# A Circuit-to-Layout Design and Characterization of a 1T1R ReRAM Memory System with an Integrated Sense Amplifier

**Project Report submitted**

**In the Partial of Fulfillment of the Requirements for**

**ECE IOT – V Semester CMOS**

***by***

1. **BT23ECI037 Chilla Vivek Reddy**
2. **BT23ECI035 G Aprameya Goud**

***Under the Guidance of***

**Dr. Paritosh Peshwe**

**Indian Institute of Information Technology, Nagpur**

## Introduction:

**Research Device:** Operates with very low resistances (LRS is 4-14 Ω, HRS is 30-70 Ω) and very high currents (compliance currents of 20-100 mA). This is typical for certain metal-oxide RRAMs intended for high-performance applications.

**My Simulation:** You modeled much higher resistances (LRS = 1 kΩ, HRS = 100 kΩ) and lower currents. Your values are actually very common and practical for modeling large memory arrays, where low power consumption is essential.

## Observations & Results:

** Circuit Schematic

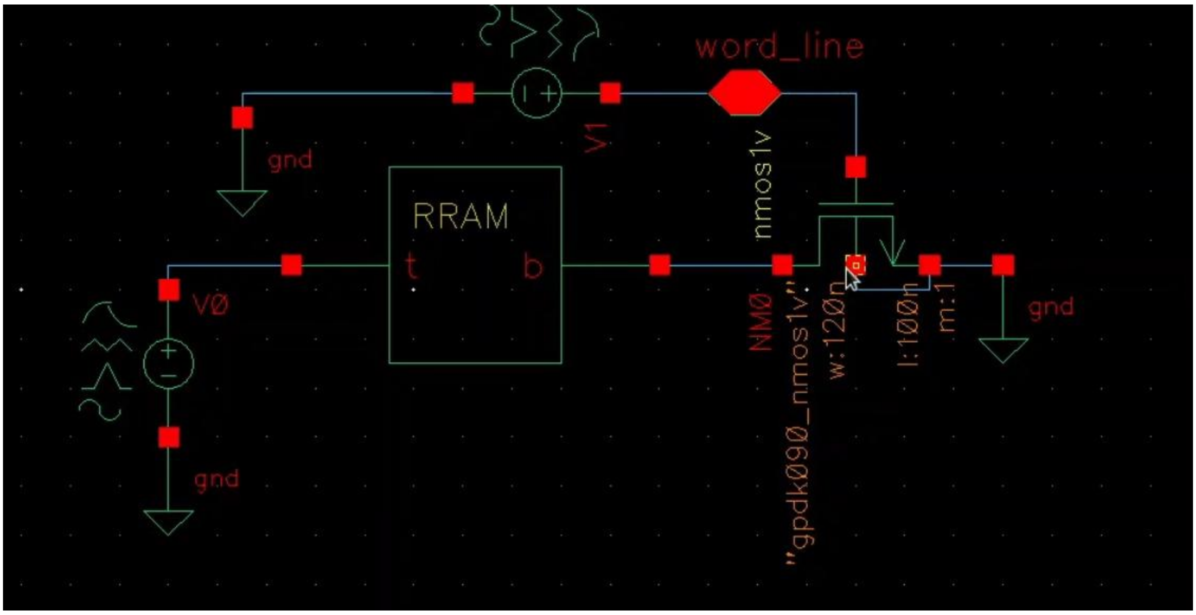


Fig.1: 1T1R ReRAM Cell Schematic – Integration of a ReRAM device with an NMOS access

*** A 1T1R circuit schematic for RRAM (Resistive Random Access Memory) includes one access transistor and one resistive memory cell, enabling precise cell selection and preventing unwanted current leakage during read/write operations. The access transistor ensures only the targeted RRAM cell is activated, thereby avoiding sneak current paths that could cause data corruption or read errors in a large crossbar array. This configuration provides better reliability and lower error rates compared to selectorless architectures, making it a preferred choice for practical, scalable RRAM designs.**

** Ngspice simulation of above ckt

| Time Interval (μs) | Bitline Voltage (V) | Operation Expected Outcome |
|---|---|---|
| 0.0 – 1.1 | 0V | Idle / Initial State No change. The RRAM is in its initial High Resistance State (HRS). |
| 1.1 – 2.0 | +0.2V | Sense the initial state. A very low current flows, confirming HRS. |
| 2.1 – 3.1 | 0V | The cell retains its HRS state with zero power |
| 3.1 – 4.0 | +1.5V | Switch the RRAM from HRS to the Low Resistance State (LRS). |
| 4.1 – 5.1 | 0V | The cell now retains its new LRS state with zero power. |
| 5.1 – 6.0 | +0.2V | Sense the new state. A high current flows, confirming LRS. |
| 6.1 – 7.1 | 0V | The cell continues to retain the LRS state with zero power. |
| 7.1 – 8.0 | -1.5V | Switch the RRAM from LRS back to HRS |
| 8.1 – 9.1 | 0V | The cell now retains its erased HRS state with zero power. |
| 9.1 – 10.0 | +0.2V | Sense the final state. A very low current flows, confirming the cell is back in HRS. |

***For NGSpice simulation, a static RRAM model is often used instead of a memristor or physics-based RRAM because NGSpice cannot replicate the stochastic and random nature of physical RRAM switching behavior. Consequently, the simulated results remain fixed or predictable, reflecting only the simple ON/OFF states, rather than real device variability. This approach allows for circuit-level validation, but does not capture the randomness and variability observed in actual fabricated RRAM device**

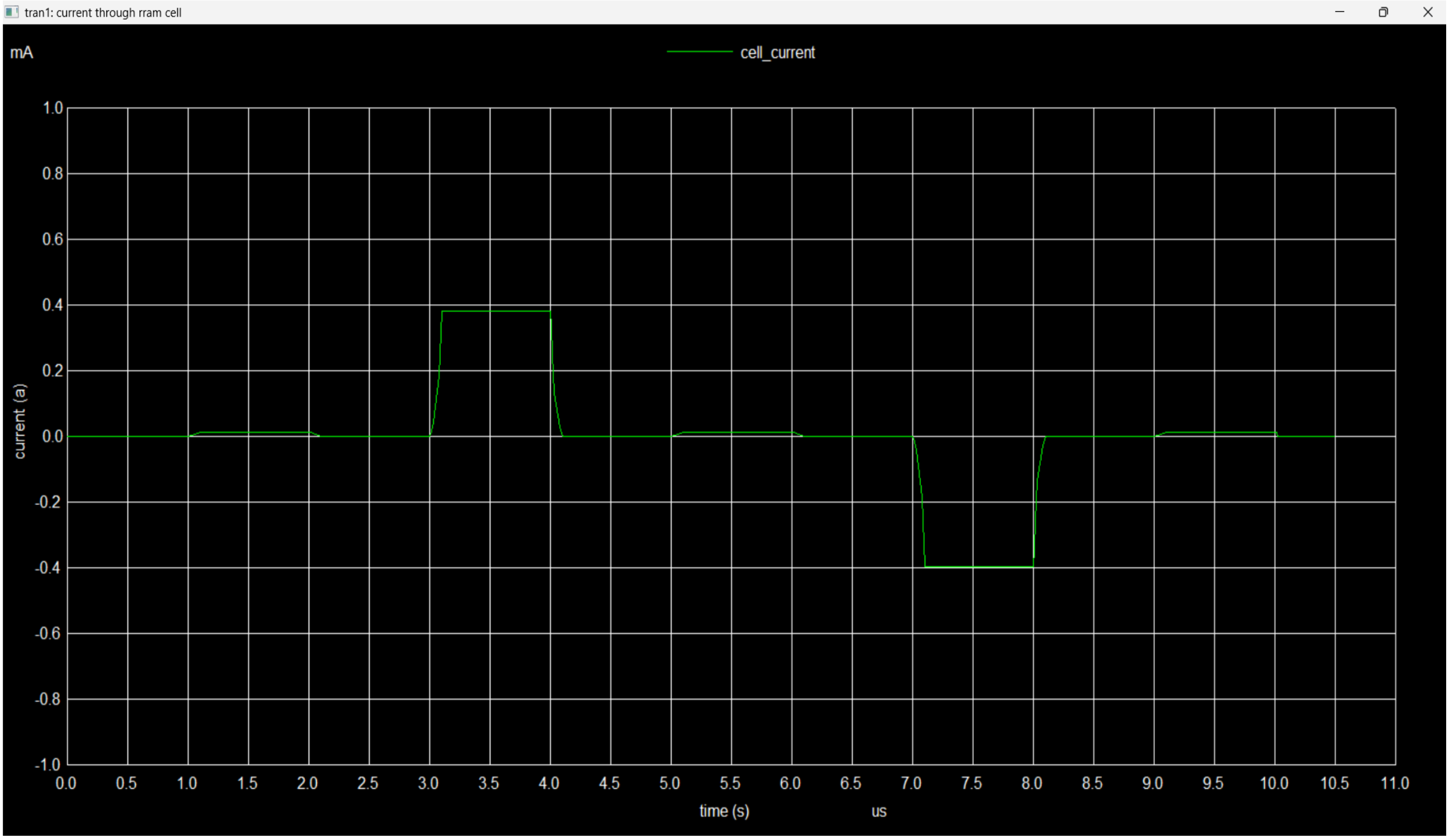


Fig.2: Current across RRAM

.

