A Novel Transpose 2T-DRAM based Computing-in Memory Architecture for On-chip DNN Training and Inference

Aditya Patil - 22110188
Mohit Maurya - 22110145

https://doi.org/10.1109/AICAS57966.2023.10168641

Index

- Introduction
- Algorithm analysis of FP and BP in DNN learning
- Proposed 2T-DRAM Design
- Proposed mapping method of CONV layers
- Proposed mapping method of FC layers
- Results & Conclusion

Introduction

Problems:

- Continuously increasing demand of DNN's have lead to increase in higher computations.
- Traditional von Neumann architectures suffer from high latency and power consumption due to data movement between memory and computation units.

CIM can be an effective approach. But issue with existing CIM architectures are

1) SRAM CIM Architecture:
   - Hardware overhead
   - Limit scalability due to large cell area
1) DRAM CIM Architecture:
   - Lack support for Back Propagation (BP) needed for on-chip DNN training.

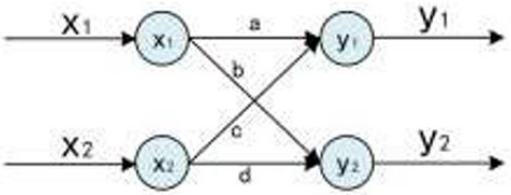
To solve this issues, this paper proposes a transpose 2T-DRAM based CIM architecture which can efficiently implement on-chip learning while keeping both memory array and peripheral circuit minimally changed.

Key Contributions :

1. Transpose 2T-DRAM Memory:
    - Enables efficient BP in-memory realization without extra circuit overhead.
    - Achieves higher weight reuse and eliminates memory updates, saving energy during transpose operations.
1. Digital-Domain CIM Arithmetic:
    - Utilizes digital logic for high accuracy and robustness during training and inference.
1. Weight Mapping for FC & CONV Layers:
    - Novel mapping techniques to accelerate BP with weight reuse and parallel computation for improved performance.

Algorithm analysis of FP and BP in DNN learning

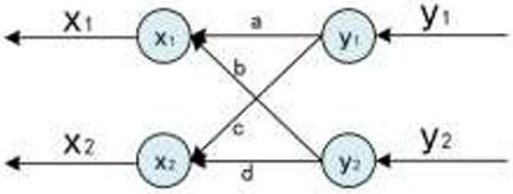
Fully connected layers :



Forward propagation

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} a & c \\ b & d \end{bmatrix}}_{W} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} W_{1,1} & \cdots & W_{i,1} \\ \vdots & \ddots & \vdots \\ W_{1,j} & \cdots & W_{i,j} \end{bmatrix} \times \begin{bmatrix} X_1 \\ \vdots \\ X_i \end{bmatrix} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_j \end{bmatrix}$$



Back propagation

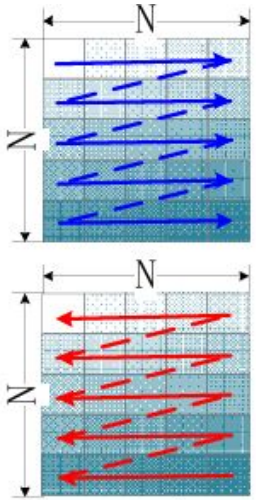
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{W^T} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\begin{bmatrix} W_{0,0} & \cdots & W_{1,j} \\ \vdots & \ddots & \vdots \\ W_{i,1} & \cdots & W_{i,j} \end{bmatrix} \times \begin{bmatrix} \Delta Y_1 \\ \vdots \\ \Delta Y_j \end{bmatrix} = \begin{bmatrix} \Delta X_1 \\ \vdots \\ \Delta X_i \end{bmatrix}$$

The weight matrix of FC layers needs to be **Transposed** during the BP computation.

Algorithm analysis of FP and BP in DNN learning

Convolutional layers：



FP CONV equation:

$$\begin{bmatrix} X_{1,k} & \cdots & X_{1,k} \\ \vdots & \ddots & \vdots \\ X_{k,1} & \cdots & X_{k,k} \end{bmatrix} conv \begin{bmatrix} W_{1,1} & \cdots & W_{1,n} \\ \vdots & \ddots & \vdots \\ W_{n,1} & \cdots & W_{n,n} \end{bmatrix} = \begin{bmatrix} Y_{1,1} & \cdots & Y_{1,k-n+1} \\ \vdots & \ddots & \vdots \\ Y_{k-n+1,1} & \cdots & Y_{k-n+1,k-n+1} \end{bmatrix}$$

BP CONV equation:

$$\begin{bmatrix} 0 & \cdots & & \cdots & & \cdots & 0 \\ \cdots & & & & & & \\ & & \Delta Y_{1,1} & \cdots & \Delta Y_{1,k-n+1} & & \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ & & \Delta Y_{k-n+1,1} & \cdots & \Delta Y_{k-n+1,k-n+1} & & \\ \cdots & & & & & & \\ 0 & & & \cdots & & & 0 \end{bmatrix} conv \begin{bmatrix} W_{n,n} & \cdots & W_{n,1} \\ \vdots & \ddots & \vdots \\ W_{1,n} & \cdots & W_{1,1} \end{bmatrix} = \begin{bmatrix} \Delta X_{1,1} & \cdots & \Delta X_{1,k} \\ \vdots & \ddots & \vdots \\ \Delta X_{k,1} & \cdots & \Delta X_{k,k} \end{bmatrix}$$

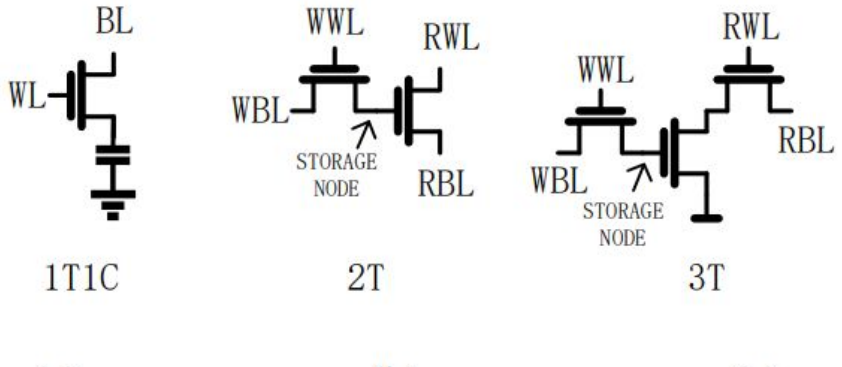
Proposed 2T-DRAM Design

Functional Symmetry:

- Interchangeable Read Word Line (RWL) and Read Bit Line (RBL) enable non-destructive and transposable reads.

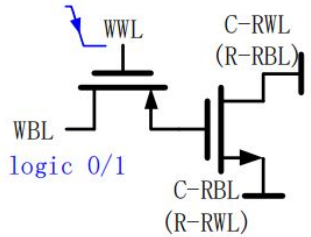
Transpose Operation:

- Only using peripheral selector switches controlled by transpose enabling signal.

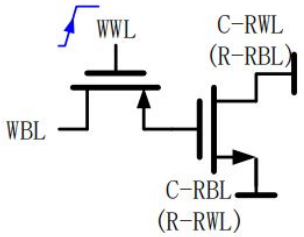


There are 4 modes of operation :

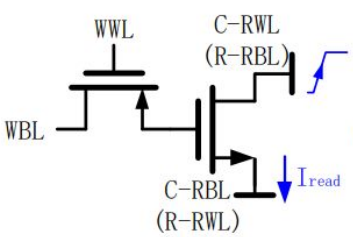
Write Mode



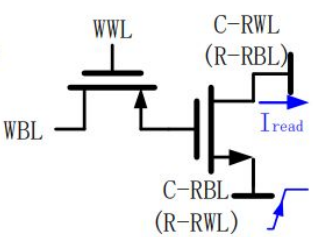
Standby Mode



Normal Read

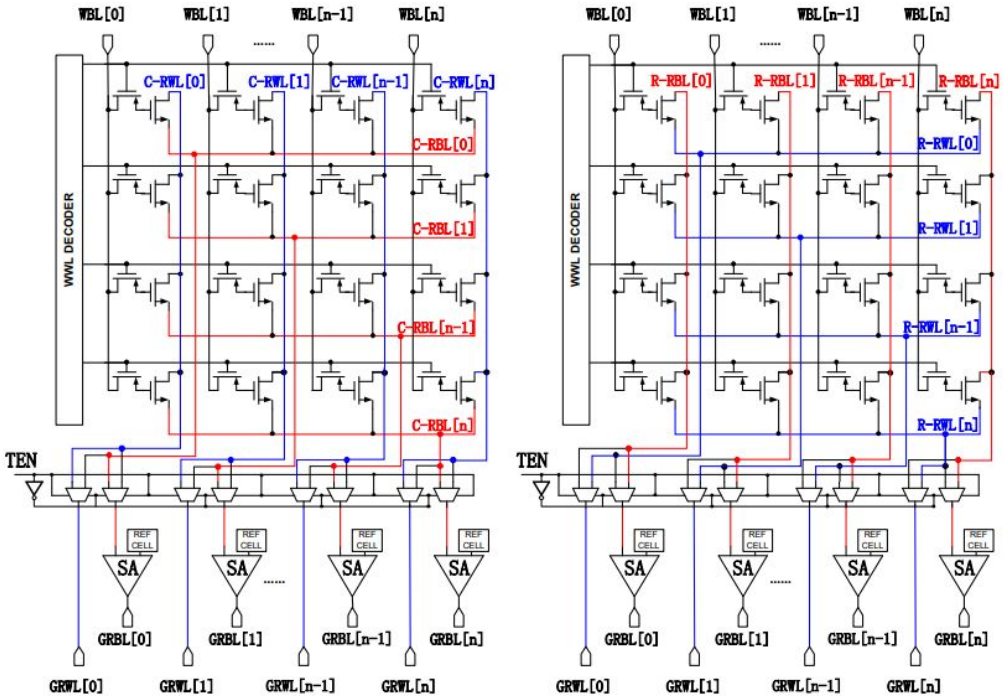


Transpose read



2T-DRAM can perform bitwise AND operation (multiplication)
of stored data and input (fed from C-RWL , R-RWL)

