Enabling In-Memory Computations in Non-Volatile SRAM Designs

Siddhartha Raman Sundara Raman, S. S. Teja Nibhanupudi, *Graduate Student Member, IEEE*, and Jaydeep P. Kulkarni, *Senior Member, IEEE*

***Abstract*—The rapid growth in development of neural networks has necessitated the requirement of large capacity on-chip SRAM's for Machine Learning accelerators. This has resulted in SRAM's occupying significant portion of the die area. Furthermore, due to increased short channel effects in advanced CMOS technology nodes, the Vt of the transistors are increased to reduce the leakage power effectively. Vt increase results in direct increase in V_{MIN} (minimum operating voltage) of the device. The conventional 6T SRAM with the use of RRAM(R) to store the bitcell storage node values and PTM(S) as a selector device (6T-2R-2S) can help in decoupling V_{MIN} and Vt requirement with minimum area overhead. Functionalities of 6T-2R-2S bitcell are investigated to present a 2T-2R-2S mode and SRAM-RRAM hybrid mode of operation, further utilized in performing Compute in Memory(CIM). The above functionalities can be presented in a 8T2R bitcell, that makes use of transistor in place of PTM. 2T-2R-2S/2T-2R mode is a fully non-differential mode of operation, leveraging only the NVM portion of the bitcell. Furthermore, SRAM-RRAM hybrid mode is proposed making use of the SRAM read port transistors to perform read operation of the data stored on the RRAM, during standby mode. The architecture study of set-associative cache made of 6T-2R-2S array is also presented. The 2T-2R-2S/2T-2R mode coupled with SRAM-only mode can be efficiently used to perform CIM for dot product and XNOR computation with co-locating the weights and activations stored onto the same bitcell. System level study highlighting energy efficiency and speedup along with the proposed CIM architecture's analysis on CIFAR-10 dataset is presented. Design sensitivity analysis with respect to PTM and RRAM parameters is discussed for both the bitcells.**

***Index Terms*—Non-volatile SRAM, 6T-2R-2S, 8T-2R, resistive RAM, phase transition material, compute in memory.**

I. Introduction

THE unprecedented growth in data intensive compute has resulted in an increase in the size of Deep Neural Networks, further resulting in the requirement of large capacity on-chip Static Random Access Memory(SRAM)'s to perform energy efficient Machine Learning computations. This has resulted in SRAM's occupying more than 50% of die area, thereby incurring increased leakage power consumption during retention/standby mode. SRAM leakage power has dominated the overall power in advanced technology nodes. For instance, around 90% of the total power is contributed by the standby leakage power in 15nm technology node [1]. Leakage reduction is typically achieved by increasing the threshold voltage (Vt) of the SRAM transistors as compared to the logic transistors [1], [2].

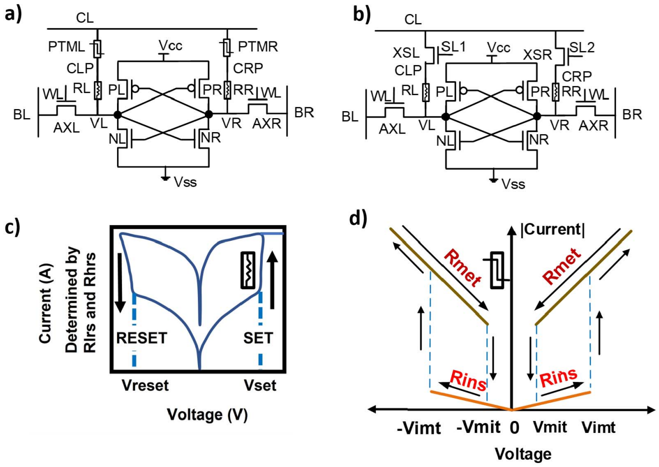


Fig. 1. (a) Non-Volatile SRAM bitcell with the 6T-2R-2S configuration (b) Non-Volatile SRAM bitcell with 8T2R configuration (c) I-V characteristics of RRAM (d) I-V characteristics of PTM selector.

Furthermore, to perform more computations in a given area, the bitcell density needs to be increased substantially. SRAM transistors are typically minimum sized in a given process technology, making them susceptible to systematic and random process variations, resulting in increased Vt mismatch between transistors. The cumulative outcome of these 2 effects (systematic and random variations) translates to increased sensitivity of circuit parameters, when operated at reduced supply voltage. This drastically increases the minimum operating voltage of SRAM (V_{MIN}) limiting the entire System On Chip (SOC) V_{MIN} increasing the total power at the product level. Since V_{MIN} directly relates to the potential failures during read or write operation, different assist techniques at the circuit level, including wordline under-drive (WLUD) [6], negative bitline (NBL) [7], Vcc collapse [8] have been explored. These consume significant power, larger area for peripheral circuits and increased latency. Thus, there is a critical need to develop

Manuscript received December 18, 2021; revised February 27, 2022 and April 11, 2022; accepted April 12, 2022. Date of publication May 10, 2022; date of current version June 13, 2022. This work was supported in part by the Semiconductor Research Corporation under Grant 2824.001 and in part by The University of Texas at Austin. This article was recommended by Guest Editor B. Kim. *(Corresponding author: Siddhartha Raman Sundara Raman.)*

The authors are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: s.siddhartharaman@utexas.edu; jaydeep@austin.utexas.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JETCAS.2022.3174148.

Digital Object Identifier 10.1109/JETCAS.2022.3174148

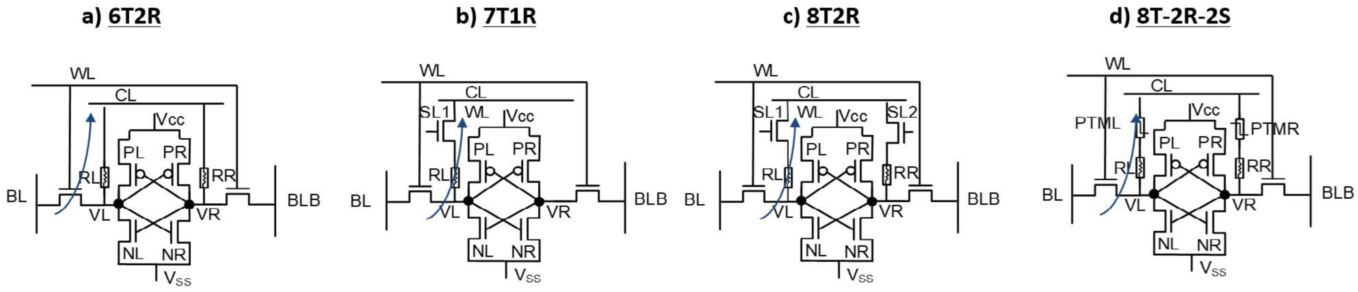


Fig. 2. NV-SRAM topologies of 6T-2R [3], 7T-1R [4], 8T-2R [5], 6T-2R-2S with arrows indicating current flow direction to program the RRAM.