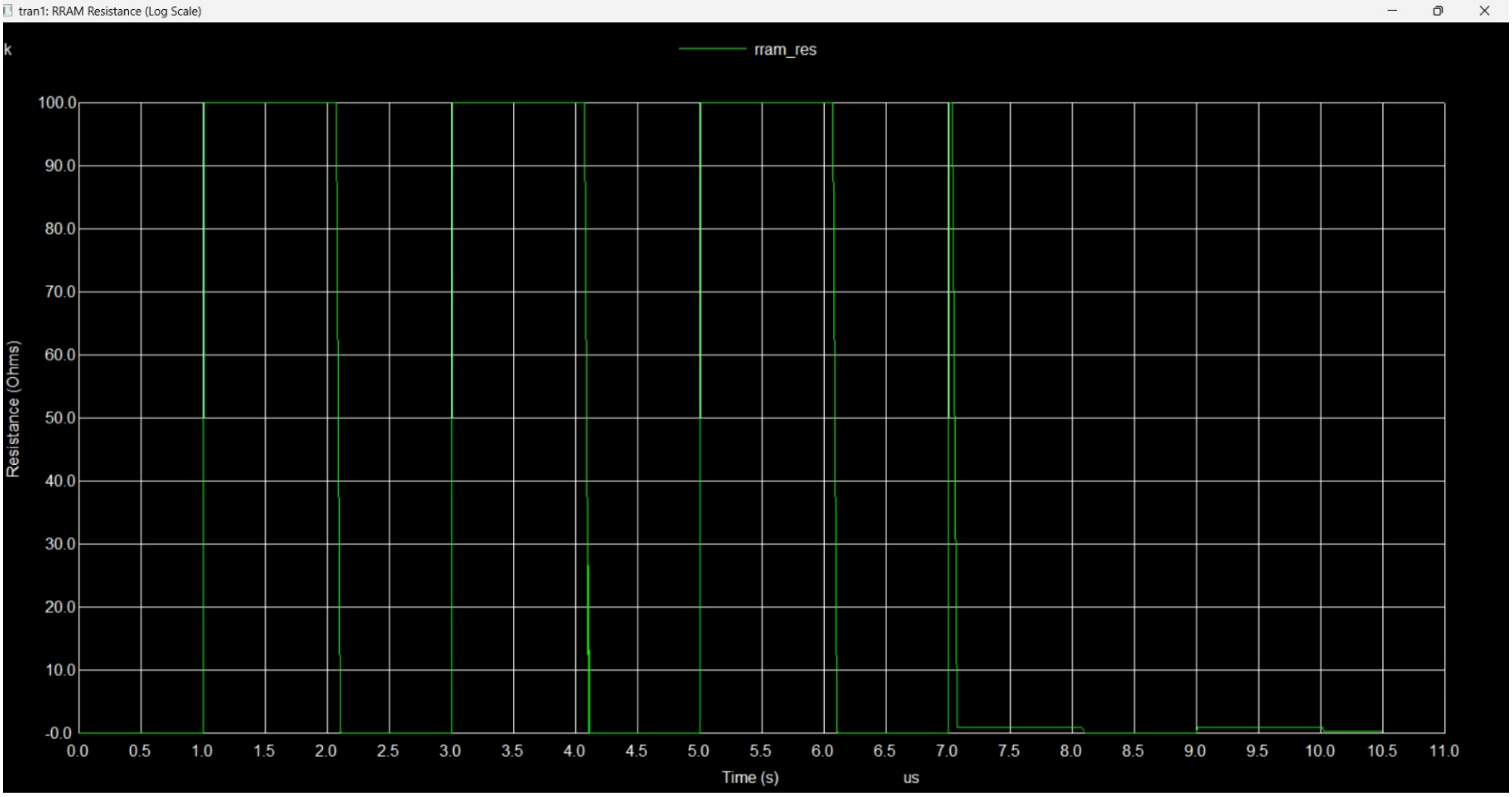


Fig.3: RRAM Resistance (Log Scale)

The resistance of the ReRAM cell alternates between high and low states, showing clear switching behavior that represents the memory's binary storage characteristics

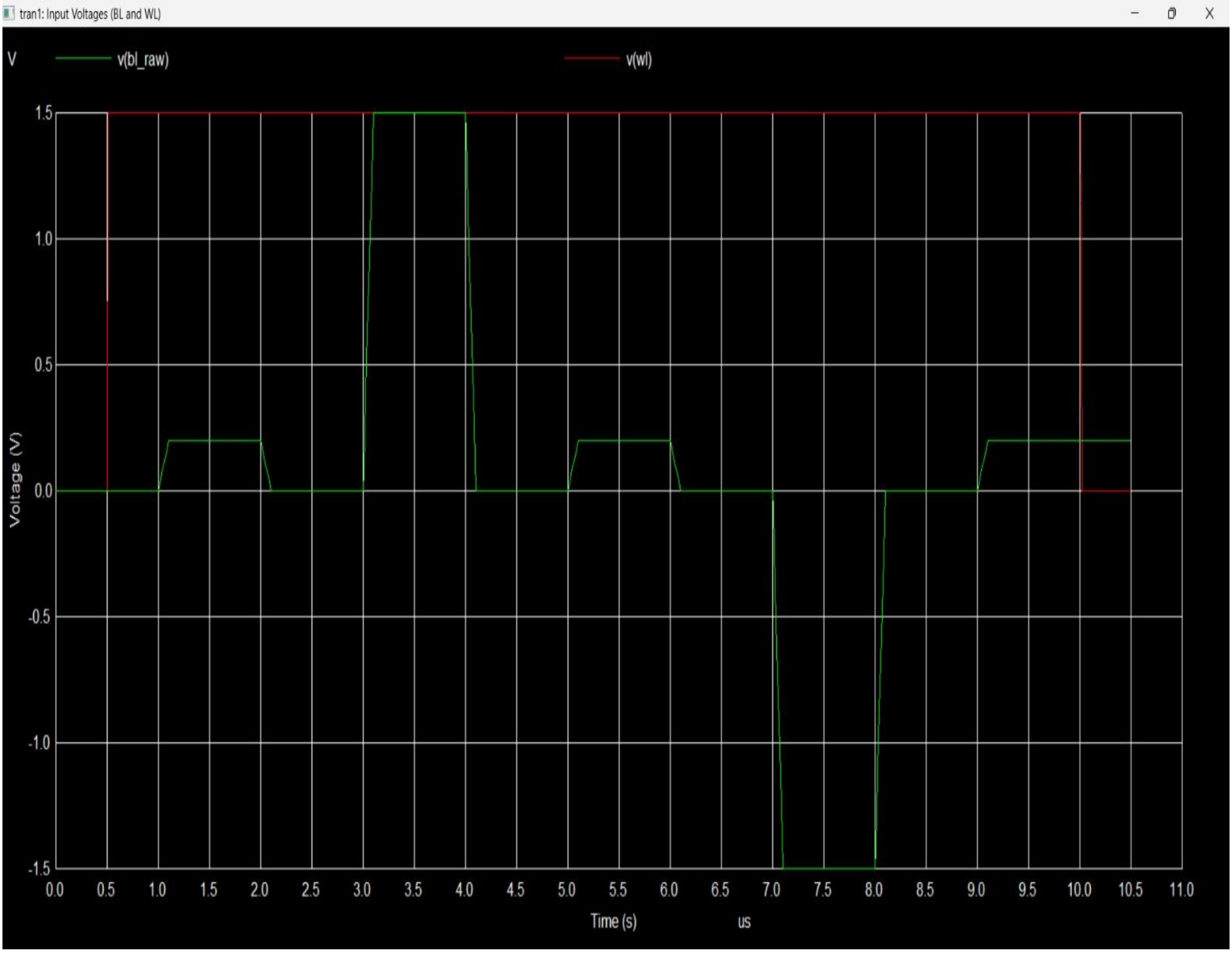


Fig.4: Input Voltages (Word Line and Bit Line)

The applied BL and WL voltages control the SET and RESET operations of the RRAM cell, determining its transition between HRS and LRS.

Netlist (Reference : standford standard Verilog-A model )

verilog-A model

```
* ---------------- Global parameters ----------------
.param Ron=1k      ; Low Resistance State (LRS)
.param Roff=100k   ; High Resistance State (HRS)
.param Vth_nmos=0.4  ; Transistor threshold voltage

* ---------------- RRAM SUBCIRCUIT (Simplified & Stable Model) ----------------
* p = top terminal, n = bottom terminal
.SUBCKT RRAM p n PARAMS: Vset=1.0 Vreset=-1.0 Ron_sub={Ron} Roff_sub={Roff}

* --- Part 1: Create Trigger Signals ---
B_s s_trig 0 V = { (V(p,n) > Vset) ? 1V : 0V }
B_r r_trig 0 V = { (V(p,n) < Vreset) ? 1V : 0V }

* --- Part 2: A Stable SR Latch to Hold the State ---
* FIX: Replaced ideal switches with smooth tanh() functions to prevent convergence errors.
* This creates "analog-friendly" NOR gates that ngspice can solve.
B_nor1 q 0 V = { 0.5 * (1 - tanh(20 * (max(V(q_bar), V(s_trig)) - 0.5))) }
B_nor2 q_bar 0 V = { 0.5 * (1 - tanh(20 * (max(V(q), V(r_trig)) - 0.5))) }
* Resistors to ensure nodes are not floating
```

```
Rq q 0 1Meg
Rqbar q_bar 0 1Meg

* --- Part 3: The Switchable Resistor ---
* If V(q) is high ( > 0.5V ), resistance is Ron. Otherwise, it's Roff.
G_mem p n VALUE = { V(p,n) / ( (V(q) > 0.5) ? Ron_sub : Roff_sub ) }

.ENDS RRAM

* ---------------- Top-level 1T1R Circuit ----------------
* Wordline (gate) pulse
Vwl wl 0 PULSE(0 1.5 0.5u 10n 10n 9.5u 20u)

* Bitline source (Formatted vertically with comments)
Vbl_src bl_raw 0 PWL(
+ 0s    0V     ; Initial state at t=0
+ 1.0u  0V
+ 1.1u  0.2V   ; t=1.1us: READ 1 (Check initial HRS)
+ 2.0u  0.2V
+ 2.1u  0V
+ 3.0u  0V
+ 3.1u  1.5V   ; t=3.1us: SET Pulse (Write to LRS)
+ 4.0u  1.5V
+ 4.1u  0V
+ 5.0u  0V
+ 5.1u  0.2V   ; t=5.1us: READ 2 (Confirm LRS)
+ 6.0u   0.2V
+ 6.1u   0V
+ 7.0u   0V
+ 7.1u  -1.5V    ; t=7.1us: RESET Pulse (Erase to HRS)
+ 8.0u  -1.5V
+ 8.1u   0V
+ 9.0u   0V
+ 9.1u   0.2V    ; t=9.1us: READ 3 (Confirm HRS)
+ 10.0u  0.2V
+ )

* Zero-volt source to measure Bitline current
Vmeas bl bl_raw DC 0

* Instantiate the RRAM subcircuit
XR1 bl bl_cell RRAM

* Access NMOS Transistor (W=1u for strong drive)
M1 bl_cell wl sl sl nmos_model W=1u L=50n

* Ground for the source of the transistor
Vsl sl 0 DC 0

* Transistor model
.MODEL nmos_model NMOS (LEVEL=1 VTO={Vth_nmos} KP=120e-6)
* ---------------- Simulation Control ----------------
```

```
.options reltol=1e-3
.tran 50n 10.5u
.control
run
* ----- Create variables for plotting -----
let cell_current = -i(Vmeas)
let rram_res = (v(bl)-v(bl_cell)) / (cell_current + 1p)
* ----- Plots -----
* Plot 1: Input Voltages
plot v(bl_raw) v(wl) title 'Input Voltages (BL and WL)' xlabel 'Time (s)' ylabel 'Voltage (V)'
* Plot 2: Cell Current
plot cell_current ylimit -250u 250u title 'Current through RRAM Cell' xlabel 'Time (s)' ylabel 'Current (A)'
* Plot 3: RRAM Resistance
set ylog
plot rram_res title 'RRAM Resistance (Log Scale)' xlabel 'Time (s)' ylabel 'Resistance (Ohms)'
unset ylog
.endc
.end
```