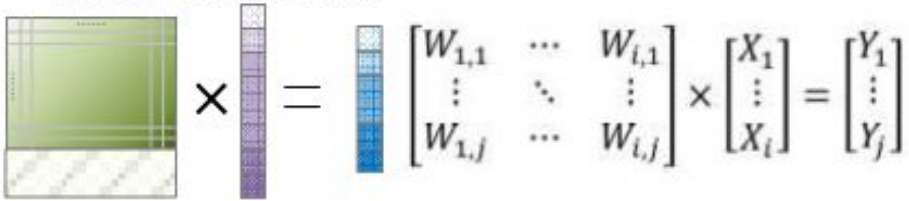
Proposed mapping method of FC layers



a) Normal read b) Transpose read

The INPUT signal is applied to **global RWL (GRWL)** with **Blue color**, and the bitwise multiplication result is read through the sense amplifier (SA) connected to **global RBL (GRBL)** with **Red color**

FP FC equation:



$$\begin{bmatrix} W_{1,1} & \cdots & W_{i,1} \\ \vdots & \ddots & \vdots \\ W_{1,j} & \cdots & W_{i,j} \end{bmatrix} \times \begin{bmatrix} X_1 \\ \vdots \\ X_i \end{bmatrix} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_j \end{bmatrix}$$

So let say

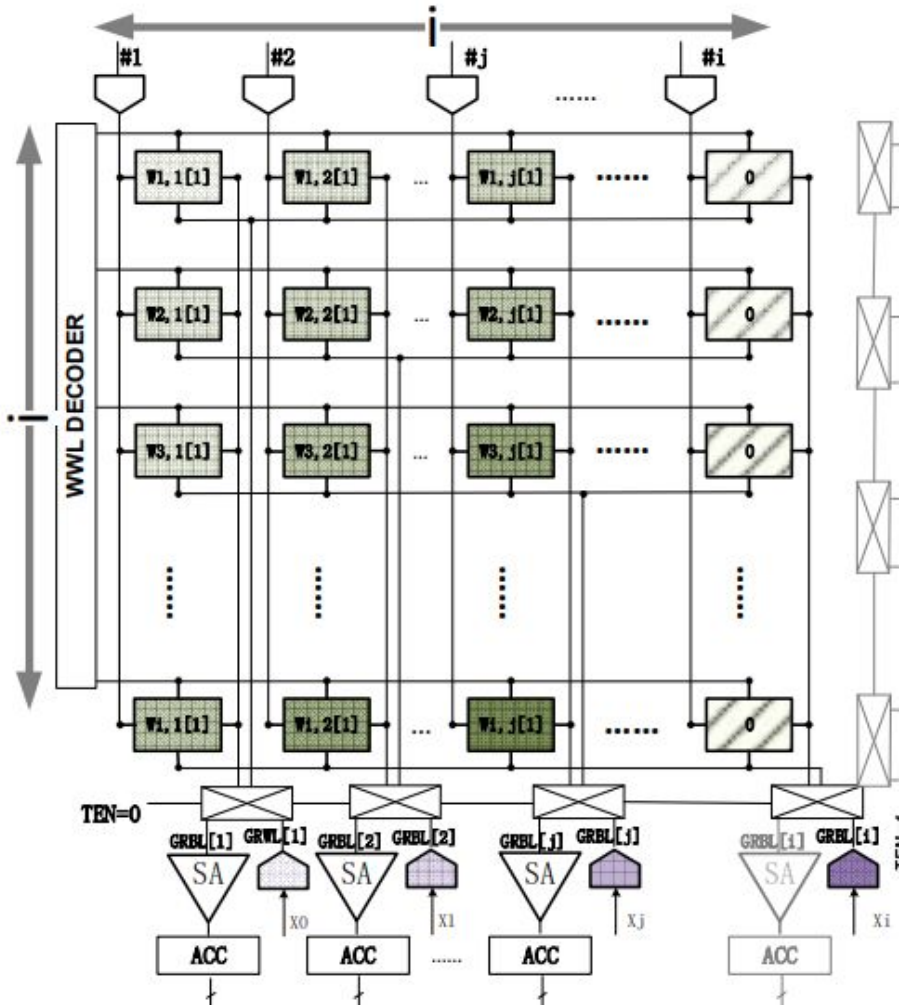
$Y_1 = W_{1,1}X_1 + W_{2,1}X_2 + \ldots\ldots W_{i,1}X_i$

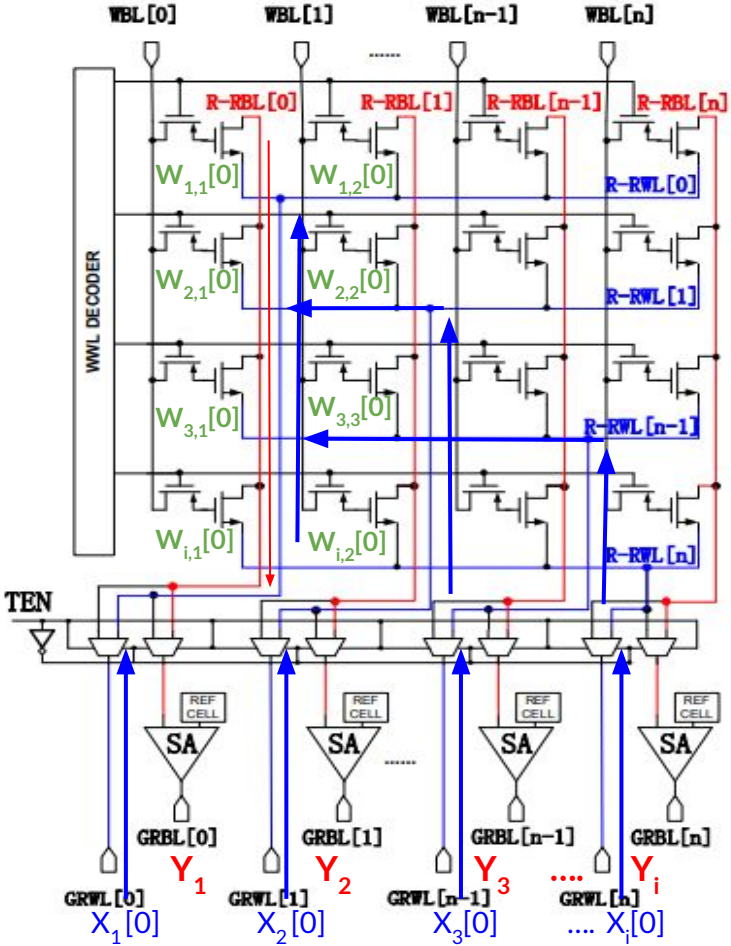
Similarly for Y_2, Y_3 …

Weights are stored in **Bit-slice fashion.**

And the weights are stored in this order in memory. (Row in formula is column in the memory)

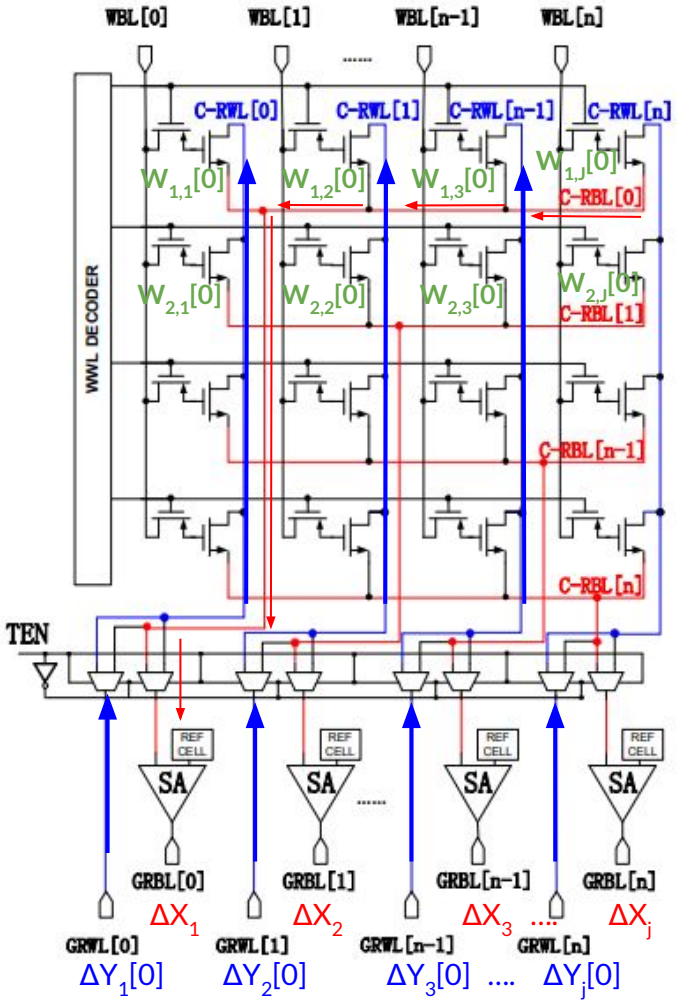
$$\begin{matrix} W_{1,1} & \cdots & W_{1,j} \\ \vdots & & \vdots \\ W_{i,1} & \cdots & W_{i,j} \end{matrix}$$



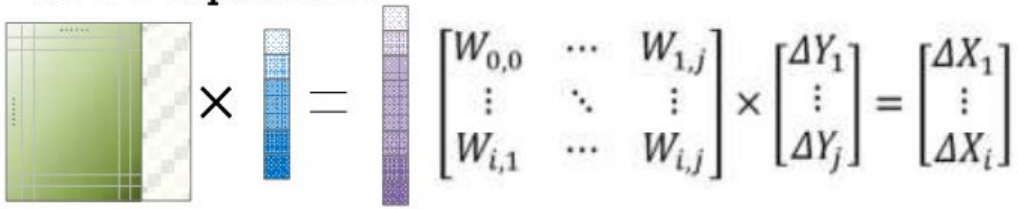


$Y_1 = W_{1,1}X_1 + W_{2,1}X_2 + \ldots\ldots W_{i,1}X_i$

$Y_2 = W_{1,2}X_1 + W_{2,2}X_2 + \ldots\ldots W_{i,2}X_i$



BP FC equation:

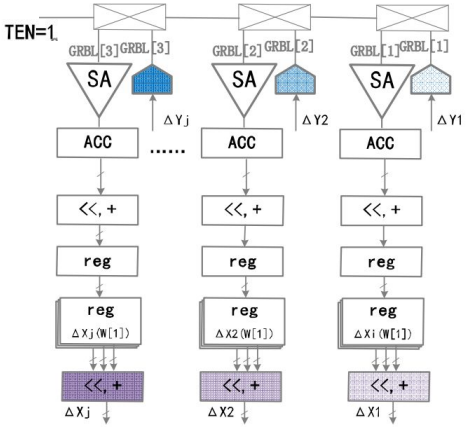


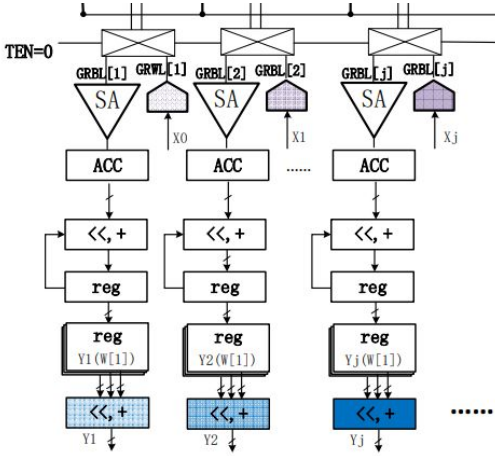
So let say

$\Delta X_1 = W_{1,1}\Delta Y_1 + W_{1,2}\Delta Y_2 + \ldots\ldots\ W_{1,j}\Delta Y_j$

Similarly for ΔX_2 , ΔX_3 …

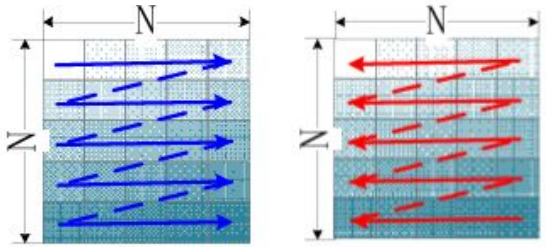
Peripheral Circuit looks like this …





$\Delta X_1 = W_{1,1}\Delta Y_1 + W_{1,2}\Delta Y_2 + \ldots\ldots\ W_{1,j}\Delta Y_j$ $\Delta X_2 = W_{2,1}\Delta Y_1 + W_{2,2}\Delta Y_2 + \ldots\ldots\ W_{2,j}\Delta Y_j$

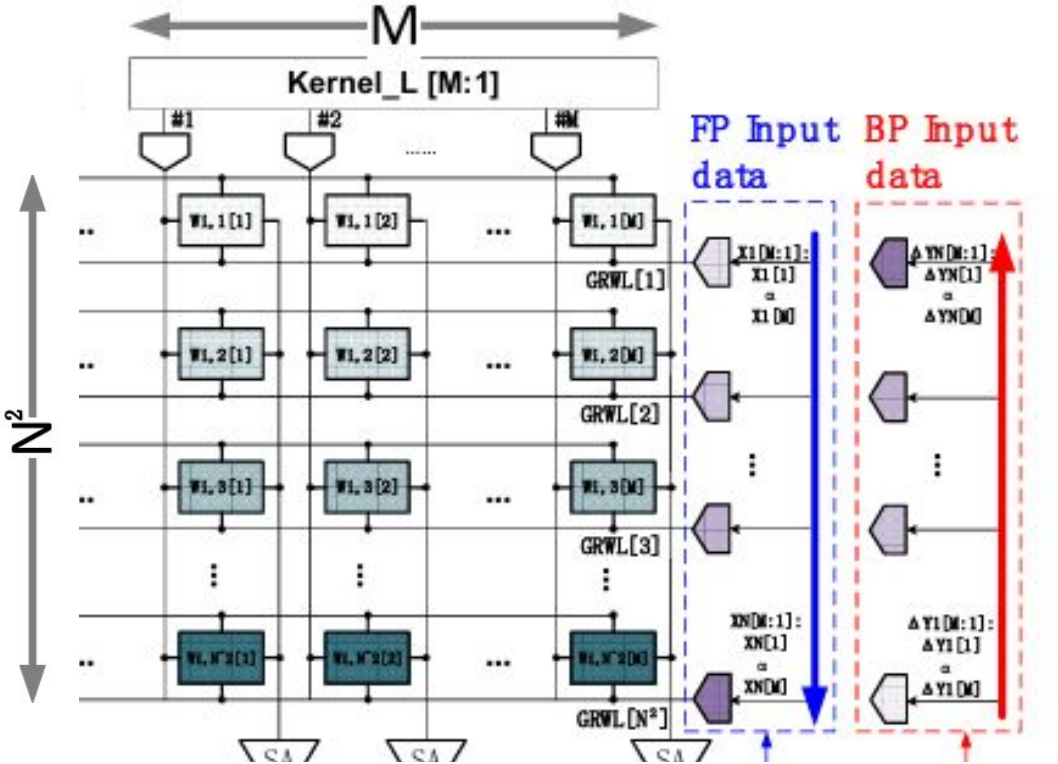
Proposed mapping method of CONV layers



$X_1W_1 + X_2W_2 + ... = Y_1$

	110	W_1		001	W_2
*	001	X_1	*	100	X_2
	110			000	
+	000x		+	000x	
+	000xx		+	001xx	
	00110			00100	

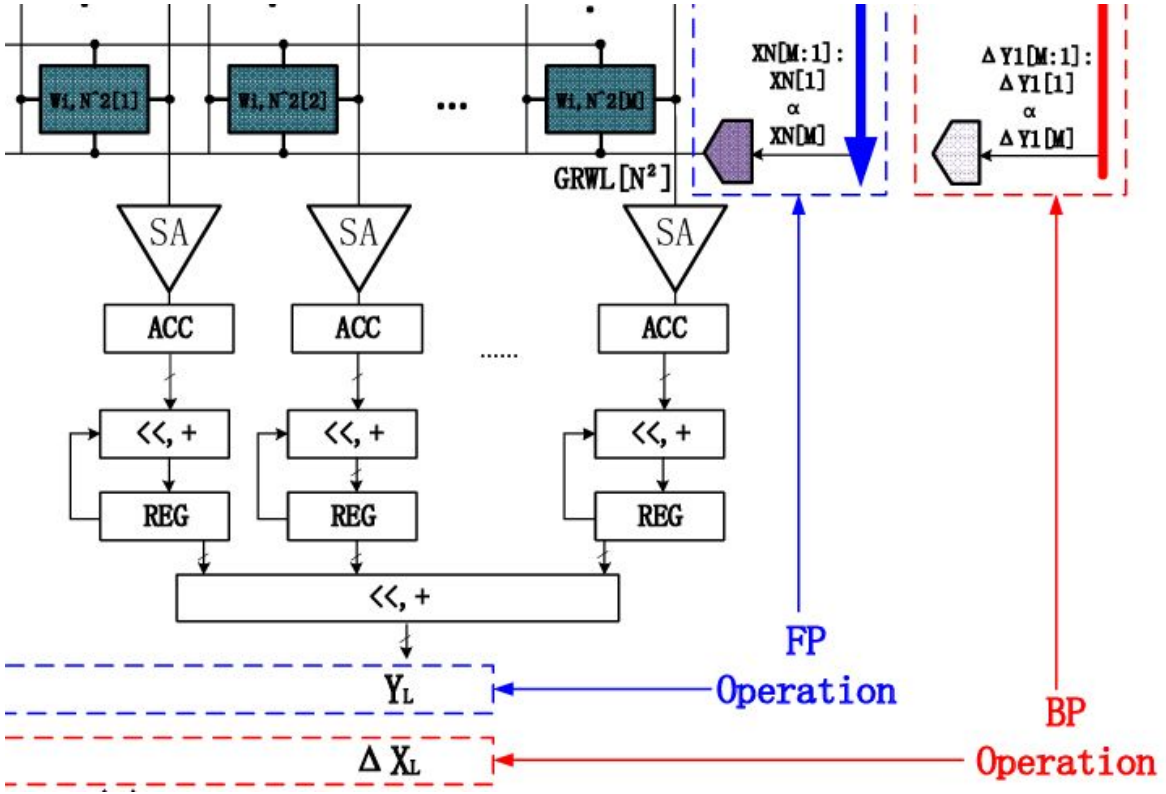
00110 + 00100 = 01010



Proposed mapping method of CONV layers

The CONV kernels of different
channels can be mapped to different memory groups to provide parallel computing.

The input data is changed from the first order
of the last row to the first row by the input address decoder.



Results & Conclusion

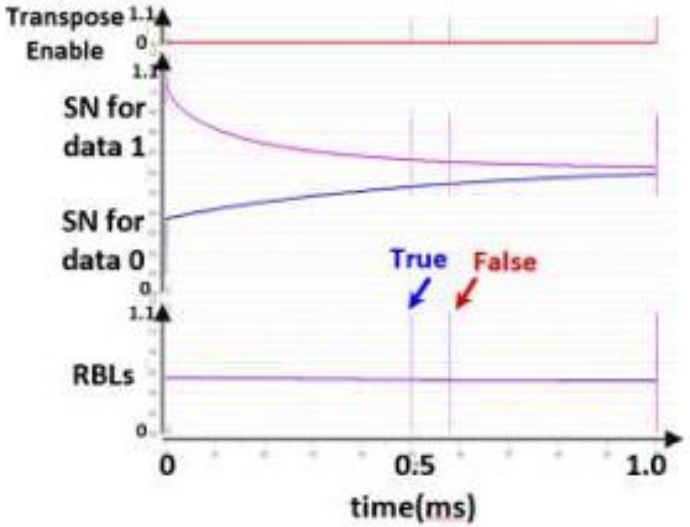
4×4 transpose 2T-DRAM array designed in 40nm CMOS technology using Cadence Virtuoso.