a unique SRAM based technology which can decouple higher Vt (for lower leakage) and higher active-V_{MIN} constraints with minimal area overhead. The existing methods at the device level that try to decouple the above mentioned conflict involve the usage of an extra transistor, thus increasing the bitcell footprint area from 4 to 5 diffusion tracks affecting the storage density [3], [9], [10]. It is important to explore alternative device technologies that could be used for decoupling the fundamental conflict between Vt and V_{MIN} requirements to lower the V_{MIN} of the 6T SRAM. In this paper, we make use of the 6T-2R-2S bitcell proposed in [11] which uses RRAM to store the bitcell contents during standby mode with zero standby leakage and PTM as a selector device. 6T-2R-2S bitcell can be configured to a NVM-only mode by using only the access transistor and RRAM devices to perform a 2T-2R mode design. During sleep, the storage node values are stored onto the RRAM and post sleep, these values are restored back onto the SRAM storage nodes. The read-out process can be fastened by using the cross-coupled inverter pair instead of using the RRAM based read. In addition, this paper discusses the feasibility of performing 2T2R mode and SRAM-RRAM hybrid mode with 8T-2R bitcell as well. In case of 8T-2R bitcell [5], the selector devices are replaced by access transistors(XSL, XSR) with gate connected to SL1 and SL2 respectively as shown in Fig.1b).

Furthermore, with the advent of large size Deep Neural networks(DNN) for image processing, speech recognition, etc., there is an increased movement of data from off-chip memory to on-chip processing engines, leading to “memory wall” bottleneck. To mitigate this “memory wall” bottleneck, computations need to be embedded within the memory. In the existing compute in memory designs, the activations are usually stored in off-chip memory/separate storage memory and are transferred to the compute memory to perform computations. This results in increased data movement, and a true in-situ storage of activations is necessary to decrease this data movement. In this paper, we make use of 2T2R mode coupled with SRAM only mode to perform a truly in-situ compute in memory design that is capable of co-locating both weight and activation in a single 6T-2R-2S and 8T2R bitcell. As the size of the Machine Learning models increase, leading to large data storage requirement, the need for data compression without much loss in accuracy for inference is becoming increasingly common. One such data compressed model is the use of binarized neural networks, [12] with the input activations and the weight being 1 bit each. In such a scenario, multiply and accumulate operation performed in DNN can be realized using a simple XNOR operation. XNOR CIM is presented along with a system level study highlighting energy efficiency and performance for CIM.

The article is organized as follows. Section II provides a brief summary of the bitcell structures, along with I-V characteristics of the PTM and RRAM devices. NVM mode of operation of both the bitcells is discussed in Section III. Simulation results for SRAM-RRAM hybrid mode are presented in Section IV. Section V presents a circuit-architecture study for a set-associative cache operating in SRAM-RRAM hybrid mode made of the proposed 6T-2R-2S bitcell. Section VI discusses compute in memory design for performing dot product and XNOR computations and the key results for the proposed design with system level study in Section VII. Section VIII discusses the design space exploration along with the variability analysis followed by layout studies in Section IX and conclusion in Section X.

II. Device Study

This section gives an overview of the different device technologies like Resistive Random Access memory (RRAM), Phase Transition Material (PTM) based selector and the bitcells using these device technologies namely 6T-2R-2S and 8T-2R.

A. Resistive Random Access Memory (RRAM)

RRAM is a resistive non-volatile memory technology that relies on the principle of storing data based on the resistance of RRAM. These devices are made of Ag/Pt top and bottom electrode separated by a filament, capable of conduction. Bipolar RRAM device with positive SET voltages and negative RESET voltage is used (shown in Fig.1c). RRAM is initially in high resistance state and transitions into low resistance state once it crosses the set voltage. It further remains in the low resistance state until it crosses the negative Vreset threshold, when it transitions into low resistance state. A compact Verilog-A model considering the different device parameters with non-linearity of RRAM has been used [13]. The parameters of RRAM are listed in Table I.

B. Phase Transition Material (PTM) Based Selector

Phase Transition material(PTM) based selector devices are typically made of metal oxides like VO_2 and NbO_2 or doped

TABLE I
PARAMETERS OF RRAM

Symbol	Quantity	Chosen values
V_{SET}	Set voltage	1.15V
V_{MIT}	Reset voltage	-1.15V
R_{LRS}	Resistance in low resistance state	25k Ω
R_{HRS}	Resistance in high resistance state	1.25M Ω

TABLE II
PARAMETERS OF PTM SELECTOR

Symbol	Quantity	Chosen values
V_{IMT}	Insulating to metallic transition voltage	0.6V
V_{MIT}	Metallic to insulating transition voltage	20mV
R_{MET}	Resistance in metallic state	1k Ω
R_{INS}	Resistance in insulating state	0.12 GΩ
T_{TS}	Switching time	50ps

with other transition metals like in the case of Ag doped HfO_x and Cu doped HfO_x exhibiting hysteresis [14]. A phenomenological model of PTM selector using Verilog-A has been used for modelling the device. PTM is modelled to undergo an abrupt switch of resistance from a large insulating resistance to small metallic resistance for voltages greater than the insulating to metallic threshold V_{IMT} and vice versa for voltages lesser than metallic to insulating threshold V_{MIT}. The time required to undergo this transition is measured in terms of switching time of the device. I-V characteristics of the device are shown in Fig 1d). The parameters of PTM used in Verilog-A modelling are listed in Table II.

C. 6T-2R-2S Bitcell

6T-2R-2S SRAM bitcell design makes use of 2 RRAM devices as storage elements during sleep state and 2 phase transition materials as selector device. The bottom electrode of RRAM is connected to the bitcell storage node and the top electrode of RRAM to the PTM device as shown in Fig.1a. During the standby mode, the SRAM bitcell contents are retained by programming the RRAM into either low resistance/high resistance state depending on the bitcell contents achieving zero bitcell standby mode leakage. This helps in reduction of the high-Vt constraint typically present in a 6T SRAM bitcell design, further lowering the V_{MIN} of the design during read and write mode as shown in [11]. The phase transition material being used as selector device has been optimized so as to reduce the sneak path current of the proposed bitcell. The 2R-2S stack is accessed via a control line that is routed in the same direction as that of word line. This stack can be integrated backend of the line and doesn't add any additional area overhead [11]

D. 8T2R Bitcell

8T2R bitcell proposed in [5] involves 3D stacking of RRAM devices that are capable of performing fast-write and low current devices to achieve low V_{MIN}. Furthermore, this bitcell makes use of transistor instead of the PTM material in the RRAM access transistor path.