**** Latch type sensitive amplifier -ckt schematic for Reading accurate data for small sensing margin**

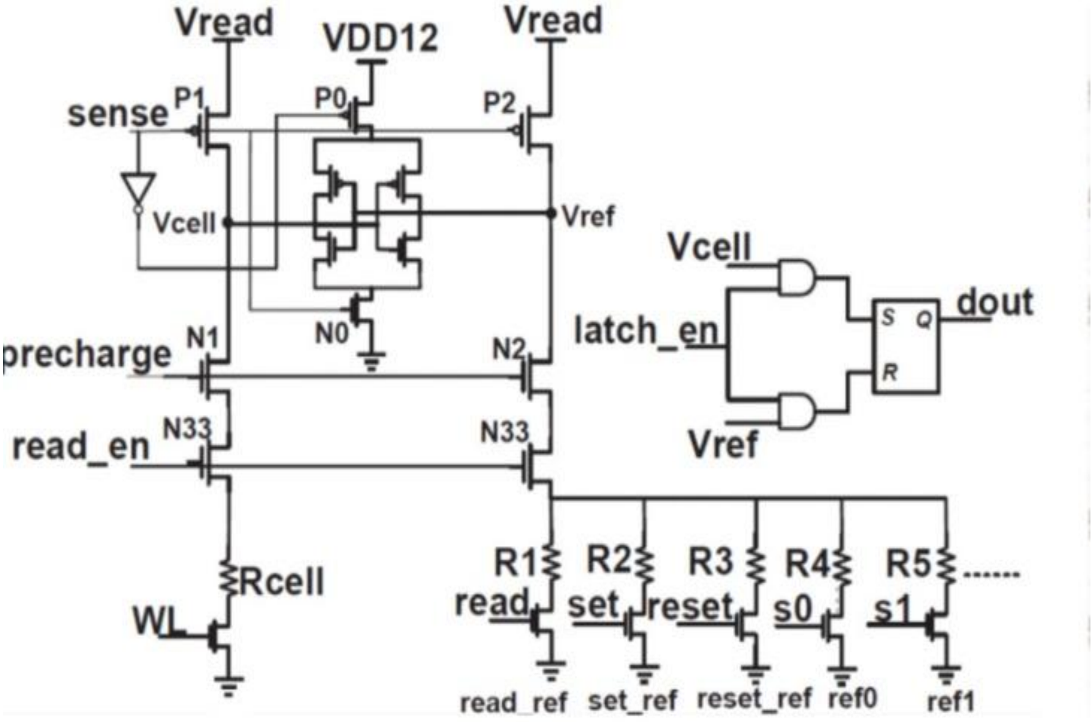
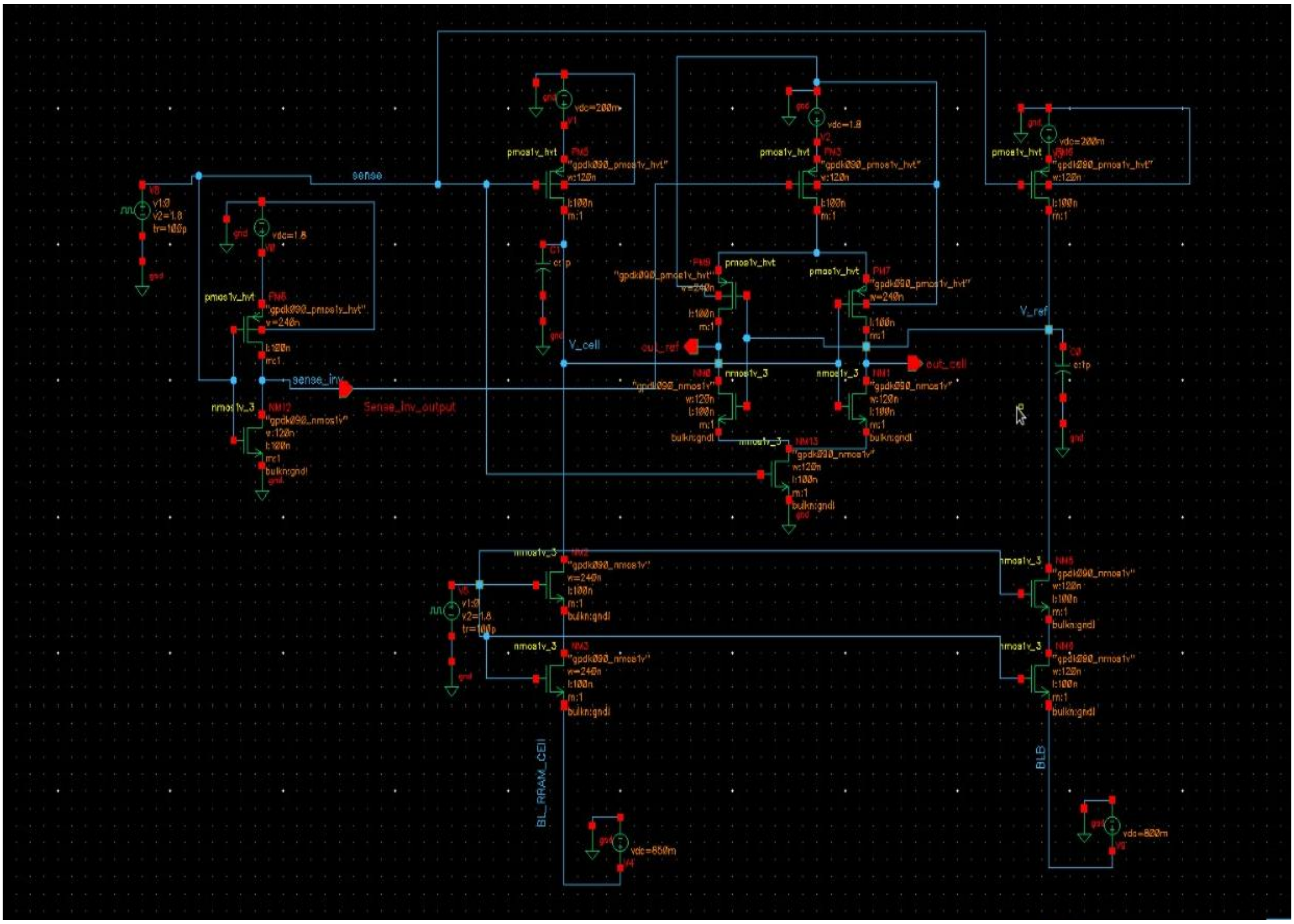


Fig.5: Latch type Sense Amplifier Circuit

- **Rcell (RRAM cell)**: This component represents the non-volatile resistive memory element. Its resistance encodes the stored data, changing between high (RESET, logic 0) and low (SET, logic 1) states. The word line (WL) selects the desired cell for read or write operation.
- **Reference Resistors (R1–R5)**: These resistors provide precise reference voltages for the sense amplifier and SR latch. Each reference (read_ref, set_ref, reset_ref, ref0, ref1, etc.) is used to compare against the cell voltage during the read cycle, improving detection reliability between stored states.
- **Selection Transistors** (below each reference resistor): These switches (NMOS) connect a particular reference resistor to ground, enabling selection of which reference voltage is fed into the sense circuit—critical for accurate sensing and distinguishing bit values of RRAM.

**This configuration ensures robust readout by balancing the cell voltage against multiple reference points, reducing misread errors caused by device variations.

**Also RRAM cell is not dynamic and is not favourable to use static Resistor in circuit simulation of sense amplifer , hence used custom inputs required for HRS (High resistive state) and LRS (Low resistive state)