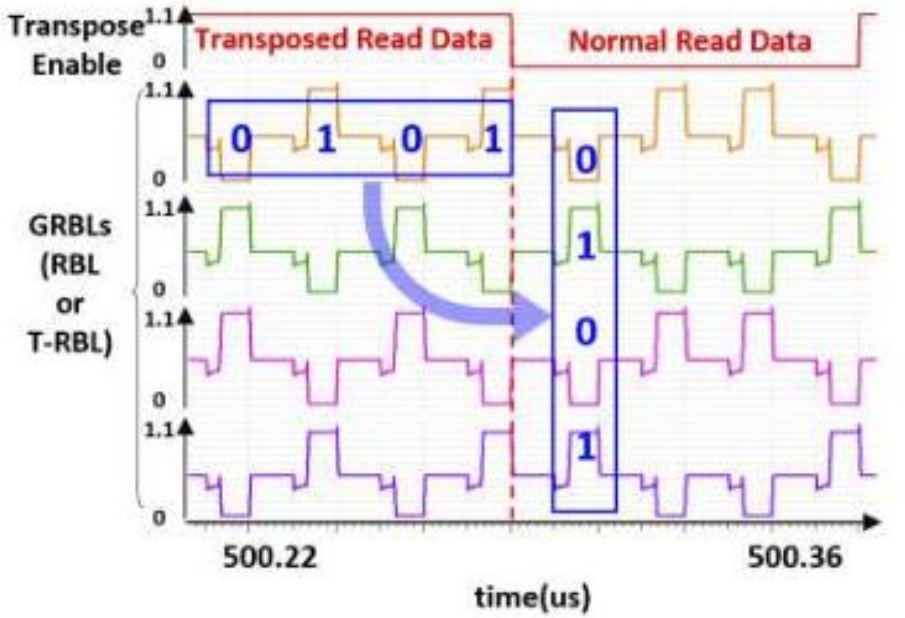
Results:

- Retention Time: 500 µs.
- Transpose Function: Correctly executed within 500 µs.
- At 100 MHz, 25,000 1-bit multiplications and addition operations completed within retention time.

Within the retention time, the macro can complete one FP and one BP operation of a typical layer (e.g., 256×10 FC layer), which enables a refresh-less training process for light weight DNN networks.

Results & Conclusion





Results & Conclusion

4×4 transpose 2T-DRAM expanded to 16K array

For a typical 256×10 FC layer in a 4CONV+3FC network, only **5.12% of the retention time** is used for 1-bit readout and calculations in FP.

The proposed transpose circuitry avoids interruptions during BP, eliminating refresh requirements for weight matrix.

Results & Conclusion

To validate the efficiency, used the Top-1 **CIFAR-10** classification accuracy.
8-bit NN with 4 CONV and 3 FC

Achieved **82.15%** accuracy

Due to high parallelism in the mapping, the proposed 2T DRAM method can achieve **2.4× GOPS @100 MHz**

Results & Conclusion

Table compares the computing power and CNN on-chip training accuracy between the proposed 2T-DRAM array and prior CIM works based on Python simulation.

CIM Solution	6T-SRAM [3]	7T-SRAM [2]	3T-DRAM [1]	1T1C-DRAM [4]	Proposed 2T-DRAM[a]
CIM domain	Digital	Digital	Analog	Analog	Digital
Array Size	80Kb	64Kb	8Kb	16Kb	16Kb
BP Support	Yes	Yes	No	No	Yes
Input Precision	8 bits	16 bits	4 bits	8 bits	8 bits
Weight Precision	8 bits	8 bits	4 bits	8 bits	8 bits
Params.	NA	NA	NA	2.168 M	2.296 M
Model	ResNet-20	4CONV+1FC	VGG16	4CONV+2FC	4CONV+3FC
Dataset	CIFAR-10	LFW	CIFAR-10	CIFAR-10	CIFAR-10
Accuracy	91.94 %	83.9 %	90.6 %	80.1 %	82.15%
GOPS	NA	NA	NA	4.71	11.26

a. Estimation based on Python, Top-1 accuracy.

Reference

Citation:

Zhao, Y., Shen, Z., Xu, J., Chai, K. C. T., Wu, Y., & Wang, C. (2023). A novel transpose 2T-DRAM based computing-in-memory architecture for on-chip DNN training and inference. 2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS), 1–4. https://doi.org/10.1109/aicas57966.2023.10168641

Project flow

- Testing of 2T-DRAM cell (with different sizing)
- Problem and its solutions
- Implementation of 2T-DRAM array (4x4)
- Results and it failure.
- Our proposed solutions

Testing 2T-DRAM cell

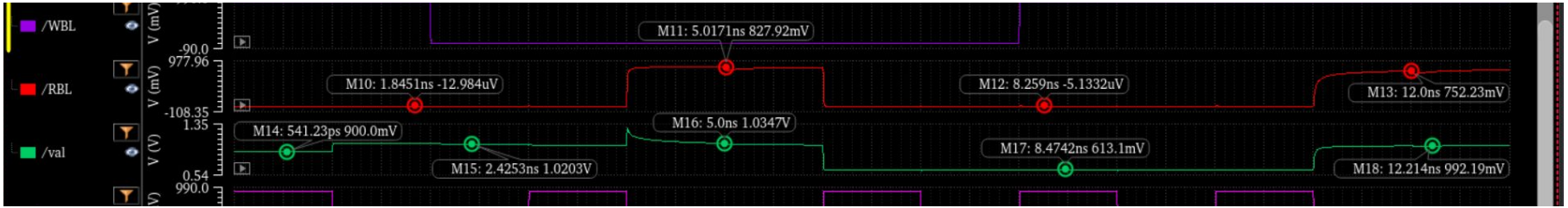
We tried 3 different sizing options

- **Wn = 100n, Wp = 100n:** This is the default case with minimum area. The RWL and WWL leaks into the stored value, thus destructing it slightly
- **Wn = 100n, Wp = 250n:** This size has around the same capacitance as 3T-tg cell. So we tested it to compare with the 3T cell. Increasing the pmos size increases the coupling of data with WWL, which is 1. So, this causes destruction of a stored 0
- **Wn = 180n, Wp = 100n:** We reduced the relative coupling due to pmos to reduce the destruction caused by WWL. But this increases the coupling with RWL

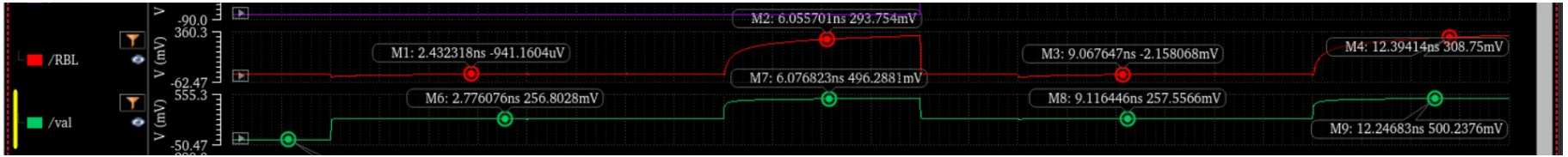
So, overall, the best sizing was Wn = 100n, Wp = 100n. This also has the minimum area.

Wn = 100n, Wp = 100n

1 stored

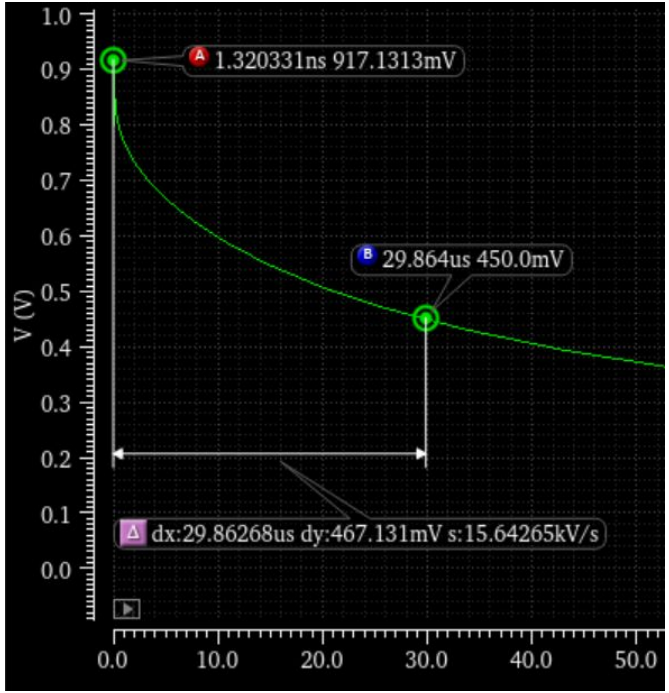


0 stored

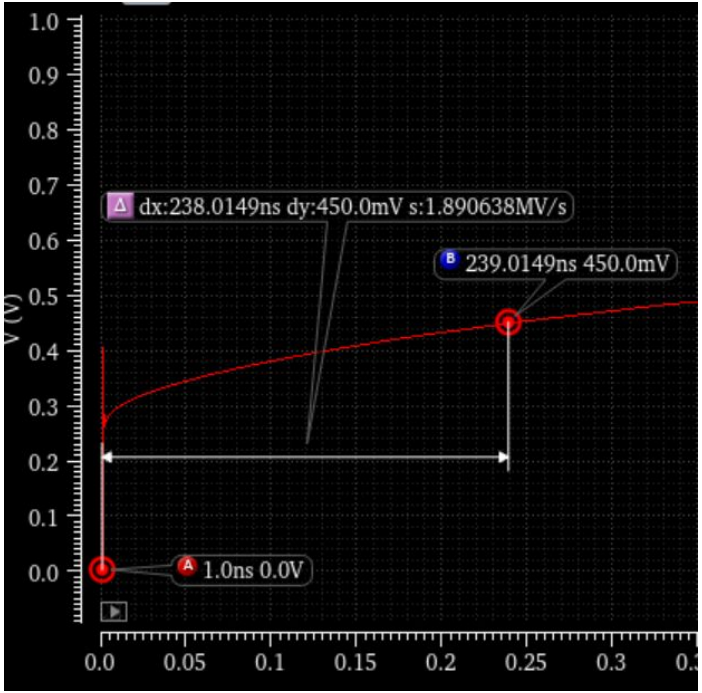


Wn = 100n, Wp = 100n

- **1 stored:** The value is pulled up when WWL and RWL switches to a 1. The value then drops more than its initial value when RWL is again switched to a 0.
  - Max Value: 1.1V
  - Min value: 610mV
- **0 stored:** The value is pulled up when WWL and RWL switches to a 1. RBL is also pulled up when RWL switches to a 1.
  - Max Value: 500mV
  - Max RBL: 310mV