E. NV-SRAM Topology Comparative Study

This proposal aims at enabling non-volatile feature in the conventional SRAM by using additional CL and non-volatile RRAM stack. Fig.2 describes the different NV-SRAM topologies and the current direction needed to program the RRAM in the bitcell. The existing NV-SRAM topologies along with the proposed 8T2R2S topology need to be resilient to higher RRAM currents and the CL/WL needs to be routed in such a way that large currents can flow through the bitcell. The 6T-2R NV-SRAM bitcell suffers from extremely high short circuit current, leading to high short circuit power. 7T1R, on the contrary does not store the complementary storage node value onto RRAM and involves single ended sensing. 8T2R does not suffer from short circuit current and also stores both the true and complementary values onto RRAM, during standby mode, thus enabling differential sensing. This bitcell has an additional area overhead in comparison to the 6T-2R-2S bitcell because of the presence of front end transistor, thus increasing the number of diffusion tracks in the layout. This leads to higher routing congestion, when built with other standard cells in a typical processor design. The back-end integration of the PTM device, the low off-current of PTM, ability to store complementary values onto RRAM during standby mode, with ability to perform differential sensing in 8T-2R-2S cell provides an optimized version of a NV-SRAM bitcell

III. NVM MODE

NVM mode in case of 6T-2R-2S involves making use of 2 access transistors of SRAM, 2 RRAM's and 2 PTM's called as 2T-2R-2S mode. 2T-2R-2S mode is a fully differential Non-volatile memory only mode that makes use of RRAM to store the bitcell contents. In retention mode, the bitcell VCC and VSS nodes are connected to ground. CL, WL, BL, BR are also grounded until read or write operation is initiated. Connecting all nodes to ground is important because it prevents any charge build up on the bitcell storage nodes and helps in avoiding any accidental RRAM bit-flips while in the retention mode. NVM mode can be investigated in both 8T2R and 6T-2R-2S bitcells as shown in Fig.3a), Fig.3b), Fig.3c)

In write mode, bitcell Vcc and Vss nodes are connected to ground. The write operation is performed in 3 steps. In the first step, the residual charges on VL and VR nodes are removed by asserting the WL with both BL and BR connected to Vss (Fig.3d). In the second step, programming the RRAM to LRS is achieved by raising the corresponding bitline (BL) to around 1.3V and connecting the CL to a voltage lesser than Vss (around -0.2V). Negative voltage is kept at -0.2V to avoid forward biasing of the Source-substrate/ Drain-substrate junction diodes. Negative voltage at the CL is applied so as to reduce the voltage at BL, which can reduce the dynamic capacitance (Cdyn), thus leading to better performance. This is because the bitlines are shared across multiple rows in a single column, leading to overall performance impact. V_{IMT} of PTM is optimized so as to ensure that the sneak path current though the inactive cross-coupled inverter pair do not influence the RRAM programming. For example, for programming RL to LRS and RR to HRS state; BL is connected to 1.3V with CL connected to -0.2. This induces a current flow from BL→ AXL→ VL→RL→PTML→CL which triggers a phase transition in PTML and programs the RRAM RL to LRS. In the

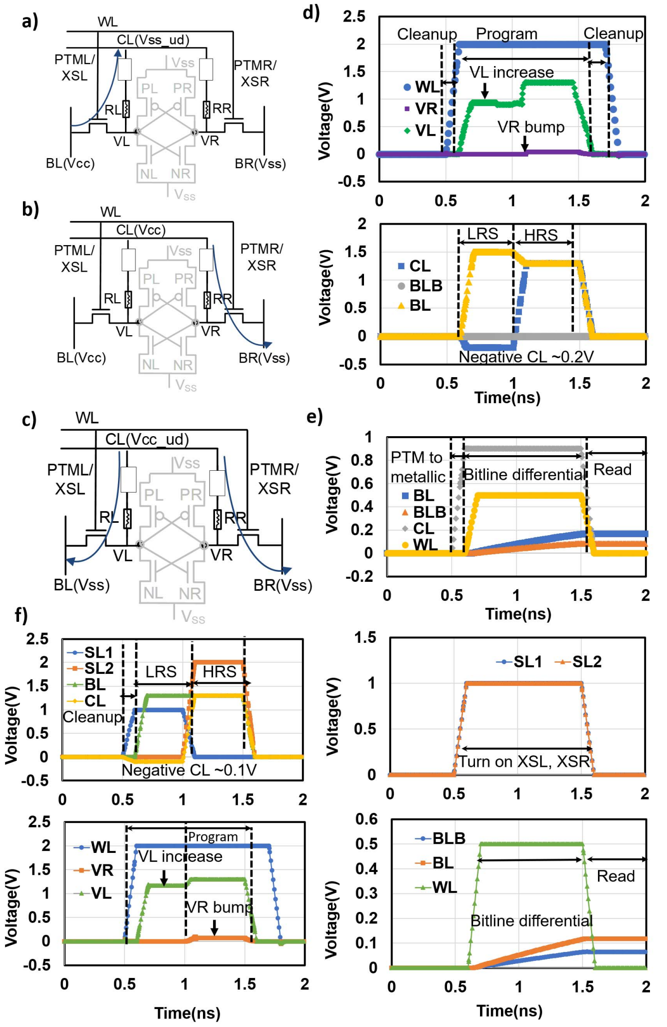


Fig. 3. (a) Current flowing from BL to CL in the direction shown for programming RL into LRS (b) Current flowing from CL to BLB in the direction shown to program RR into HRS (c) Reading the RRAM contents by current flow from CL to BL and BR. (d) Timing diagram for 2T-2R program with charge cleanup followed by program in LRS and HRS respectively followed by another charge cleanup at storage nodes (e) Timing diagram for 2T-2R read with transitioning the PTM into metallic state, allowing the bitline differential to develop, then read the RRAM contents (f) Timing diagram for 8T2R by turning ON XSL, XSR (g) Performing 2T-2R-2S read operation by turning on XSL, XSR.

third step, the CL voltage is raised to 1.3V thereby inducing a current flow from CL→PTMR→RR→VR→AXR→BR which triggers a phase transition in PTMR and programs the RRAM RR to HRS. In the fourth step, the bitcell Vcc and Vss are grounded while WL being active. This removes any existing residual charge and prevents any charge build up on the bitcell nodes. Biasing unselected row CL's to Vcc/2 ensures that respective selector devices (PTM) remain in the insulating phase and provides a high resistance path, thus alleviating the problem of programming of RRAM.

In the read mode, Vcc and Vss are connected to ground, ensuring nodes VL and VR do not contain any residual charge. The read operation is initiated by charging the CL node to Vcc. Most of the applied voltage appears across the selector devices (PTML and PTMR) which are in the insulating phase. This triggers a metallic to insulating phase transition for both PTM's. Once both PTMs are in metallic phase, the current flow is determined by the RRAM resistance values. It is assumed that RL is programmed to LRS and RR is programmed to HRS. In the second step, the asserted WL develops a bitline voltage differential on the BL and BR nodes because of the difference in resistances on RL and RR. A sense amplifier can resolve this BL voltage differential to full rail output. However, asserting WL to VCC with both bitlines to ground, CL is under driven to reduce the risk of programming both RRAM's to HRS leading to a read failure. To reduce the risk further, WL voltage level is modulated by under driving WL for successful non-destructive read operation in 2T-2R mode as shown in Fig.3e.