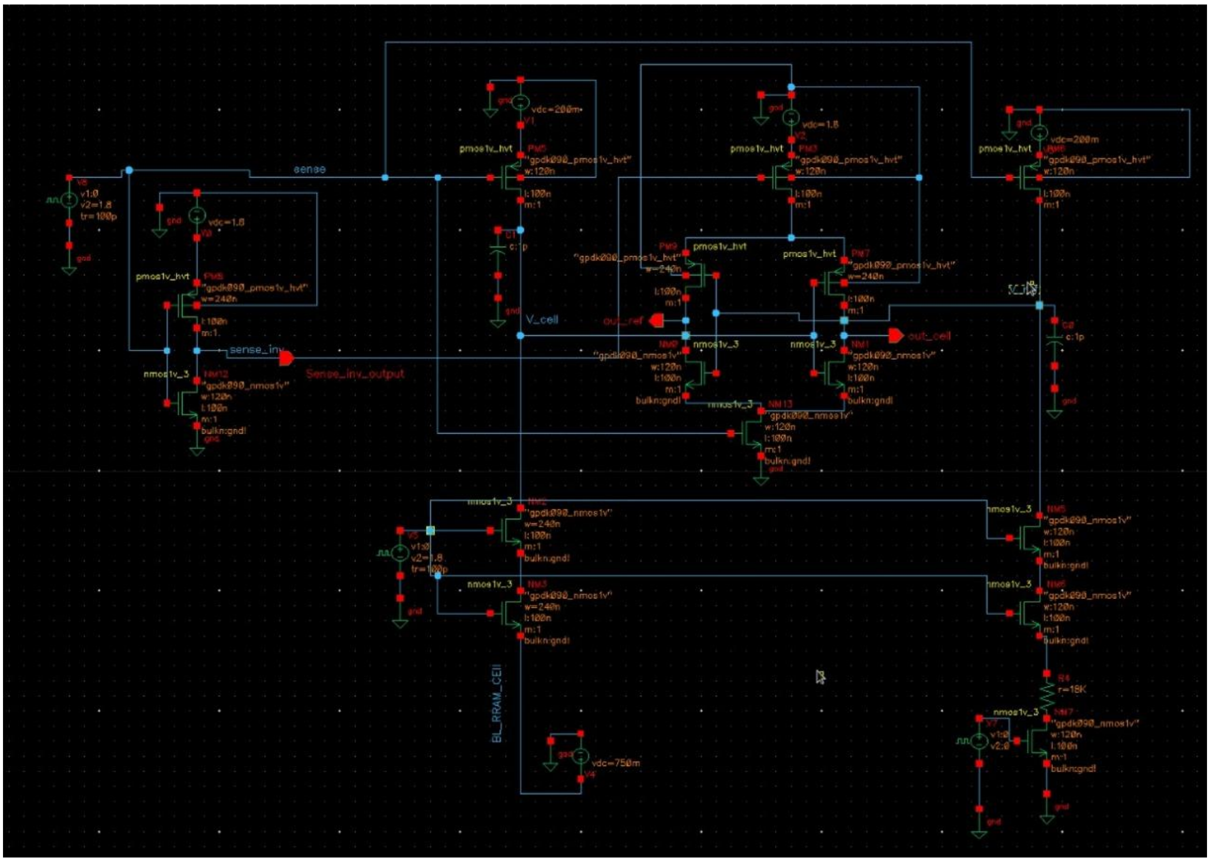


Fig.6: Detailed transistor-level design of the sense amplifier implemented in Cadence Virtuoso for post-layout simulation and performance characterization.

**Simulation for checking correctness of circuit (keeping BL=0.85 v & BLB = gnd)

**Start with prechrg (0-5ns ), then sense(evaluation ) (5-10ns )

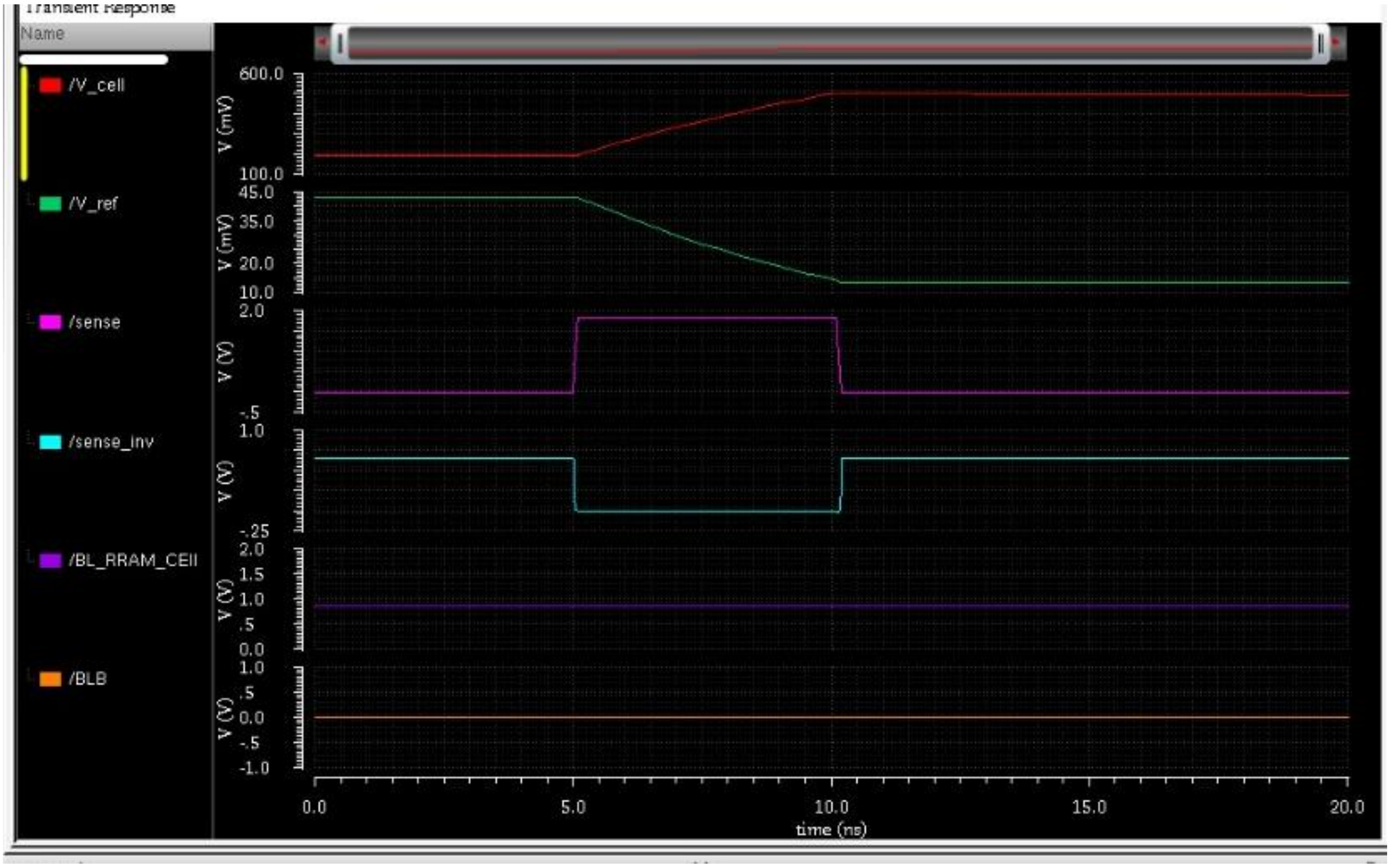
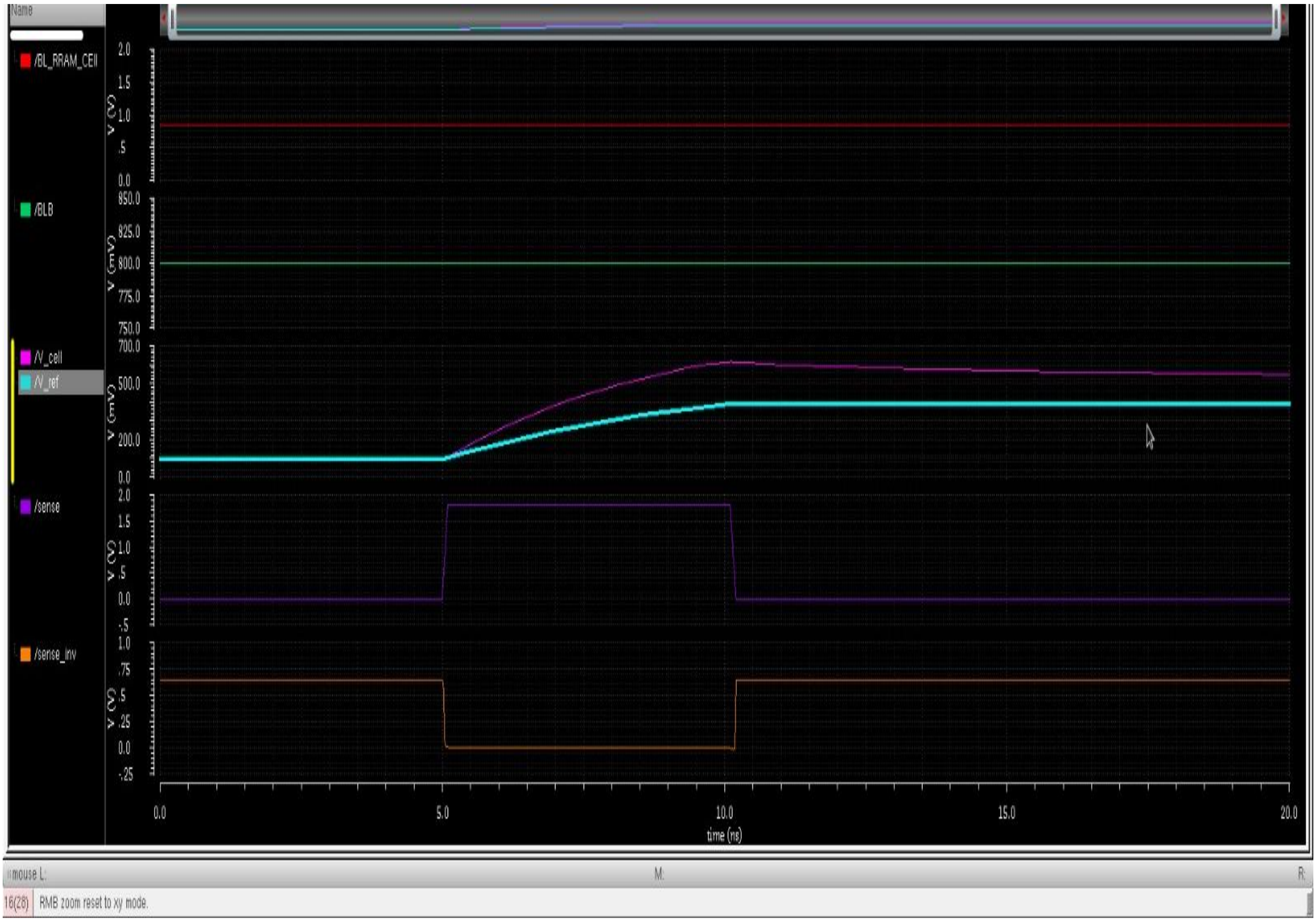


Figure.7: Transient Response of Sense Amplifier

Simulation of the integrated sense amplifier showing voltage differentials between cell and reference lines, validating its ability to distinguish resistive states during read operation.

**Simulation for checking correctness of circuit (keeping BL=0.85 v & BLB =0.80v)



16(28) RMB zoom reset to xy mode.