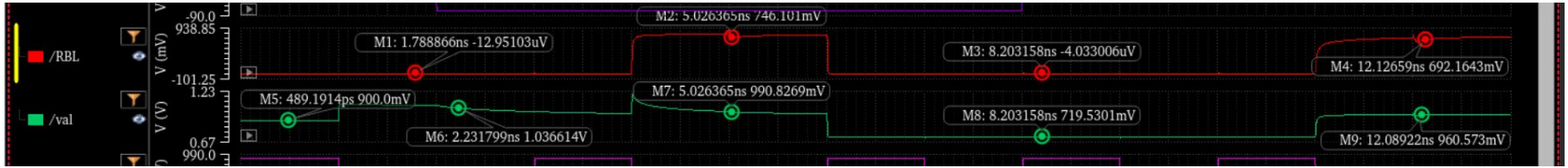
Retention time for stored 1 = 29.86 uS



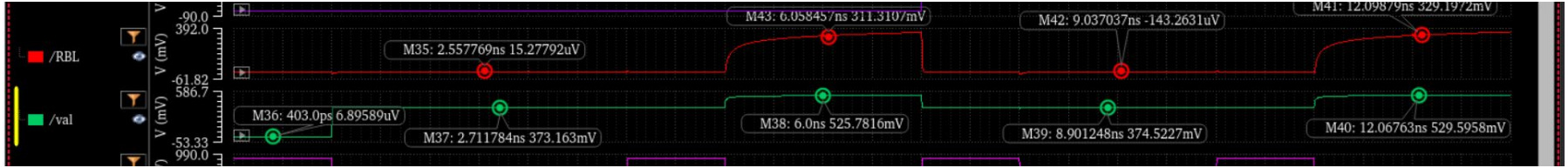
Retention time for stored 0 = 239 nS

Wn = 100n, Wp = 250n

1 stored



0 stored



Wn = 100n, Wp = 250n

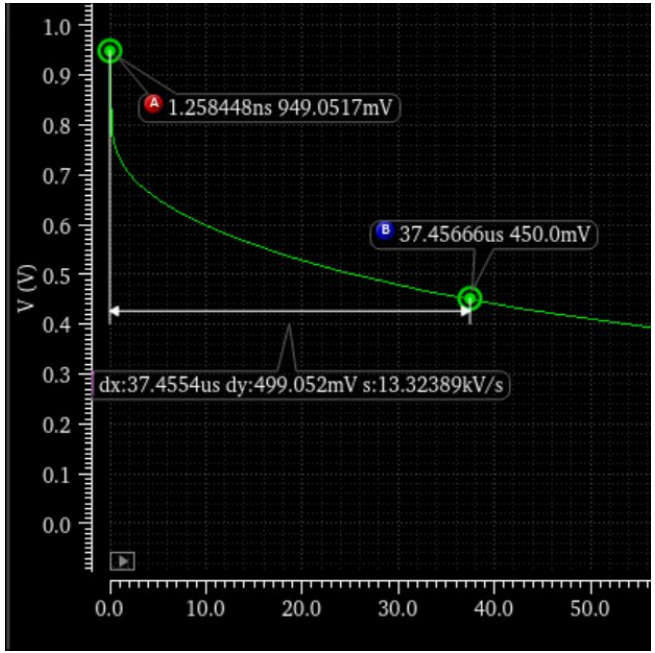
1 stored:

- Max Value: 1V
- Min value: 720mV
- Min RBL: 750mV (when logic 1 is expected)

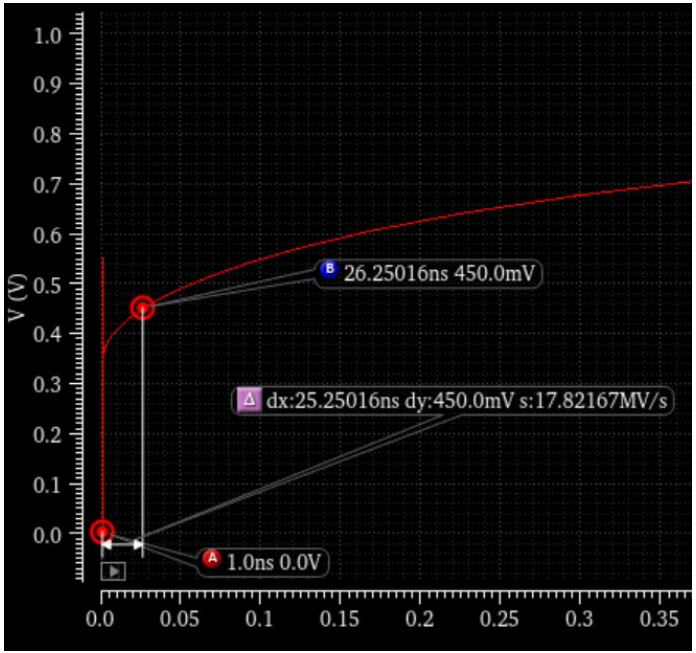
0 stored:

- Max Value: 530mV
- Max RBL: 330mV

  (These values are higher than the previous case)



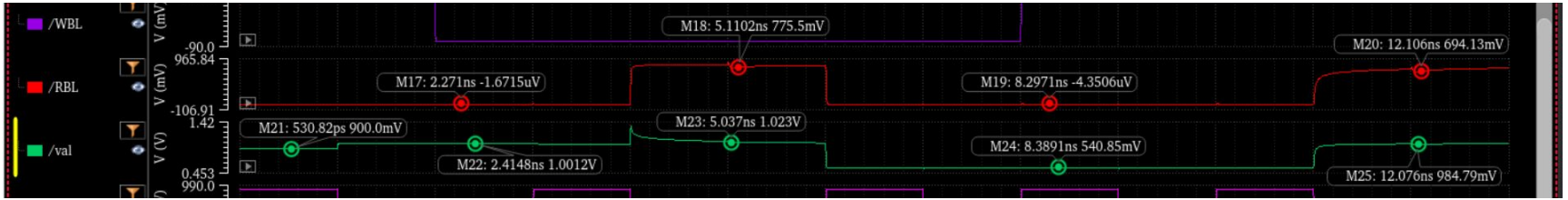
Retention time for stored 1 = 37.45 uS



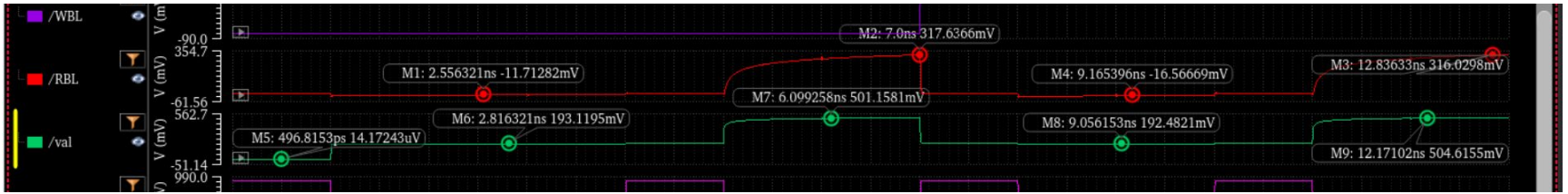
Retention time for stored 0 = 25 nS

Wn = 180n, Wp = 100n

1 stored



0 stored



Wn = 180n, Wp = 100n

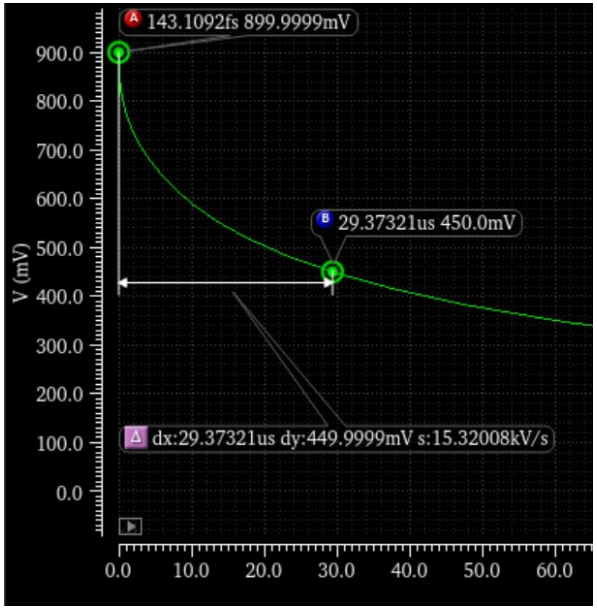
1 stored:

- Max Value: 1V
- Min value: 540mV
- Min RBL: 775mV (when logic 1 is expected)

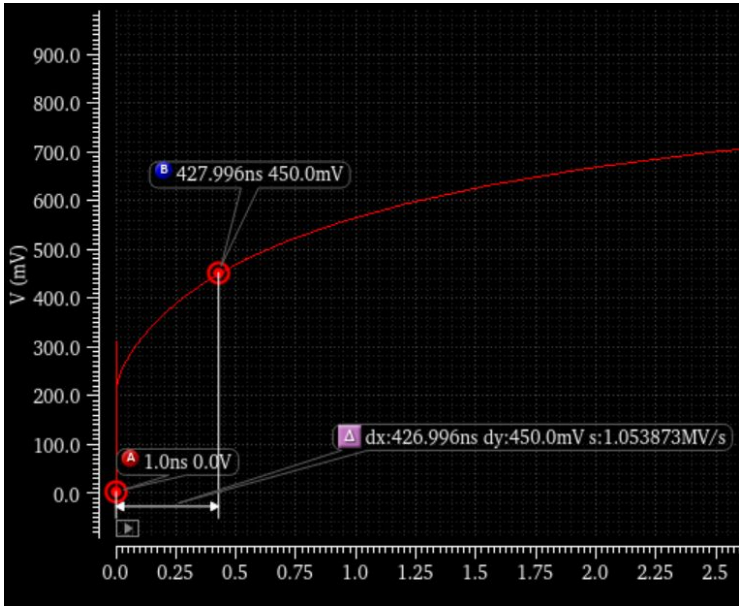
0 stored:

- Max Value: 500mV
- Max RBL: 320mV

  (Max and min values are very close, so we have a very less noise margin)



Retention time for stored 1 = 29.37 uS



Retention time for stored 0 = 427 nS

Problem faced in the 2T cell

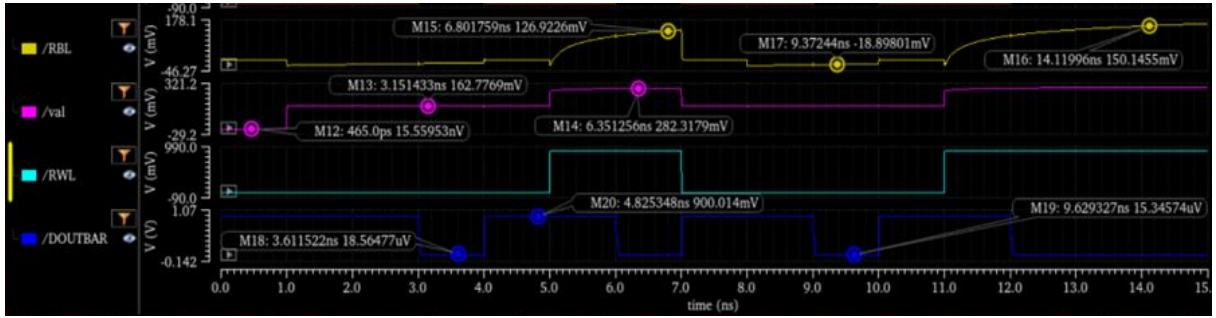
Problems :

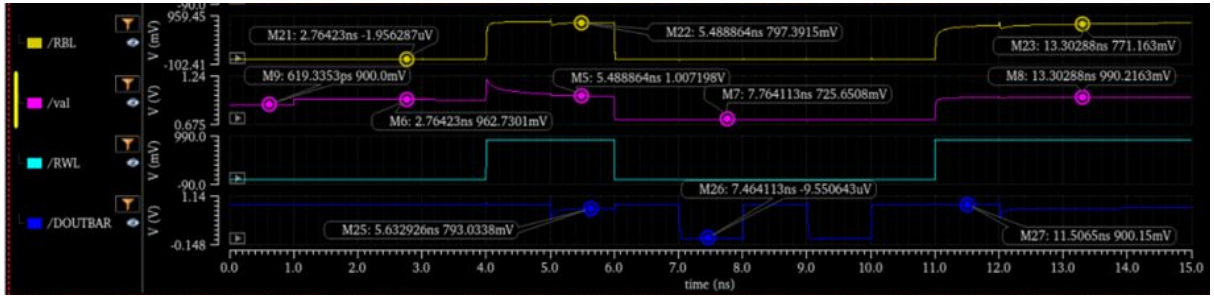
- Coupling between stored data and other signals (WWL, WBL, RWL)

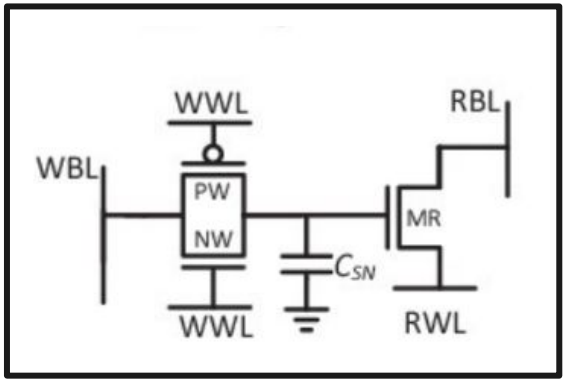
Solutions :

1. 3T-TG cell : It can store a strong 0 with the tradeoff of area

3T-TG cell





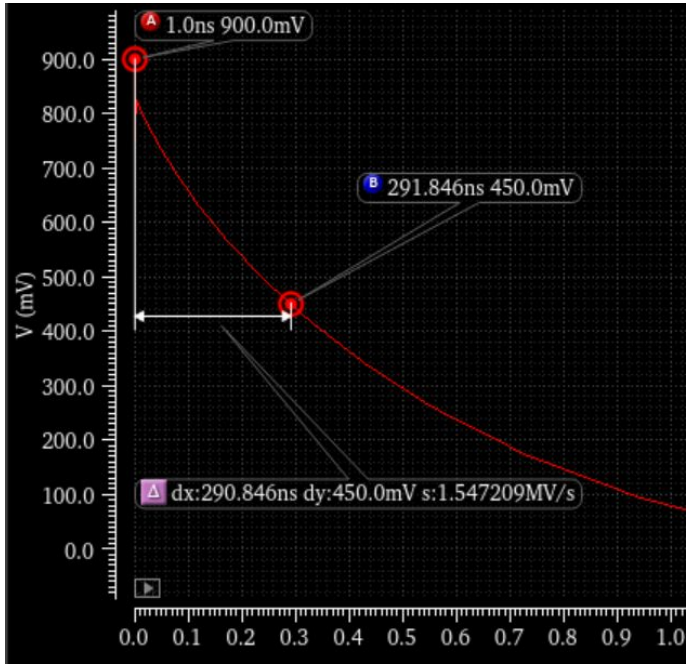


Energy for 0 (stored): 3.44nJ

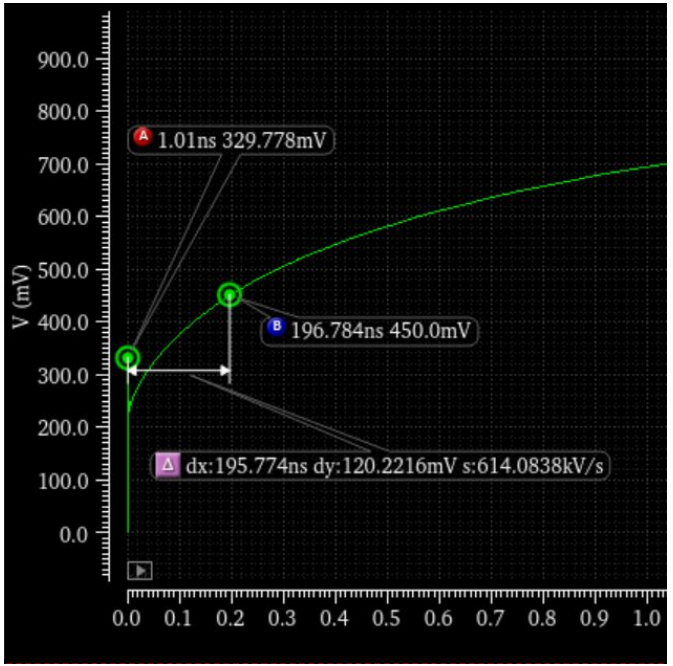
Energy for 1 (stored) : 59.77 pJ.

Static power (0 stored) = 11.49nW.

Static power (1 stored) = 1.32nW.



Retention time for stored 1 = 291 nS



Retention time for stored 0 = 196 nS

Array Implementation

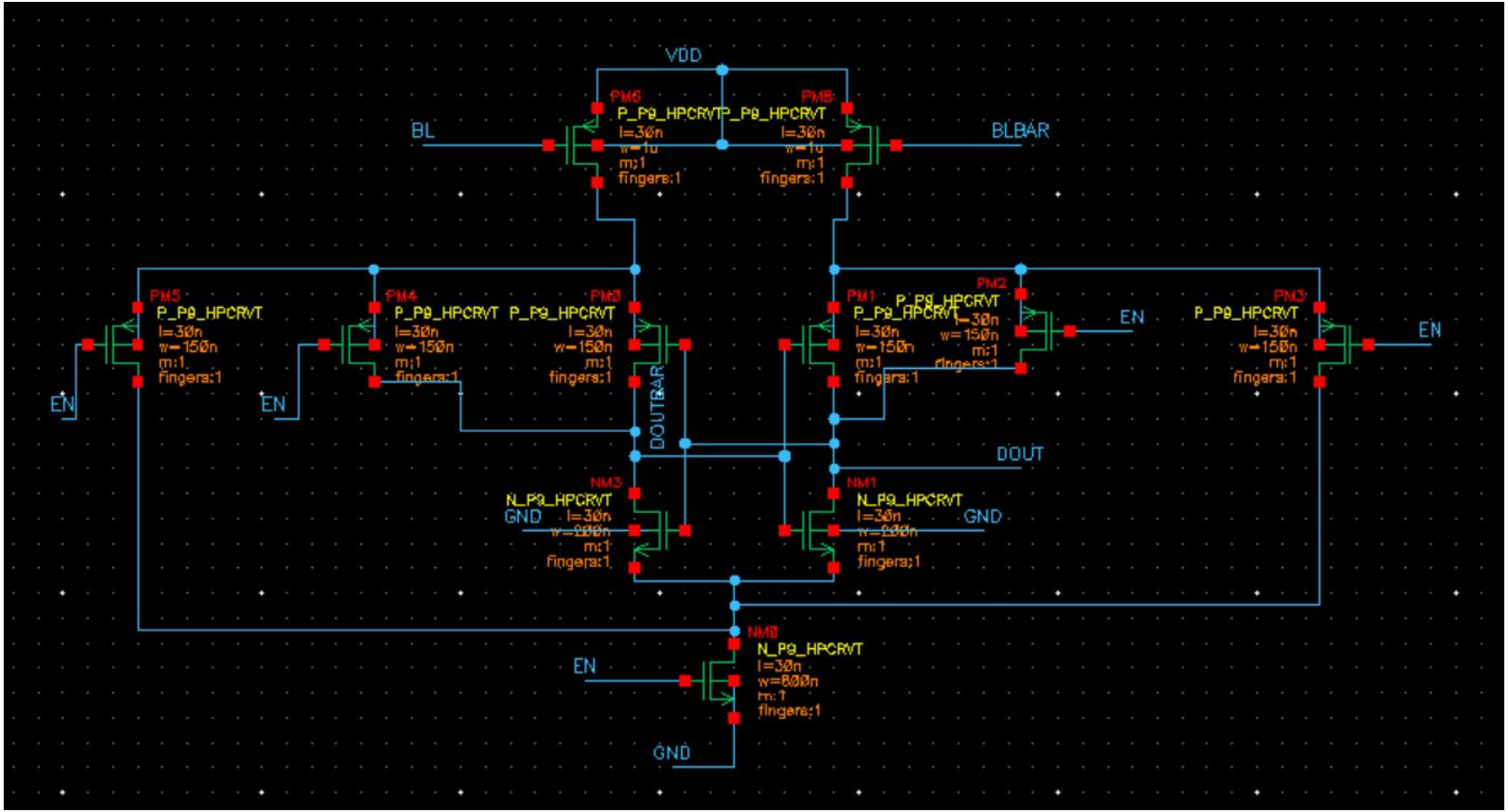
This aboves circuit is for Normal read mode

Precharge: It charges the circuit RBL to 0V. We precharge to a 0 as the 2T cell passes weak 0s to the RBLs.