NVM mode in 8T2R bitcell involves making use of 2 access transistors of SRAM and 2 RRAM devices, called as "2T-2R mode". In the case of exhibiting 2T-2R mode using 8T2R bitcell, the 2 access transistors XSL and XSR are turned ON during different stages for current flow from CL to BL/BLB. For programming the RRAM RL into LRS, SL1 voltage is increased sufficiently to turn on the access transistor, ensuring current flow from BL to CL. The negative voltage at the CL is kept at −0.1V to avoid forward biasing of S/D junction diodes. It is important to note that all the MOSFET's in this configuration are of iso-width and iso-channel length. This leads to an increase in voltage at VL and voltage drop across RL is close to VSET voltage for programming into LRS. Similarly, SL2 is turned ON to program the RRAM in HRS with the current flow from CL to BLB. In the read mode, both the SL's are turned ON so as to allow the read current to flow from CL to BL and BLB.

IV. SRAM-RRAM Hybrid Mode

In this mode, the 6T portion of the proposed 6T-2R-2S bitcell is used as a read port for performing high speed RRAM read operation as shown in Fig. 3. The RRAM read latency is typically limited by the time required to develop enough bitline differential due to smaller resistance ratios between the two states. Furthermore, the voltage drop should be limited across the RRAM to avoid bit flips during read operation. To improve the bitline differential in the 2T-2R mode and also for a more robust read, a hybrid RRAM-SRAM feature is proposed. The read-speed of the hybrid bitcell is determined by the SRAM read port (access and pull down NMOS) rather than RRAM resistance. The key idea is to first perform a state-restore operation followed by a regular SRAM read operation.

The state restore operation [11] is carried out by turning on CL, restoring the bitcell contents that is stored onto RRAM(Fig.4a). It is assumed that initially RL is in HRS and RR is in LRS. CL is driven such that the PTML and PTMR transitions to metallic state by exceeding V_{IMT} and act like an assist to deposit charge onto the storage node before the bitcell Vcc is turned ON. The restore sequence begins by charging the CL node of the selected row to Vcc while the bitcell VccL and VccR are still connected to VSS(Fig.4b). This will induce current through both RRAMs CL → (PTML/PTMR) → (RL/RR) → (VL/VR) thereby charging the VL/VR nodes

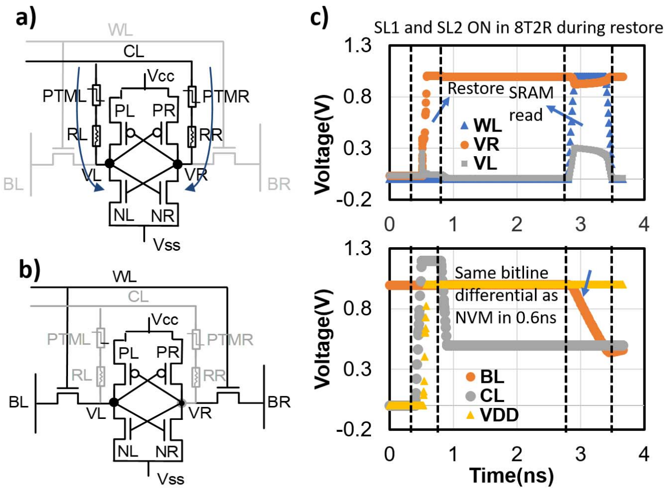


Fig. 4. (a) State restore operation by allowing the current from CL to Vss via access 2R-2S stack by transitioning PTM into metallic state/SL1 and SL2 turned ON. (b) SRAM read operation by transitioning the PTM into insulating state/ by turning off SL1 and SL2 (c) Timing diagram for SRAM RRAM hybrid mode for both 6T-2R-2S and 8T2R bitcells.

with different RC time constants. Then the bitcell VccL/VccR are charged to Vcc thereby driving the storage nodes VL/VR to full rail voltages. A steep increase in VR is observed in contrast to VL because of the LRS state of RR, thus having a smaller RC delay. Once the bitcell Vcc is turned ON,the small differential between VL and VR is amplified to a full rail voltage by the cross-coupled inverter based sense amplifier, as in the case of conventional 6T SRAM. Thus, the restore time is a function of the ratio of RRAM LRS to HRS resistance for the cross coupled inverter pair to contribute.

In the second step during the SRAM read only mode operation, the WL is asserted while the bitlines are pre-charged to Vcc (similar to a baseline 6T SRAM read). Unlike the 2T-2R read where the access time is a strong function of RRAM resistance ratios, the hybrid mode can achieve access times similar to the baseline SRAM because of the usage of read port transistors of the baseline SRAM. It can be seen from Fig.4c) that the BL differential is observed at a much lesser time in contrast to the read using RRAM in 2T2R (Fig.2e)

Similarly, in case of 8T2R, SL1 and SL2 needs to be turned ON before ramping up Vcc during restore operation so that the current flows from CL into storage nodes. On the other hand, SL's need to be turned OFF to ensure that SRAM read is not disturbed by the current flow from CL to storage nodes. Furthermore, it is important to note that the simulation results are similar for the case of 8T2R and 6T-2R-2S bitcell in terms of timing diagram, except for the fact that instead of PTM being in metallic state, the WL of the transistor is turned ON.

V. Architecture Study

As the proposed 6T-2R-2S bitcell helps in reduction of V_{MIN} by storing the bitcell contents onto RRAM devices during standby mode reducing the entire SOC power, it is important to explore the potential of such designs for the purpose of large scale cache designs. In this section, we analyze the aspects of energy, latency during write and read for a 6T-2R-2S based 8KB 4-way set associative cache using NVSim [15], a circuit and architecture co-design simulator that is capable of modelling emerging Non-Volatile Memories. A custom bitcell design is used with an area of standard 6T SRAM design because of the ability to integrate the RRAM and PTM back-end of the line and is further optimized for latency of the chip, with simulation details given as follows:

TABLE III
Performance Metrics of 8KB 6T-2R-2S Based Cache in SRAM-RRAM Hybrid Mode

Metric	Values	Comment
Area	0.0020 mm^2	Similar to 6T SRAM because of BEOL integration
Restore latency	64ns	Assumed that a byte in a cache line is accessed parallelly
Read latency	38ns	Assumed that a byte in a cache line is accessed parallely
Restore energy	501pJ	Restore energy per cache line of 64B using RRAM and TS branch
Read energy	490fJ	Read energy per cache line of 64B using SRAM cross coupled inverter pair

A. Memory Array Organization

Memory design is organized into different banks, which is the top level hierarchy. Each bank is assumed to be consisting of different mats which can be parallelly accessed, with each bank consisting of different sub-arrays and a pre-decoder block, used for pre-decoding the address bits. Sub-array is the basic building block consisting of bitcell array along with peripheral circuits for column multiplexing, row decoders, sense amplifiers for differentiating the bitcell content.