Figure.8: Transient Response with Modified Resistor Width

The variation in transistor width **(double the read_en and precharge nmos transistor width on BL side )**affects the sense amplifier’s sensitivity

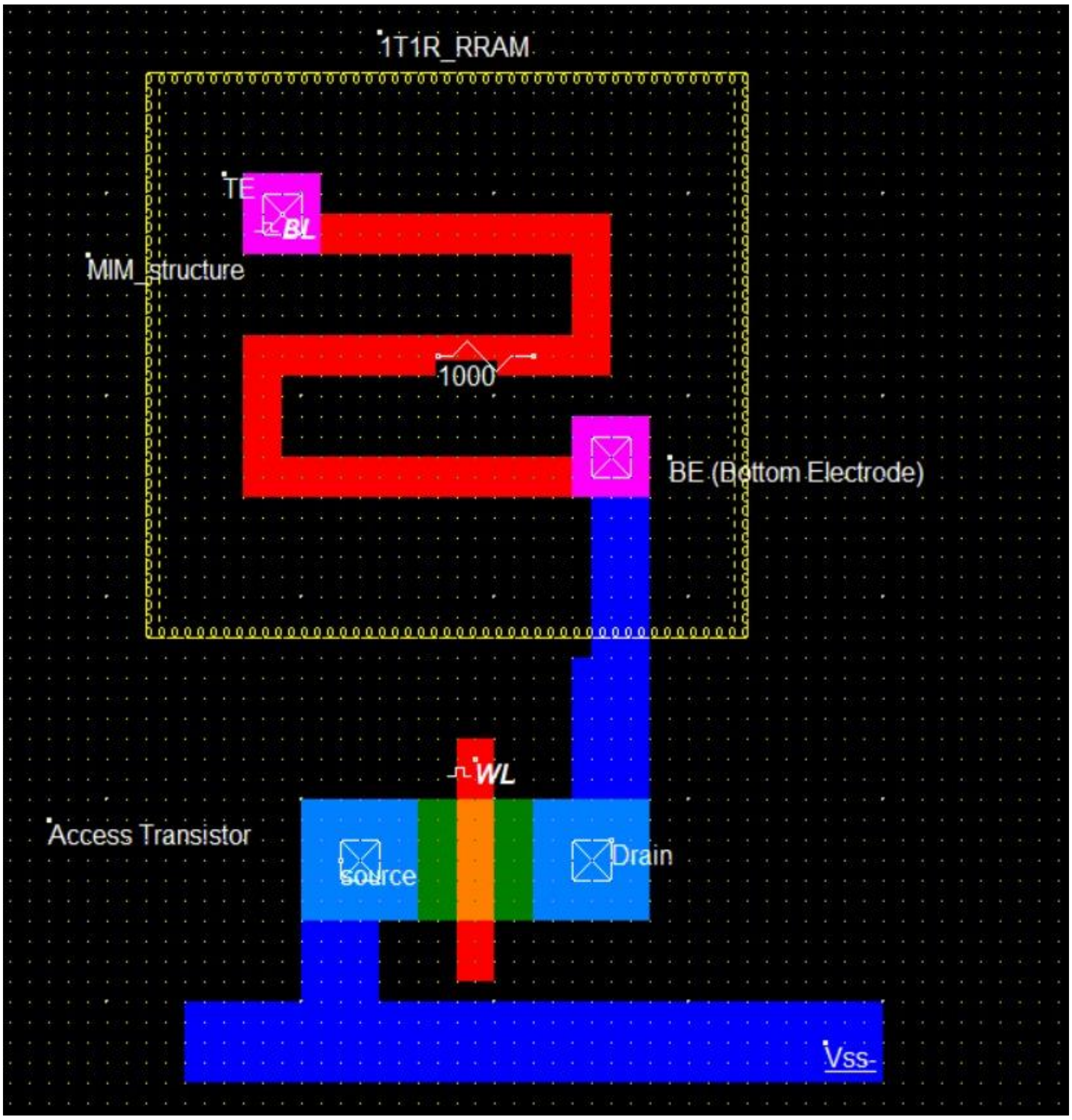
### Core Amplifier Unit Test

- **Objective:** Validate amplifier sensitivity.
- **Setup:** RRAM/Reference disconnected. DC voltage sources applied to inputs bl and blb.
- **LRS Test:** bl = 0.85V, blb = 0.8V
- **HRS Test:** bl = 0.75V, blb = 0.8V

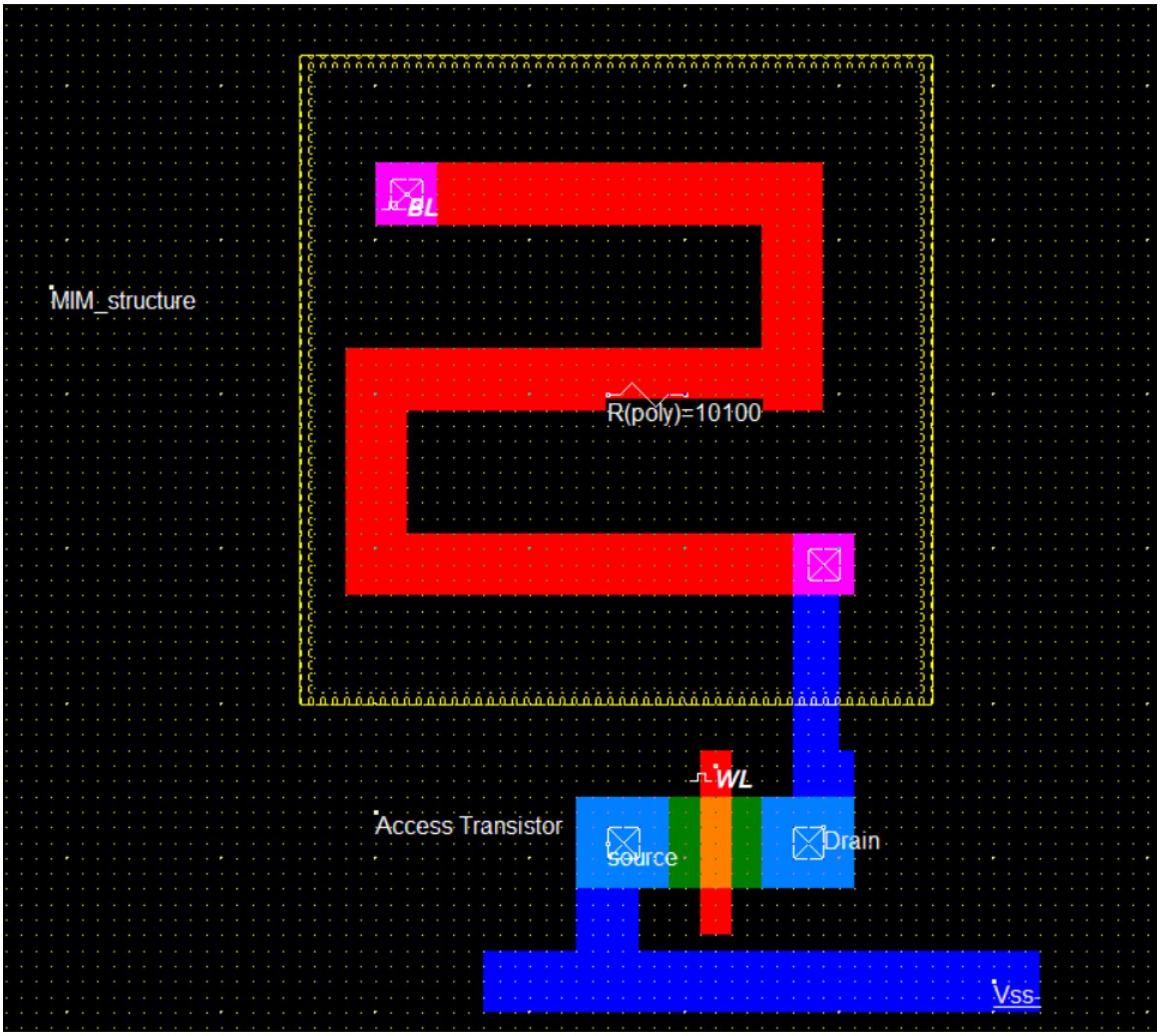
** Layout Design

**1T1R ReRAM Cell Layout design:** (45nm tech node)

**LRS :**



**HRS:**



### Step 1: Simulate the "SET" Operation (Write '1')

The goal of the SET operation is to apply a high voltage that would force a real RRAM into its **Low Resistance State (LRS)**.

- **How to Simulate:**
  1. Keep your **Word Line (WL)** high to keep the transistor ON. A pulse from 0V to 1.2V is perfect.
  2. Apply your high voltage **Bit Line (BL)** pulse. You mentioned 1.5V, so apply a pulse from 0V to 1.5V.

- **What to Observe:**
    - Measure the current that flows through the cell during this pulse. This is your "SET current."
    - In a real device, this action would lower the resistance. In your simulation, this step is just to verify the electrical conditions for a SET pulse.

**Step 2: Manually Create the LRS and "READ" It**

Now, you will pretend the SET operation worked. You need to manually change your RRAM proxy to a low resistance and then perform a gentle READ to see the result.

- **Manual Action:**
    1. Go back to your layout editor.
    2. Select the polysilicon layer you are using as the resistor.
    3. **Manually change its resistance to a low value**, for example, **R_{LRS} = 1 kΩ**.
- **How to Simulate the READ:**
    1. Run the simulation again.
    2. Set the **WL** high (0V to 1.2V pulse) to access the cell.
    3. Apply a **low voltage READ pulse** to the **BL**. It's critical to use a low voltage (e.g., **0.2V**) so you don't disturb the state.

- **What to Observe:**
    - Measure the current. You should see a relatively **HIGH current**.
    - For example (expected behaviour): I_{LRS} = V_{READ} / R_{LRS} = 0.2V / 1 kΩ = 0.2 mA. This high current represents a stored '1'**. (Expected** ).

### Step 3: Simulate the "RESET" Operation (Write '0') /(considering unipolar )

The goal of the RESET operation is to apply a specific voltage that would force the RRAM into its **High Resistance State (HRS)**.

- **How to Simulate:**
  1. Set the **WL** high (0V to 1.2V pulse).
  2. Apply a RESET voltage pulse to the **BL**. This is typically a lower positive voltage than SET (like **0.8V**). Let's use a 0.8V pulse for this example.
- **What to Observe:**
  - Measure the current that flows during this pulse. This is your "RESET current."
  - Again, this step just verifies the electrical conditions for the RESET pulse.

### Step 4: Manually Create the HRS and "READ" It

Now, you pretend the RESET worked. Manually change your proxy to a high resistance and perform the same READ operation.