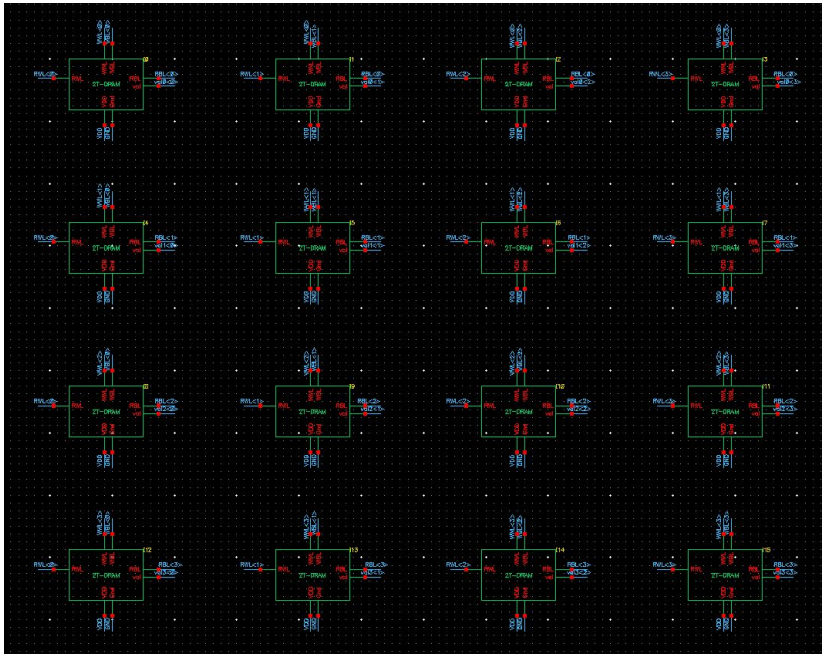
Sense Amplifier : It is a Single ended SA whose input is Bit line. It compare the bit line with the reference signal given of voltage 350mV and provide the output. As out BL is precharged to 0 thus we made our own SA which detects charging of BL.

(**Note**: The schematics shown is just to ease the analysis but both RWL and RBL are interchangeable so it should be connected to SA and Precharge circuit via MUXs.)

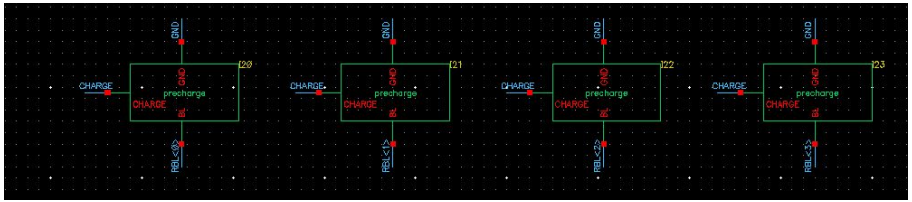
Sense Amplifier



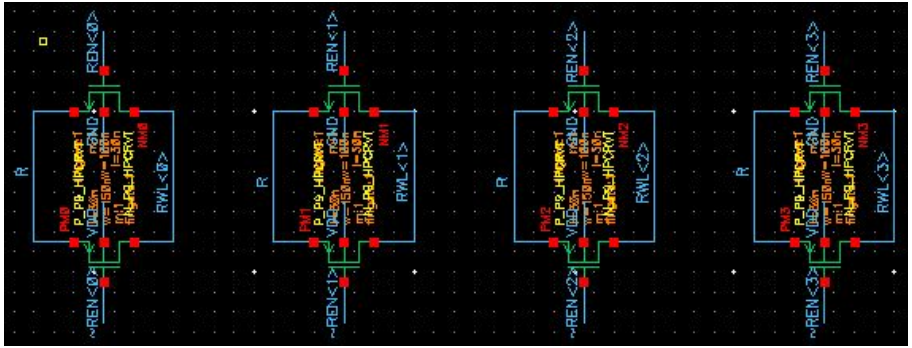
2T-DRAM 4x4 array



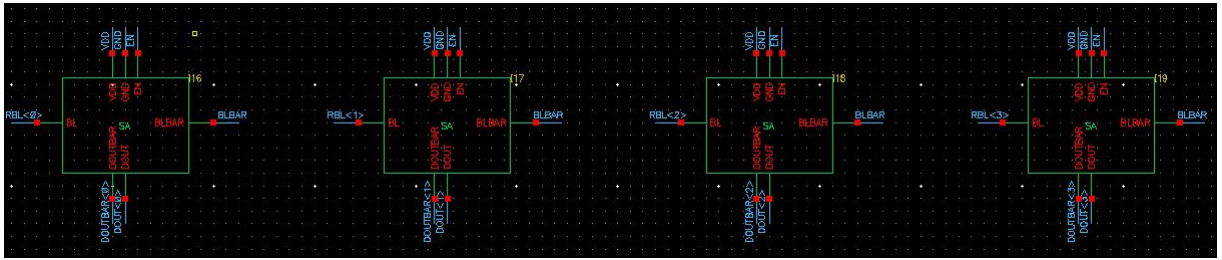
Precharge



Transmission gate

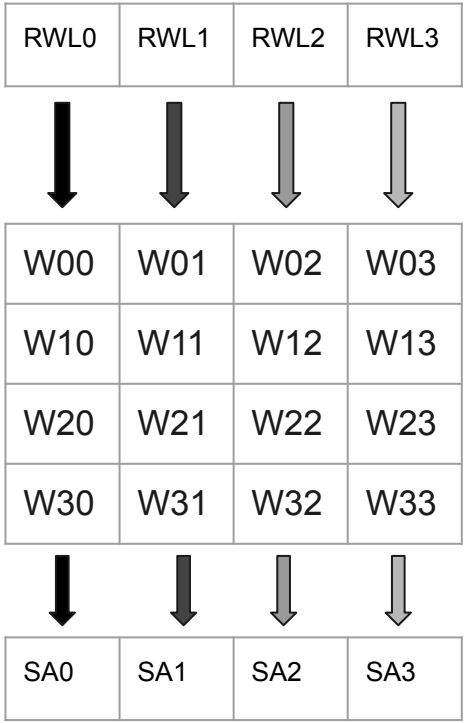


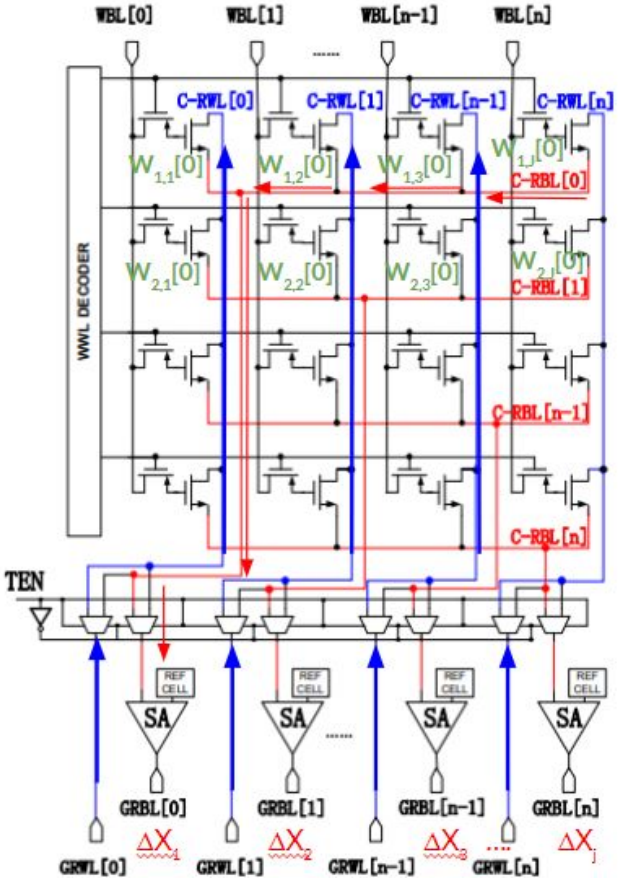
Sense Amplifier



Results

Input is given from RWLs in Bit serial fashion and its get multiplied and output is available on SAs.