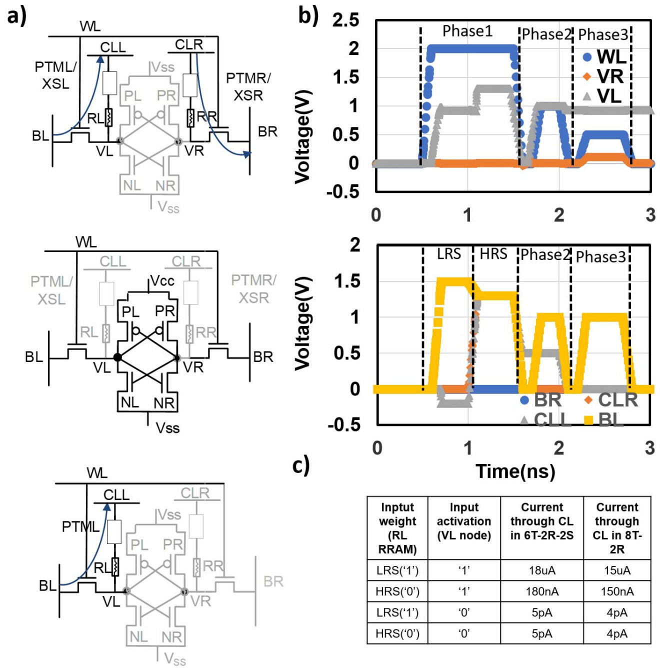
B. Memory Type

6T-2R-2S based memory is assumed to be 4-way set associative cache, 64B cache lines with decoupled data and tag array. The tag and data array of the cache is accessed simultaneously. The entire set (consisting of 4 ways) is read from the mats to I/O interface. During the same time, tag compare operation is performed between the input address and the tag stored in tag array. The desired cache line out of the entire set is then read out, provided there is a cache hit.

C. Energy and Latency for Different Operations

The SRAM-RRAM hybrid mode can be efficiently used for performing a read operation post-standby operation. The different operations involved include the restore and the SRAM read operation. Latency of these operations for accessing a cache line is listed in Table III. Read latency is the latency for performing the SRAM read operation using the SRAM-RRAM hybrid mode once the voltages are read/restored from standby mode.

Restore latency is the time taken to restore the contents using the SRAM-RRAM hybrid mode post standby operation. This latency is similar to the read latency for performing 2T-2R-2S read operation. Furthermore, it is important to note that a byte in cache line is assumed to be restored/read parallelly and the subsequent bytes are accessed serially.



Inptut weight (RL RRAM)	Input activation (VL node)	Current through CL in 6T-2R-2S	Current through CL in 8T-2R
LRS('1')	'1'	18uA	15uA
HRS('0')	'1'	180nA	150nA
LRS('1')	'0'	5pA	4pA
HRS('0')	'0'	5pA	4pA

Fig. 5. (a) 3 phases of CIM architecture with first phase storing the weight and its compliment, storage of activations onto the storage nodes during second phase and performing dot product compute during third phase followed by current comparison (b) Timing diagram of the proposed CIM design with RR and RL being programmed showing a potential of dot product design (c) Table showing the current comparison for computing the dot product in case of 6T2R2S and 8T2R.

Energy for read is the energy required for reading the stored content by using SRAM read operation after developing enough differential. Energy for restore is observed to be similar to the energy for read operation in a 2T-2R-2S because the restore path is similar to the read path, consisting of a PTM, RRAM and access/pull down/pull up transistor, characterized as:

$$E_{\text{restore}} = I^2 * R_{\text{LRS/HRS}} * t_{\text{SET/RESET}} \quad (1)$$

where $t_{\text{SET/RESET}}$ is the pulse width for obtaining the necessary read current. I is the average current during the programming and $R_{\text{LRS/HRS}}$ is the resistance during LRS/HRS state.

VI. Compute in Memory

A. Dot Product Compute

2T-2R-2S mode can be effectively leveraged to perform a true in-situ compute in memory by storing both the activations and weights in the same bitcell. The compute in memory is done in three phases. In the first phase, the weights are stored onto the RRAM by programming RL into LRS or HRS, depending on the weight value. Only the left branch is made use for programming purposes as shown in Fig.5a) with the current flow from BL to CLL via AXL, RL, PTML/XSL. The complement of weight is stored onto RR. In the second phase, the activations are stored onto the bitcell storage node by ensuring that the PTM is in insulating phase, thus ensuring SRAM only mode of operation. In the third phase, BL is charged to Vcc, the voltage difference between BL and CLL is sufficient to transition the PTM into metallic state and the current flowing through CLL branch via AXL, RL, PTML/XSL is a measure of the dot product computed between the activation and weight values. Current comparator is used with the reference current of 0.5uA is used to calculate the dot product as shown in Fig.5c). In case of 8T2R bitcell, SL1 is turned ON with SL2 turned OFF during programming RL into LRS and SL2 is turned ON with SL1 turned OFF during programming RR into HRS, i.e. for storing weights and their complement onto the bitcell. Furthermore, SL1 and SL2 both are turned OFF during the second phase where the 6T SRAM portion is utilized for storing the activations onto the SRAM storage nodes. In the third phase, SL1 is selectively turned ON so as to perform the dot product computation between the input activation and weight. Current comparison is performed as in the case of 6T-2R-2S and the current flowing through the CLL node for different cases are given in Fig.5c)

B. XNOR Compute

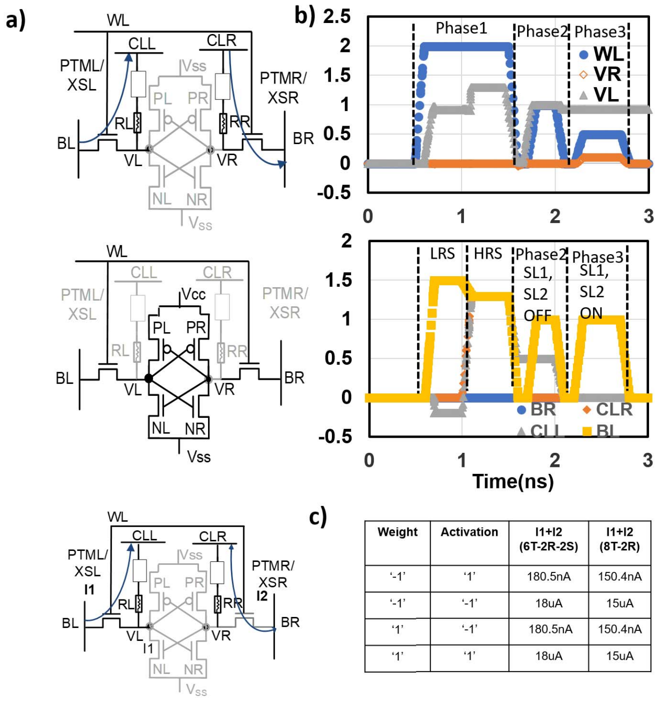
XNOR compute is extensively used in the case of binarized neural networks, where the weights and activations are both 1-bit. Weights and activations can take values of either -1 or 1 which are digitally resolved as 0 and 1 for the purpose of computation. XNOR compute between A and B can be computed as AB+A'B', where A is the weight and B is the input activation. This can be realized by a 2 step process. The first process involves performing the dot product between A and B. The second process involves performing the dot product between A' and B'. Unlike the dot product compute, both the left and right branches can be effectively used to realize the XNOR operation. RL and RR combination are programmed in complementary state for storing the input weight and its complement in phase 1 (as shown in Fig.6a) with the current flow from BL→ AXL → RL → XSL/PTML → CLL in the left branch. The current flow on the right branch is from CLR → XSR/PTMR → RR → AXR → BR in phase 1. The input activations are stored onto the storage nodes in phase 2 by disabling the selector and RRAM branch, similar to the dot product compute. In phase 3, the selectors are turned ON(transforming the PTM into metallic state/turning ON the access transistor) and the current flowing through both the access transistor, selector and RRAM branch is summed and compared against a reference current value (0.5uA) to obtain the XNOR compute as shown in Fig.6b).