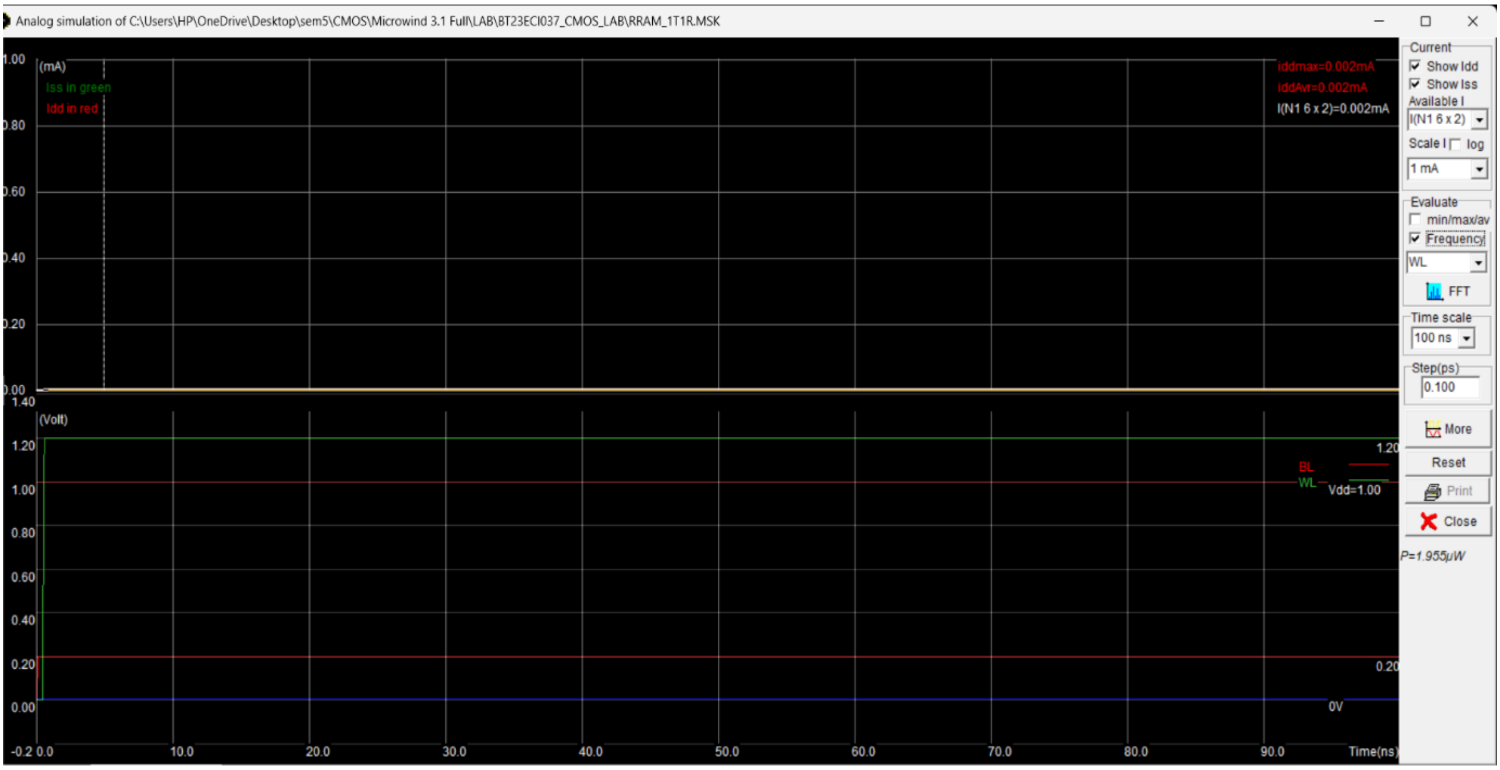
- **Manual Action:**
  1. Go back to your layout editor.
  2. Select the polysilicon resistor again.
  3. **Manually change its resistance to a high value**, for example, **R_{HRS} = 100 kΩ. (A 10x to 100x ratio between HRS and LRS is common).**
- **How to Simulate the READ:**
  1. Run the simulation with the *exact same READ conditions as in Step 2*.
  2. Set the **WL** high (0V to 1.2V pulse).
  3. Apply the same **low voltage READ pulse** to the **BL** (e.g., **0.2V**).

- **What to Observe:**
    - Measure the current. You should now see a **VERY LOW current**.
    - For example: I_{HRS} = V_{READ} / R_{HRS} = 0.2V / 100 kΩ = 2 µA. This low current represents a stored '0'.**(Expected)**

| **Operation** | **Description** | **Measured Current** | **Corresponding State** |
|---|---|---|---|
| **SET** | Programming the cell to a low resistance state. | 0.015 mA (15 µA) | (Programming Current) |
| **READ '1'** | Reading the cell after the SET operation. | **0.177 mA**<br>***(expected 0.2mA)** | **LRS (Logic '1')** |
| **RESET** | Programming the cell to a high resistance state. | 0.149 mA~0.15mA | (Programming Current) |
| **READ '0'** | Reading the cell after the RESET operation. | **0.002 mA (2 µA)** | **HRS (Logic '0')** |

Fig.9: Layout design results of 1T1R

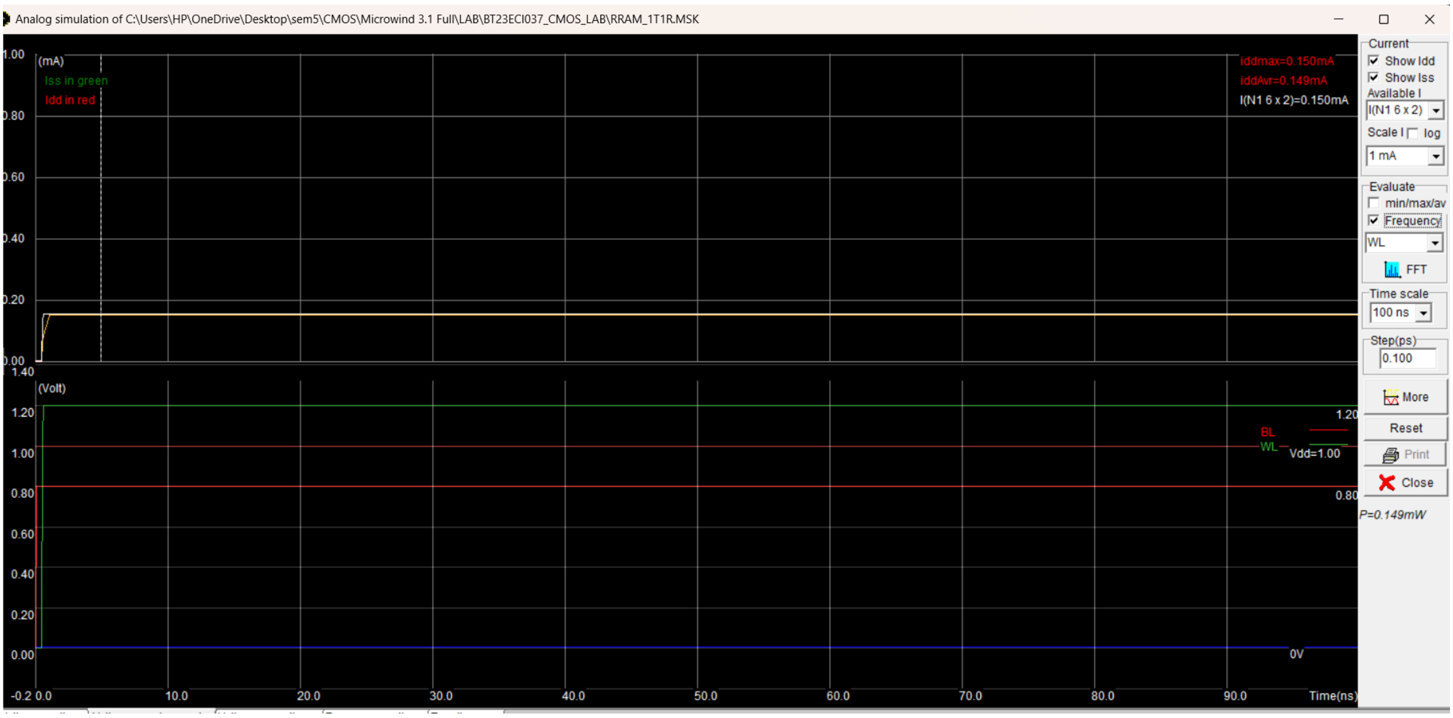
Output Waveforms:


Analog simulation of C:\Users\HP\OneDrive\Desktop\sem5\CMOS\Microwind 3.1 Full\LAB\BT23ECI037_CMOS_LAB\RRAM_1T1R.MSK

iddmax=0.002mA
iddAvr=0.002mA
I(N1 6 x 2)=0.002mA

Current
☑ Show Idd
☑ Show Iss
Available I: I(N1 6 x 2)
Scale I □ log: 1 mA
Evaluate
□ min/max/av
☑ Frequency: WL
Time scale: 100 ns
Step(ps): 0.100
P=1.955µW

