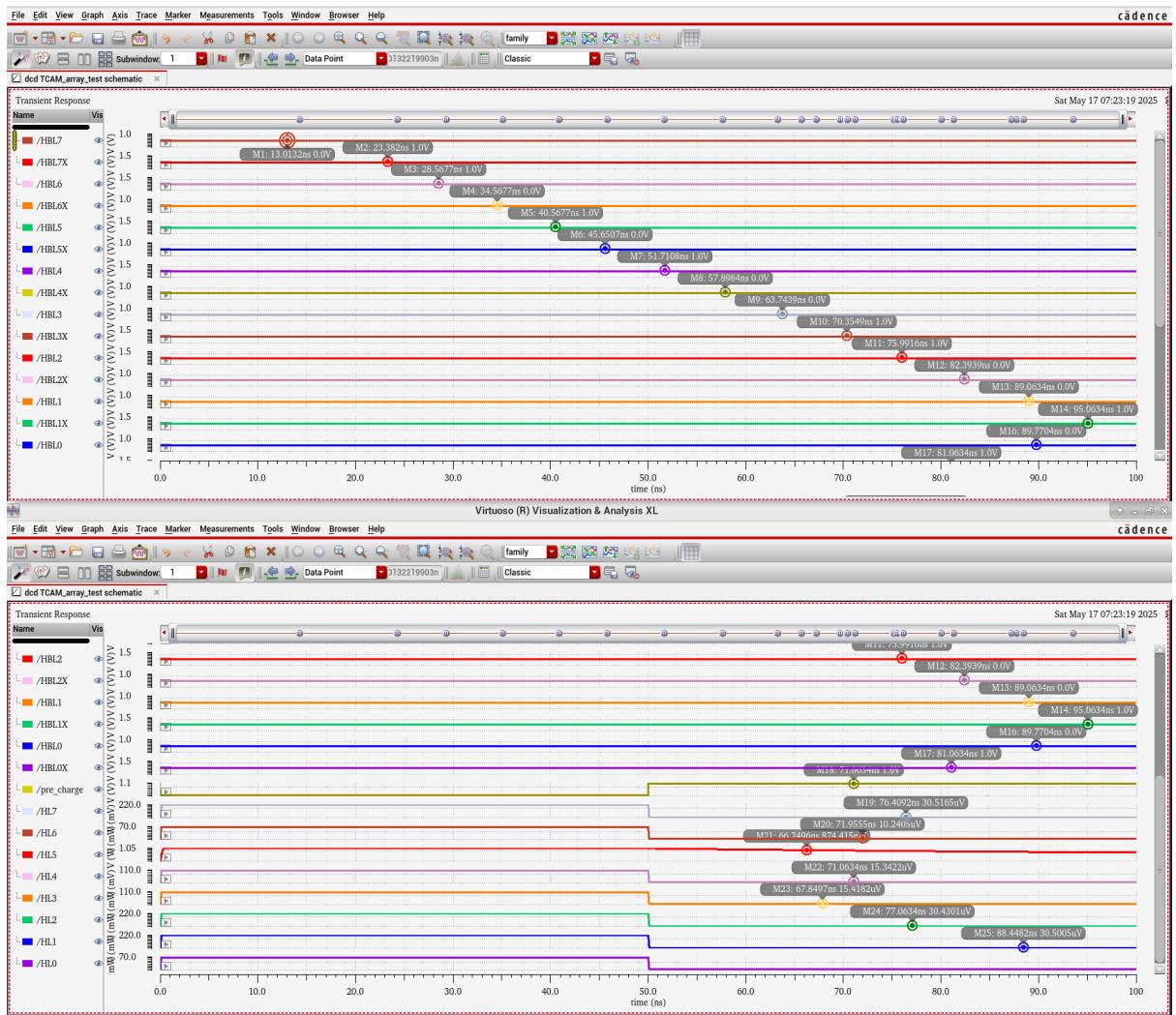
**Figure 2.7.1:** The schematic of the TCAM array after assigning the stored values

After the TCAM cells have been programmed with values, we proceed to search for the row in the TCAM array that matches a given bit pattern. The search data is provided through signals such as HBL7, HBL7X; HBL6, HBL6X; and so on.

When a row matches the input bit sequence, the corresponding HL signal for that row will be at logic level 1, indicating a Match state.

In the specific case based on the values previously written into the TCAM cells, the signal HL5 is at logic level 1, meaning that row 5 matches the search data.

We should add bypass capacitors at each HL output to reduce signal degradation and improve stability. A suitable capacitor value for this purpose could be 10 pF.



**Figure 2.7.2+2.7.3:** Waveform of TCAM array's Comperation operation

=> The waveform correctly demonstrates the compare operation of the TCAM array as expected. The signal HL5 is at logic level 1, corresponding to the **MATCH** state, while the HL output signals of the other rows remain at logic level 0, indicating the **MISS** state. This result matches the comparison value of 8'b01110100.

\**Additionally:* We can use the value X to represent the ALWAYS MATCH state for specific TCAM cells within the TCAM array. This is achieved simply by setting both the core cell and mask cell values of that TCAM cell to 0.