For example in the case of computing XNOR product between weight of '1' and '−1', RL is programmed into LRS state and RR is programmed into HRS. The input activation is stored onto storage node. The current flow summation(I1+I2) will be equal to the sum of current flow for the dot product performed between 1) weight of '0' and activation of '1' 2) weight of '1' and activation of '0'.

In the case of 8T-2R, during the phase 1, the word line of the RRAM branch transistor is turned ON. Phase 2 involves storing the input activations onto the storage node by turning off the selector transistor. Phase 3 performs XNOR by computing RRAM read current by turning on XSL, XSR

C. CIFAR-10 Analysis

The complex DNN networks for image recognition typically consist of multiple convolutional layers to capture



Weight	Activation	I1+I2 (6T-2R-2S)	I1+I2 (8T-2R)
‘-1’	‘1’	180.5nA	150.4nA
‘-1’	‘-1’	18uA	15uA
‘1’	‘-1’	180.5nA	150.4nA
‘1’	‘1’	18uA	15uA

Fig. 6. (a) Three phases of CIM architecture with first phase storing the weight and its compliment, storage of activations onto the storage nodes during second phase and performing XNOR compute during third phase followed by current comparison (b) Timing diagram of the proposed CIM design with RR and RL being programmed showing a potential of XNOR compute design (c) Table showing the current comparison for computing the XNOR compute in case of 6T-2R-2S and 8T2R.

the essential information of an image. These convolutional layers can be realized by performing Multiply and Average operations (MAV) between the input activations and weights. Convolution operation is assumed to be performed between an input feature map of M*M*C and weight matrix of R*R*C. These MAV operations are performed using 6T-2R-2S array and the accuracy of these operations are tested for a CIFAR-10 dataset on a custom CNN with 8-bit input activations and weights. CIFAR-10 dataset consists of 60000 test images and 10 classes with 10 6000 images for each class.

The key features of the proposed NV-SRAM design in comparison with the existing approaches for CIM are as follows:

- In SRAM/DRAM or volatile memory based CIM approaches, the activations need to be stored separately and weights are stored onto the leaky SRAM/DRAM storage nodes. Activations are mapped onto WL/BL during compute phase and significant data movement of activations from storage to compute array.
- In RRAM or NVM based CIM approaches, the activations need to be stored separately and weights are stored onto the non-volatile storage nodes. Activations are mapped onto the WL during compute phase and this requires data movement of activations from the storage array onto compute array.
- In contrast to the above 2 approaches, the proposed approach provides an optimized version of data movement of activations by co-locating the weights and activations onto the same bitcell. In the proposed design, the stationary weights are stored onto a non-volatile RRAM stack in-situ and the activations are stored onto the SRAM storage nodes on the NV-SRAM bitcell. As a result, the intra-memory data movement would be reduced in comparison to the above approaches. Furthermore, the data movement can further be decreased, if the the output activations of the (n-1)th layer are stored near to the weights of (n)th layer in the same compute array.

However, in all the three approaches, if the model size is higher in comparison to the on-chip compute array size, the movement of activations from compute array to storage array is inevitable. The CIM architecture and design can be summarized as follows:

1) Mapping of Weights and Activation: The activations and the weights are co-located in a single NV-SRAM bitcell with the weights stored onto the non-volatile RRAM stack and the activations stored on the storage nodes of the SRAM bitcell as shown in Fig.7a. 8-bits corresponding to the input weights are stored in a single row of 6T-2R-2S array. A single bit of input activation is stored in 8 columns of a single row in an iteration. The same activation is stored in 8 columns, thus reusing the activation for every 8 columns (bit-width of each weight). This further results in increased parallelism without additional circuitry overhead. The activations from the previous layer are read from the compute array and stored onto the bitcell that stores the corresponding weight in an iterative fashion with the lower significant bits stored in each iteration followed by the most significant bit for an n-bit activation(in this case n = 8). On computing the dot product between input feature map of M*M*C and weight matrix of R*R*C, the filter slides across the entire input feature map, with a stride of 1. When the filter slides for computing the output activations of nth layer, the input activations are read from the location where the output activations from (n-1)th are originally stored and are written back onto the locations closer to the already computed activations of nth layer. The controller logic necessary to do this sliding process is similar to the case where the input activations are stored in separate activation array and are applied on the WL or BL. Thus, the dot product is computed, and the activation is stored back onto the same compute array.

2) MAV Compute: MAV operation consists of performing dot product followed by averaging operation. The dot product is obtained by performing the current based sensing approached mentioned in Section VI.A. The sensed current is converted to an equivalent voltage using I-V converter and each bitline is connected to a capacitor. These capacitors are binary scaled so as to effectively distinguish between the different weight bits stored in a row. The capacitors are shorted so as to obtain the averaging functionality as shown in Fig.7b. This analog value is then converted into an equivalent digital value by using an 8-bit ADC, implemented in [16] so as to realize MAV operation between 1-bit activation and R*R*C*8-bit weights. The obtained value is stored in a buffer typically as in the case of any CIM design. This process is