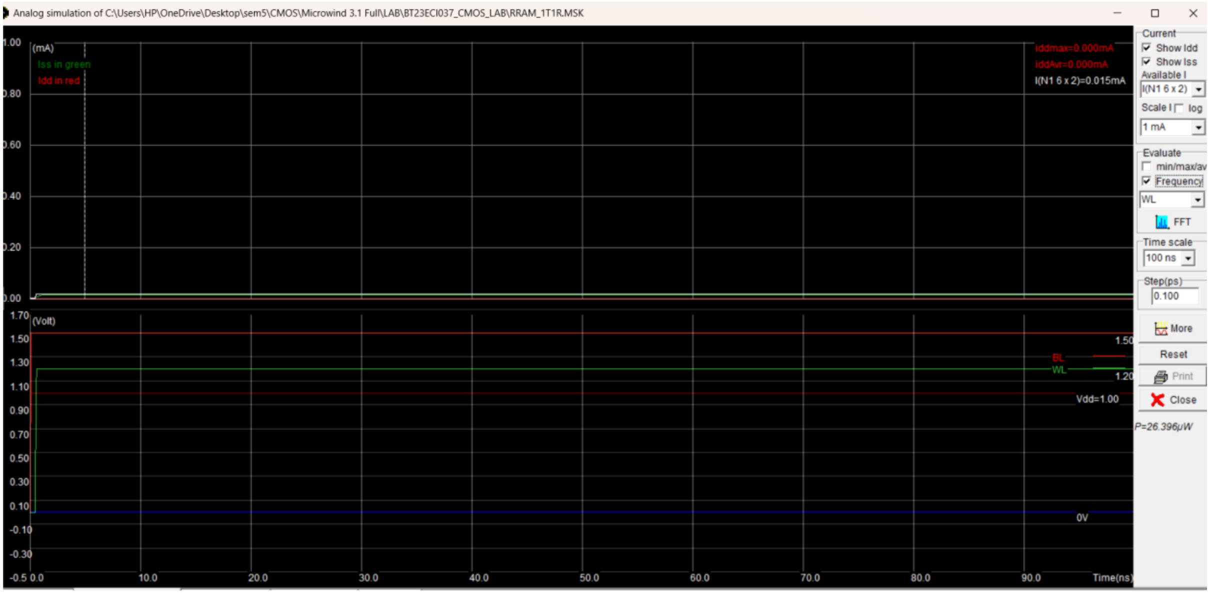
Fig.10: Read ‘0’ HRS


Analog simulation of C:\Users\HP\OneDrive\Desktop\sem5\CMOS\Microwind 3.1 Full\LAB\BT23ECI037_CMOS_LAB\RRAM_1T1R.MSK

iddmax=0.150mA
iddAvr=0.149mA
I(N1 6 x 2)=0.150mA

Current
☑ Show Idd
☑ Show Iss
Available I: I(N1 6 x 2)
Scale I □ log: 1 mA
Evaluate
□ min/max/av
☑ Frequency: WL
Time scale: 100 ns
Step(ps): 0.100
P=0.149mW

Fig.11: Reset current (current during rst operation)

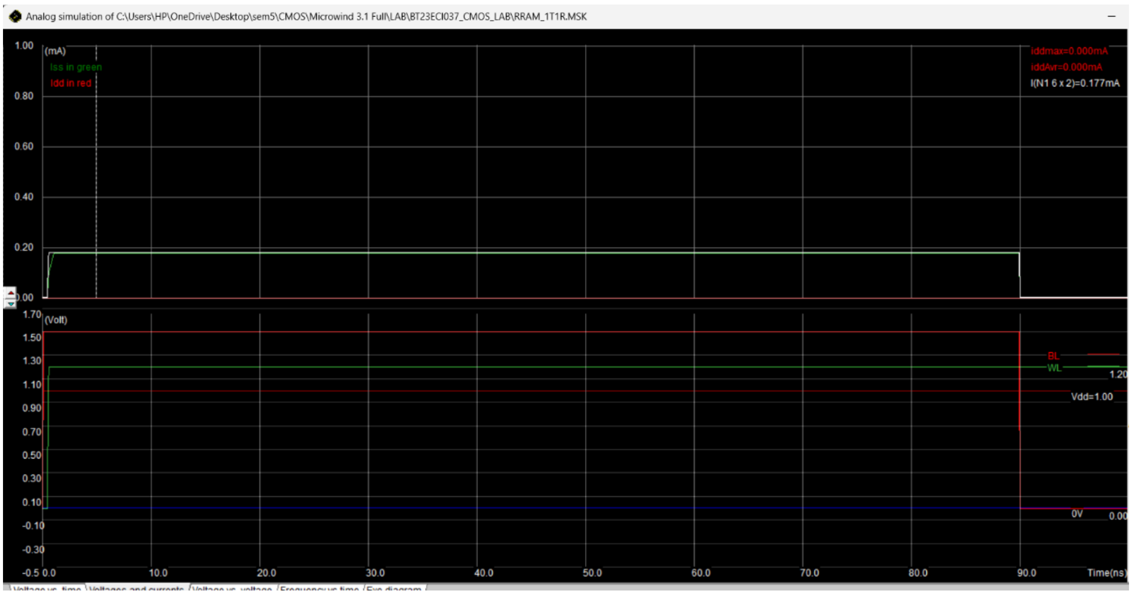


Analog simulation of C:\Users\HP\OneDrive\Desktop\sem5\CMOS\Microwind 3.1 Full\LAB\BT23ECI037_CMOS_LAB\RRAM_1T1R.MSK

iddmax=0.000mA
iddAvr=0.000mA
I(N1 6 x 2)=0.015mA

Current
☑ Show Idd
☑ Show Iss
Available I: I(N1 6 x 2)
Scale I □ log: 1 mA
Evaluate
□ min/max/av
☑ Frequency: WL
Time scale: 100 ns
Step(ps): 0.100
P=26.396µW

Fig.12: set current (current during set operation)



Analog simulation of C:\Users\HP\OneDrive\Desktop\sem5\CMOS\Microwind 3.1 Full\LAB\BT23ECI037_CMOS_LAB\RRAM_1T1R.MSK

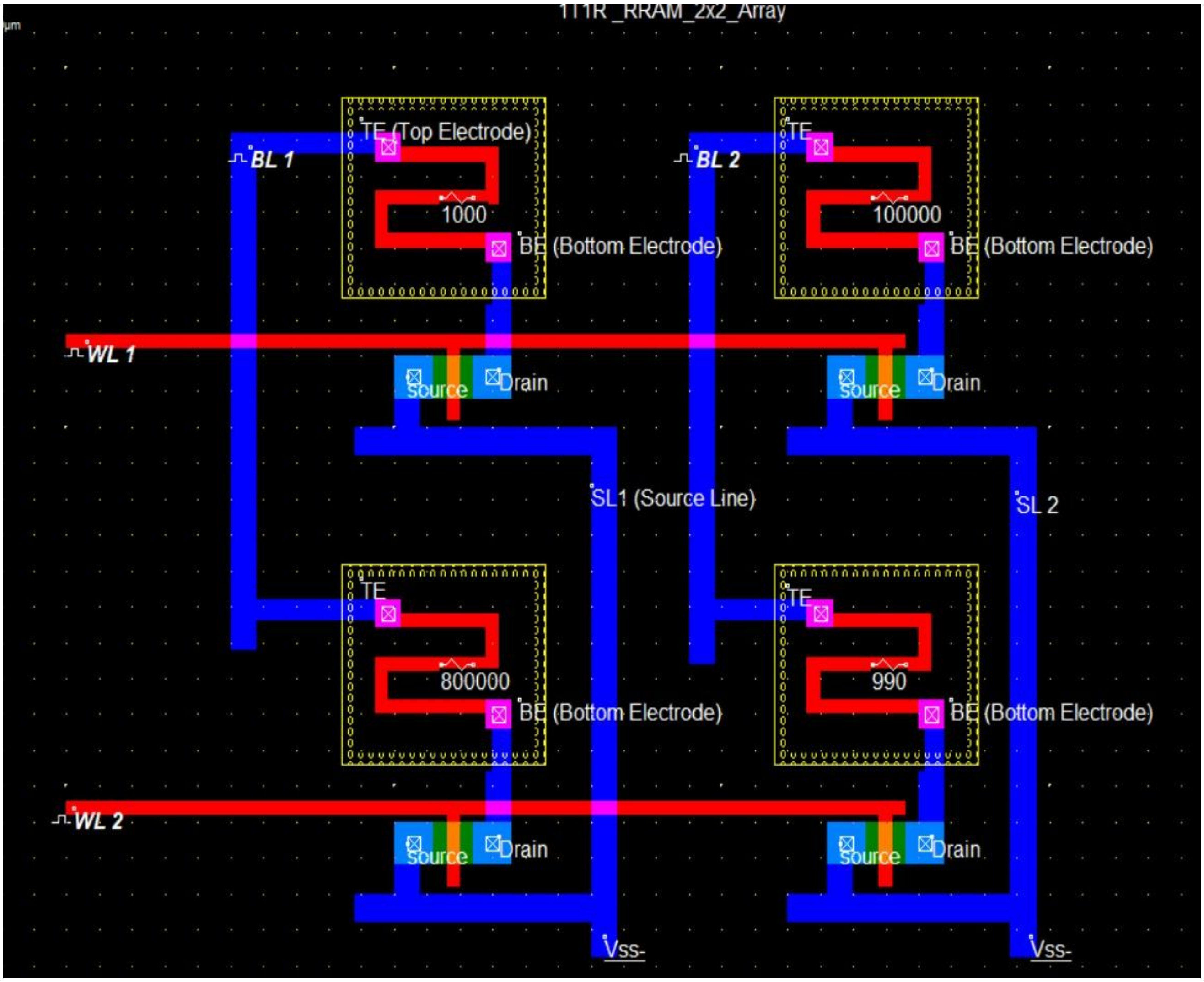
iddmax=0.000mA
iddAvr=0.000mA
I(N1 6 x 2)=0.177mA