Let say input from RWL<0:3> is **R<0:3>**

Then Output should be like this. (For each row SA's provided output when only that particular RWL is provided to weight matrix while other are in high impedance state (BIT SERIALIY)

INPUT	SA0	SA1	SA2	SA3
RWL0	R0*W00	R0*W10	R0*W20	R0*W30
RWL1	R1*W01	R1*W11	R1*W21	R1*W31
RWL2	R2*W02	R2*W12	R2*W22	R2*W32
RWL3	R3*W03	R3*W13	R3*W23	R3*W33

When weight stored like this

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

And when RWL are all

1	1	1	1

Output should be like this

SA0	SA1	SA2	SA3
1	0	0	0
0	0	0	0
0	0	0	0
1	0	0	1

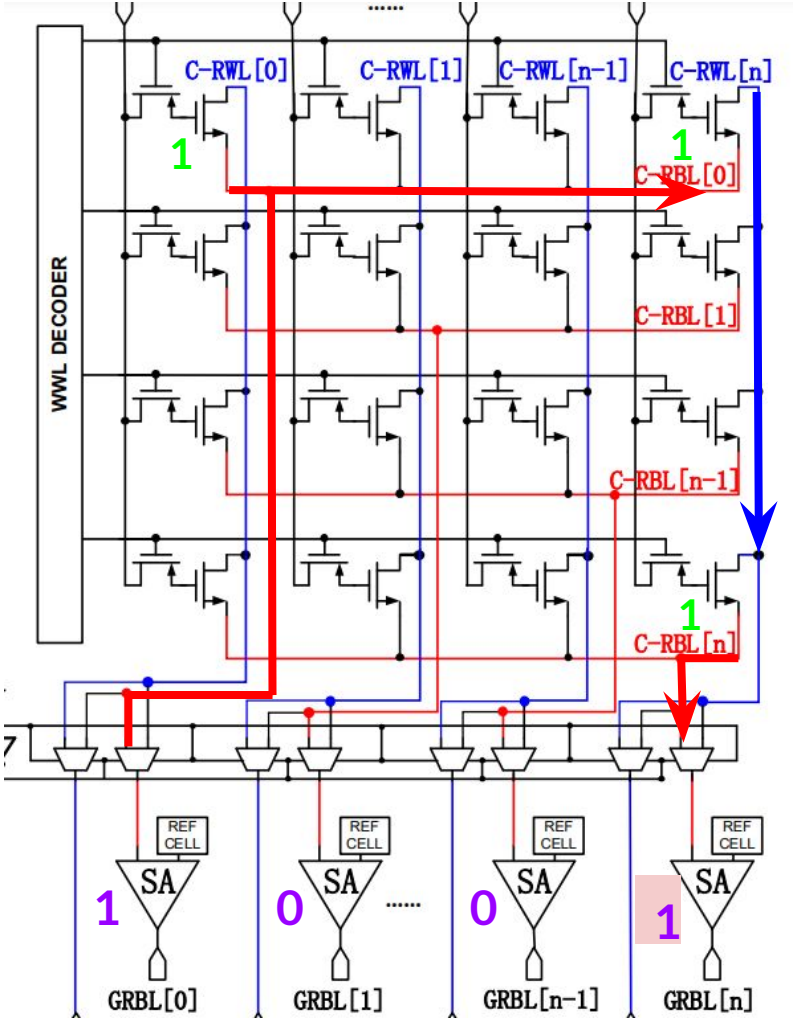
But we get

SA0	SA1	SA2	SA3
1	0	0	1
0	0	0	0
0	0	0	0
1	0	0	1

WHY is fails !!

When RWL and value stored both are 1, the RBL is reflects a 1. As this RBLis shared by all cell in that row, all the 2T cells receive RBL as a 1.

If value stored in some other cell of the same row is 1, it will switch its RWL to a 1 (1 bit AND) which was initially left as floating. Now, all cells of a same column share a RWL, so all these will receive RWL as a 1, instead of a floating value. And if data in any one cell With the shared RWL is 1 then the RBL for that Row will become 1 which will be then sensed by SA and will provide a false OUTPUT.



In all other cells data stored is 0.

The output of SA3 should ideally be 0 but dues to this it would be 1

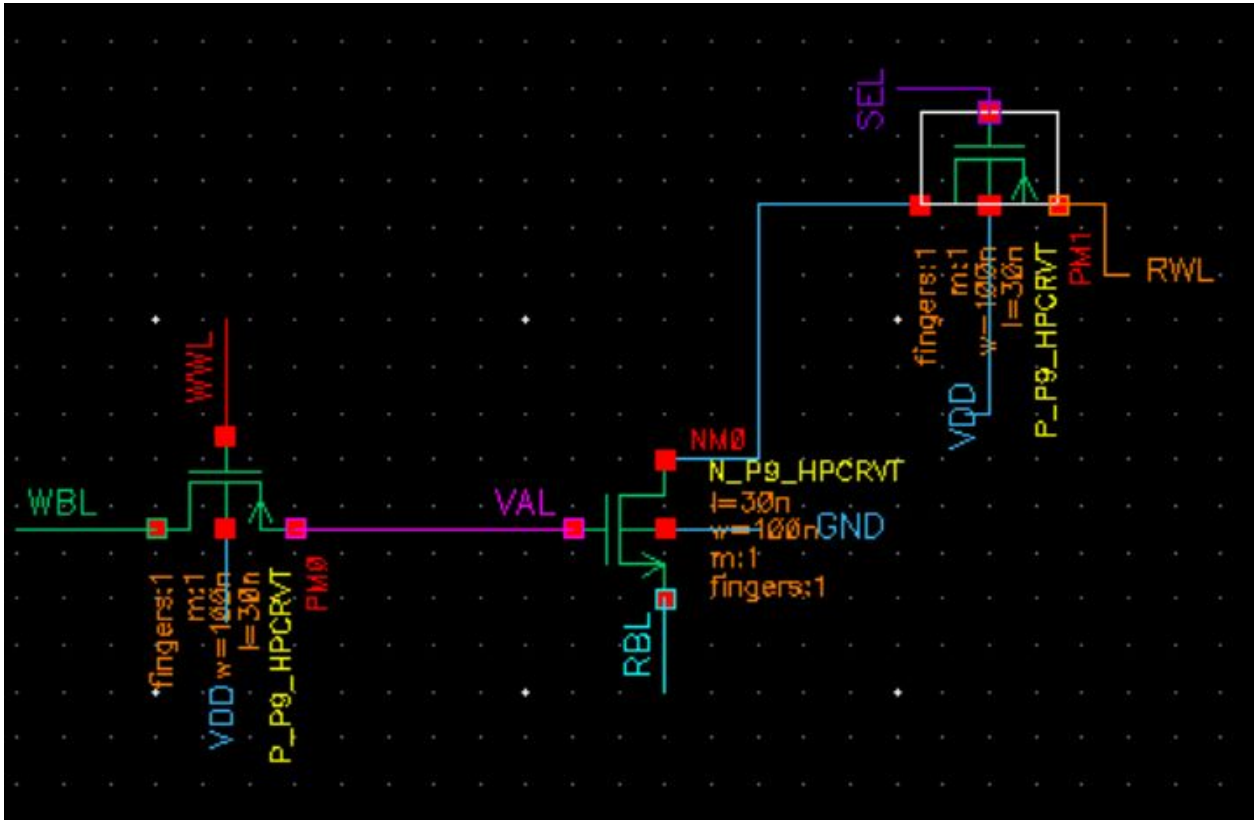
Potential Solution

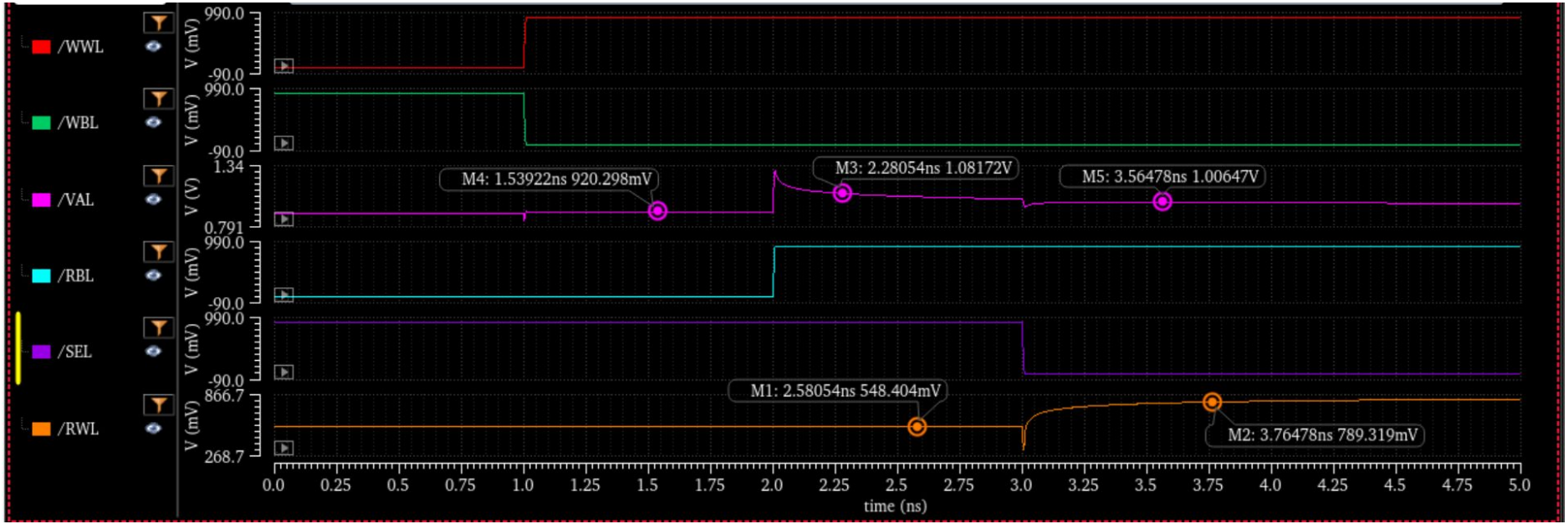
The problem arises due to shared RBLs. So, we can disconnect the RBLs of the cells with a pmos, essentially making a 3T DRAM cell. (but with 2 pmos and 1 nmos)

We have chosen a pmos as it will pass a strong 1. We don't need to pass strong 0 as we are already precharging with a 0.

This 3rd transistor (pmos) will have its own select line, common to all cells in a column.

3T-DRAM cell


SEL
RWL
PM1
P_P9_HPCRVT
l=30n
w=100n
m:1
fingers:1
VDD
WWL
WBL
PM0
NM0
N_P9_HPCRVT
VAL
GND
RBL



Thus, the RWL switches high only when the SEL signal is given (active low)

Acknowledgement

We would like to sincerely thank **Joycee** ma’am for allowing us to work on this project and for her invaluable guidance throughout. We also extend our gratitude to the **TA** for their continuous support and assistance, which greatly contributed to the successful completion of our work.