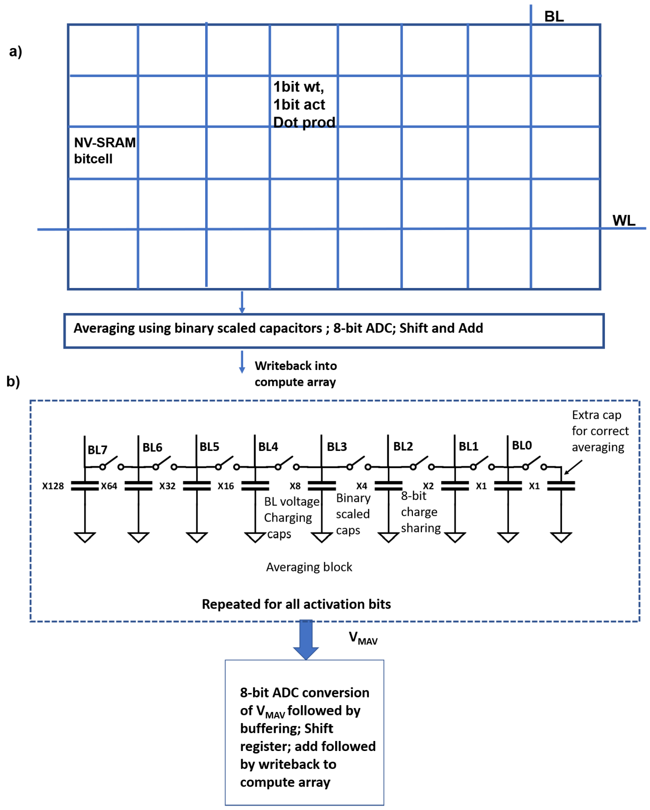


Fig. 7. (a) Compute array storing both activations and weights, BL current based sensing followed by charge sharing using binary scaled capacitors for averaging, ADC and shifting, addition operation. (b) Charge sharing operation using binary scaled capacitors for performing averaging functionality.

repeated so as to obtain the MAV operation for the other 7-bits of activation as well.

3) Shift and Add: The equivalent digital value obtained from performing MAV across a single column is then shifted and added to realize the effective MAV between 8-bit input activation and 8-bit input weight.

4) Writeback: The obtained output activation can be written to a separate row in the compute array instead of writing back onto the same row so as to avoid data corruption. This writeback cannot be performed in typical CIM designs where only weights can be stored in the compute array, thus the output corresponding to each layer needs to be written back to a separate activation storage array. The feature of co-locating weight and activation results in reduction of energy needed for data movement between compute and storage array. The activations corresponding to a single layer are co-located in locations closer to the weights of the next convolutional layer so as to reduce the data movement necessary to perform MAV operation for the next layers.

The top-1% classification accuracy for a custom CNN tested using CIFAR-10 comes to be 84%, shown in Table IV with the convolutional layers MAV operation performed in the memory array. The software accuracy is at 89%, tested using PyTorch.

VII. System Level CIM Study

The proposed design has 2 key features. Firstly, the weights are stored in non-volatile memory stack, thus leading to storing the contents on the bitcell for a long time. Secondly, the data movement for activation is reduced by storing activations onto the compute array. This section highlights the different features of the proposed Compute in Memory Design and gives the necessary highlights of the design. Furthermore, an analysis is performed with respect to different metrics like energy, latency comparing the proposed design with a baseline 1T1R based CIM design.

A. Baseline 1T1R CIM Design

The baseline 1T1R bitcell consists of 1 transistor used as a selector, activated by WL and 1 RRAM for storage. CIM design used in the baseline design involves storage of weights onto the RRAM during phase 1 and the activations are mapped onto the WL during phase 2. The current flowing through the 1T1R during phase 2 is a measure of the dot product between input activation and weight. The dot product is computed and the charge sharing of the obtained dot product results in MAV. This is followed by the shift and add operation for performing dot product between multiple bits of activation and weight.

B. Memory Architecture

Memory type for the baseline design is 1T1R and the proposed design uses 6T-2R-2S based bitcell. Size of both the designs is initially assumed to be 256*256 and this size is swept across 128*256, 512*512, etc., so as to capture the design space exploration study on the percentage of activations that can be stored. The memory architecture (Fig.8a) assumed for performing the comparison involves making use of separate subarrays for storage and compute. In case of both the baseline and proposed design, the storage sub-array stores the input activations initially. During the compute phase, the input activations are transferred onto the compute-subarray. The in-situ activation storage scheme can be leveraged to store the computed MAV which will be the input activations to the next layer onto the same compute sub-array, whereas this has to be transferred to the storage sub-array and transferred back again in case of the baseline design. The over-spilled bits which cannot be fit inside the compute sub-array need to be transferred to the storage sub-array. This leads to improved energy efficiency, performance for the proposed design. The performance impact is measured in terms of the speedup of the design that is measured as the ratio of latency of data transfer operations and the compute operations in the proposed and baseline design.

C. Performance

The proposed compute in memory design is highly optimized for parallelism starting from the mapping phase. Different activations corresponding to different weight bits can be executed in parallel by taking advantage of the storage of activations and weights in-situ. Each activation bit is re-used across 8 columns in a single row. This operation can be parallelized across the entire NV-SRAM array. In baseline design, where the dot product is performed by mapping activation onto WL, additional logic overhead is necessary to reuse the activations every 8 columns by “breaking” the WL.

TABLE IV
CONVOLUTIONAL NEURAL NETWORK PARAMETERS

Metric	Description
Convolutional Neural Network	CONV layer (MAV) – 16 3x3; ReLU CONV layer (MAV) – 16 3x3; ReLU Max Pooling layer – 2x2 CONV layer (MAV) – 32 3x3; ReLU CONV layer (MAV) – 32 3x3; ReLU Max Pooling layer 2x2 FC layer – 512 – Batch Norm – ReLU FC layer – 10
Accuracy	Software accuracy - 89% Hardware accuracy - 84%

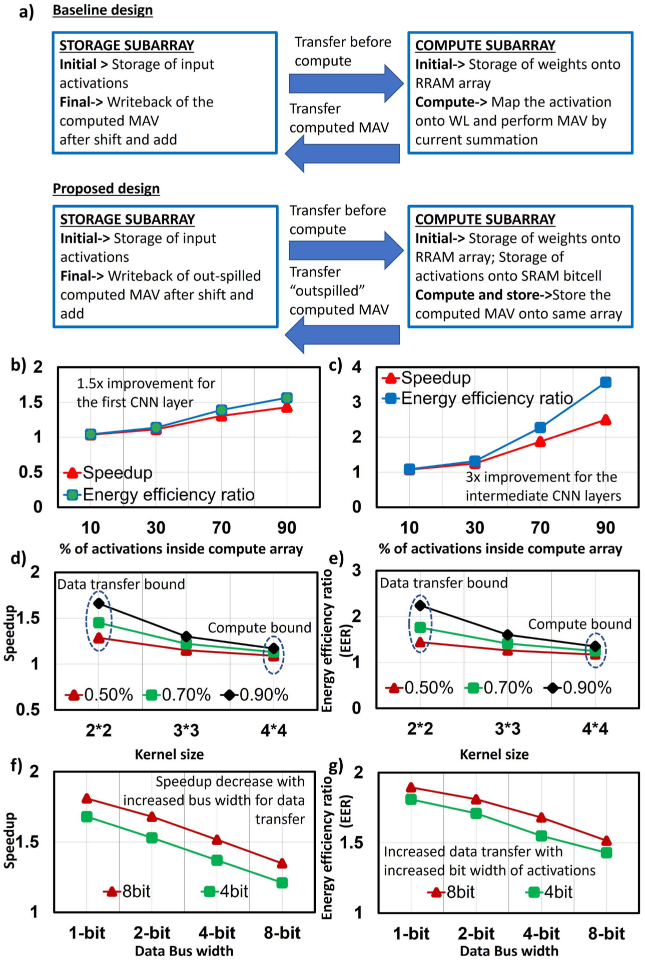


Fig. 8. (a) The memory architecture, partitioned into storage and compute sub-arrays. Proposed architecture performs better in terms of data transfer movement. (b) Speedup and energy efficiency observed as a function of the percentage of activations that can fit inside the compute array for a first convolutional layer of CNN. (c) Speedup and energy efficiency observed as a function of the percentage of activations that can fit inside the compute array for intermediate convolutional layers of CNN. (d) Speedup and (e) Energy efficiency as a function of kernel size as the percentage of activations that can fit inside the cache varies from 50-90% (f) Speedup and (g) energy efficiency variation as a function of data bus width and quantization.