Fig.13: Read ’1’ LRS

### 2×2 1T1R ReRAM Array Layout

### ** RRAM cell states are manually set

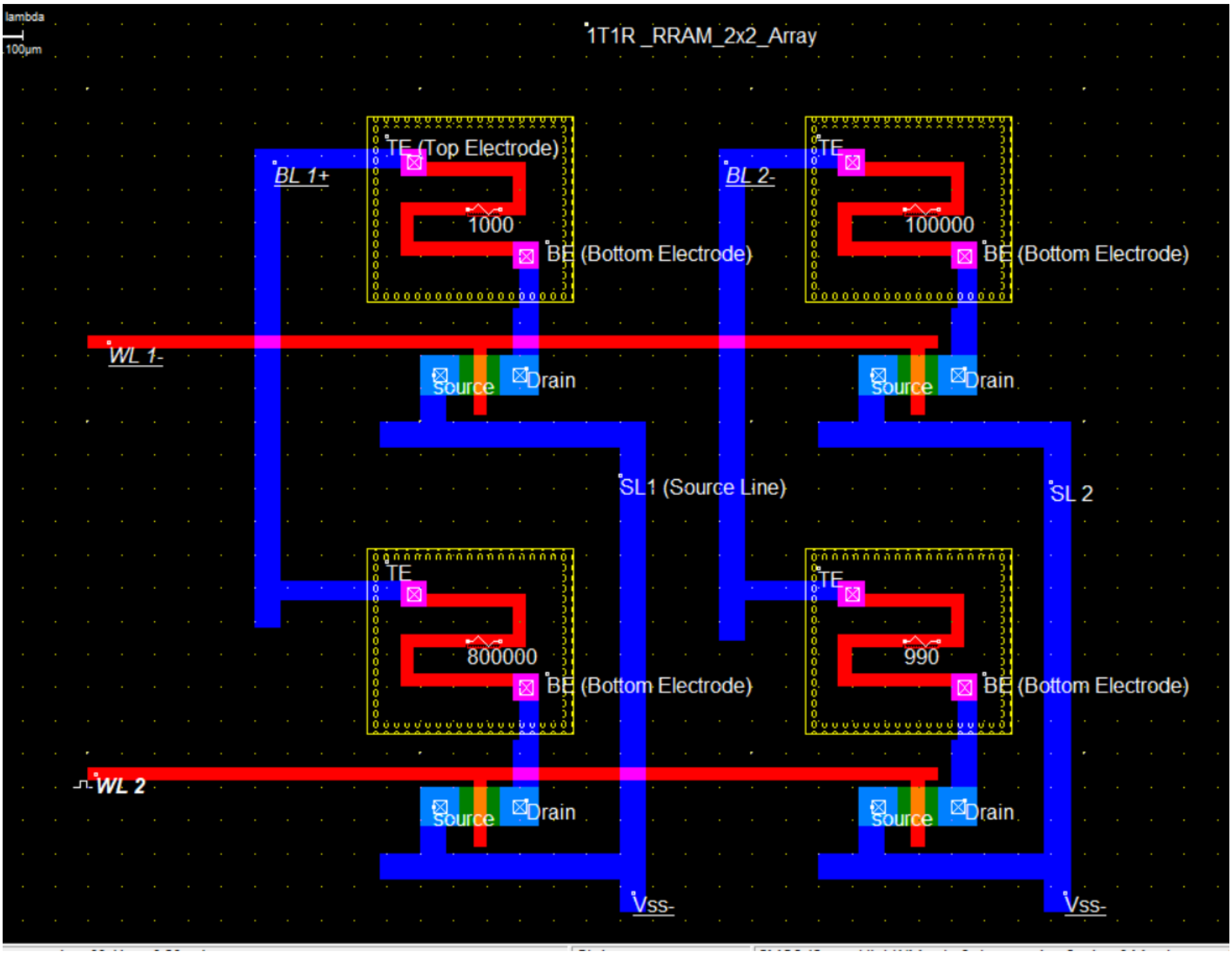
| | |
|---|---|
| **LRS** | **HRS** |
| **HRS** | **LRS** |

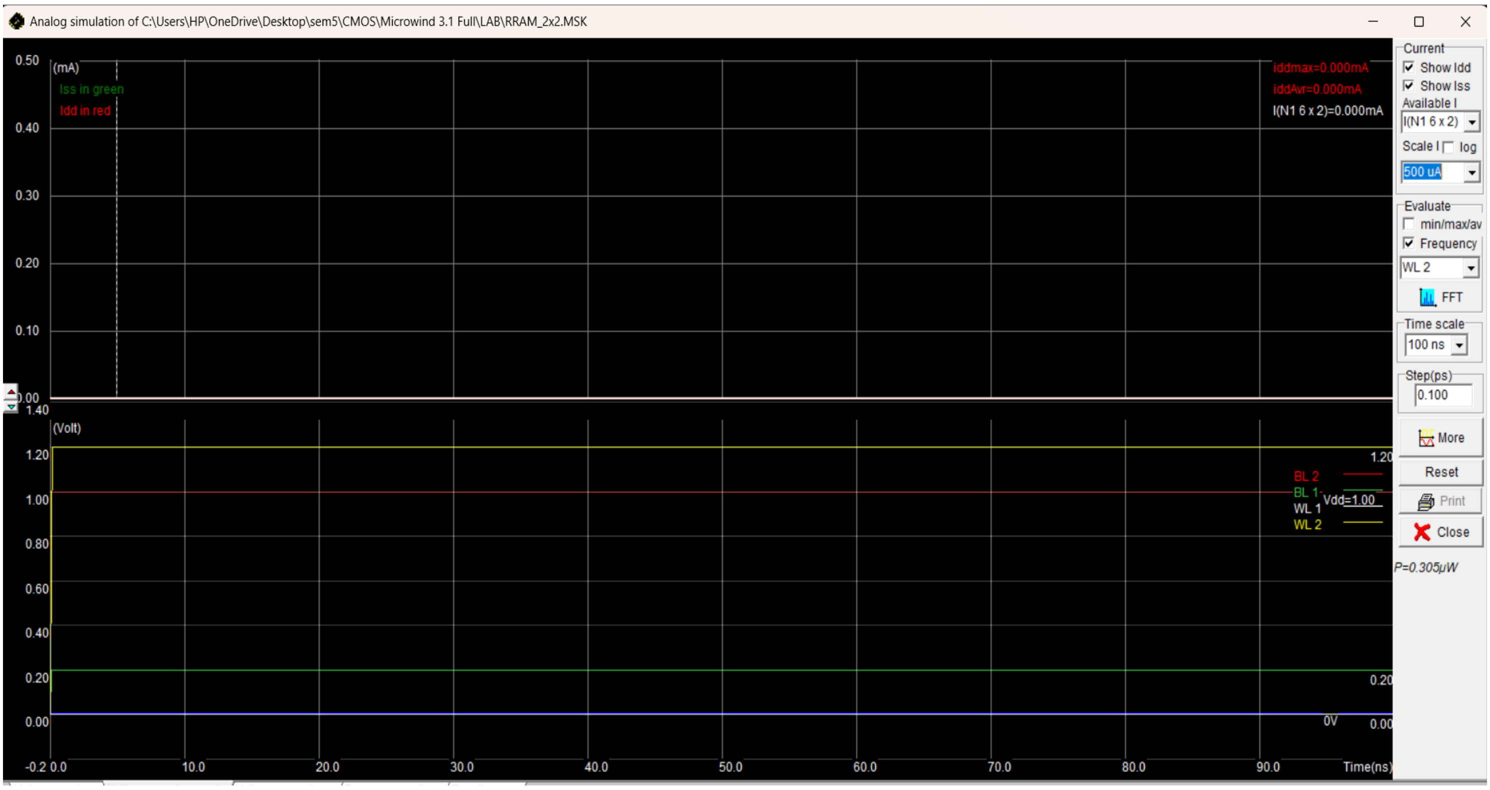


**Output waveform :**

Reading RRAM cell at bottom left :

**calculated as 0.25 micro Ampere ~ 0.00025milli Ampere





Analog simulation of C:\Users\HP\OneDrive\Desktop\sem5\CMOS\Microwind 3.1 Full\LAB\RRAM_2x2.MSK

Iddmax=0.000mA

IddAvr=0.000mA

I(N1 6 x 2)=0.000mA

☑ Show Idd

☑ Show Iss

Available I: I(N1 6 x 2)

Scale I □ log: 500 uA

□ min/max/av

☑ Frequency: WL 2

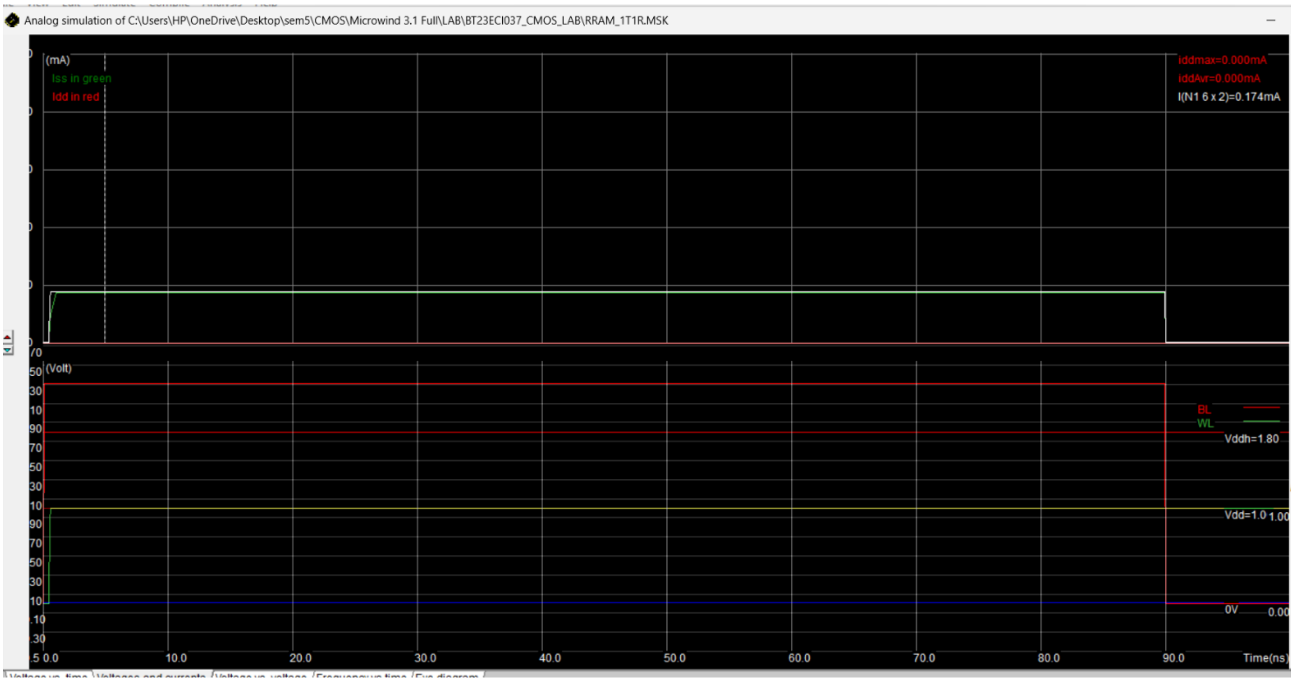
Time scale: 100 ns

Step(ps): 0.100

P=0.305µW

Fig.14: Read ‘0’ HRS **(* current in graph is zero because scale has limit upto mA but power = 0.305 micro Watt)**

Reading RRAM cell at bottom right :



Analog simulation of C:\Users\HP\OneDrive\Desktop\sem5\CMOS\Microwind 3.1 Full\LAB\BT23ECI037_CMOS_LAB\RRAM_1T1R.MSK

Iddmax=0.000mA

IddAvr=0.000mA

I(N1 6 x 2)=0.174mA

**Fig.15: Read ‘1’ LRS**