D. Energy Efficiency

Energy efficiency is a measure of the efficiency of computations with respect to energy and is defined as ratio of Throughput and Power [16]. Energy estimation is broken down into energy for performing data movement for input activations, weights, energy of performing ADC, energy for performing MAV and energy for performing the data movement for the computed output activations. The energy of data movement of activations is significantly different because the baseline design needs to read the activations from a standby "activation storage" array and write back of the computed activation value into activation storage array. In the proposed design, the activation needs to be read from the activation storage array only the first time it is fetched and the compute array can be used as a storage buffer by co-locating weights and activations. Thus, the data movement of activations is drastically reduced. The overall energy values for ADC is 3fJ in the case of multi-bit weight and activation. The overall energy varies from 40fJ-60fJ depending on the array size. The compute energy in 256*256 RRAM array is around 10fJ in NV-SRAM with 8fJ in RRAM bitcell. This difference is primarily because of the additional overhead of writing and different RC in the network. The energy of data movement (20-30fJ) roughly contributes to 50-60% of overall energy. Throughput is further defined as the number of operations computed in a single cycle. The proposed CIM design increases the throughput by 2 means. Firstly, the time required for storing the activation onto compute sub-array is very less when compared to storing activations back onto the storage sub-array because of lesser data movement. Furthermore, for performing MAV on deeper CNN's with multiple layers, significant amount of data movement can be reduced. For performing a single compute of the nth convolutional layer, transferring the (n-1)th layer outputs back to compute array followed by transfer of computed dot product from compute array to storage array is necessary. Both these data movements can be mitigated in the proposed design by storing the output activations directly onto the sub-array.

E. Design Space Exploration

The data movement contributes upto 62% of the overall energy consumption, [17] with the energy for performing data movement being 115x more energy than an addition operation. The energy efficiency and latency improvement of the proposed design is a strong function of the sub-array size, size of the convolutional neural network and data bus width. This analysis, by default assumes an image of 64*64*3 convolved with 16 filters each of 3*3 size(0 padding and unit stride), which will be described in the following section.

Fig.8b) describes the speedup and the ratio of energy efficiencies as a function of the percent of activations that can be stored in the compute array. The entire CIM process is divided into 3 phases. The first phase involving the transfer of activations from the storage sub-array (baseline)/compute array (proposed) onto compute sub-array plays an important role in data movement energy. This is followed by compute operation in Phase 2 which determines the energy for performing MAV and ADC. The write-back onto the storage or compute sub-array depending on the design is performed in the final phase, plays an important role in the data movement energy. The ADC overhead is observed to be same for both baseline and proposed design and this energy is observed to be lesser than the energy of data movement. Initially, the input activations of the first convolution layer (Fig.8b) are

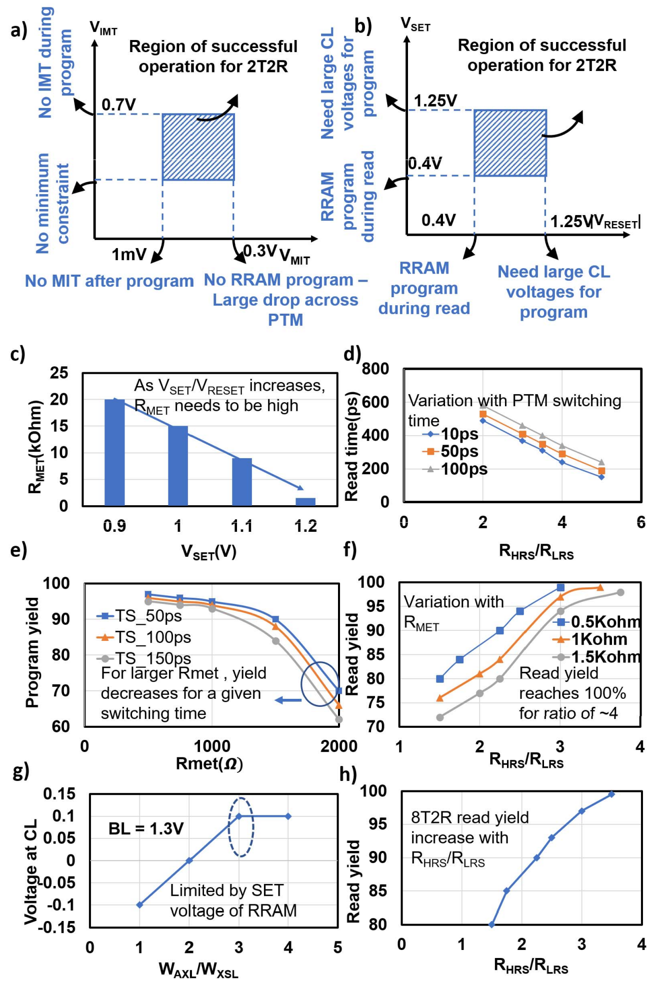


Fig. 9. a) Design considerations on PTM parameters like V_{IMT} and V_{MIT} b) Design considerations on RRAM parameters like V_{SET} and V_{RESET} c) Relationship between range of allowed values of R_{MET} and RRAM's V_{SET} for RRAM program d) Variation of the read time with Rmet e) Program yield dependence with the R_{MET} and switching time of PTM f) Variation of read yield with R_{MET} g) Variation of voltage at CL with ratio of access transistor and the selector transistor in 8T2R h) Read yield variation as a function of ratio of R_{HRS} to R_{LRS} in 8T2R.

assumed to be stored in the storage sub-array. Therefore, the speedup and energy efficiency improvement is obtained primarily because of the improvement in Phase 3. As the percentage of output activations that can be fitted inside the cache increase, the speed-up and energy efficiency also increases because of the reduced data movement. Fig.8c) shows the speedup for an intermediate convolutional layer where in x% of the input activations and output activations are stored onto the compute array. Thus, the speedup/energy improvement comes during both Phase 1 and Phase 3. This further results in speedup obtained in Fig.8c) greater than speedup in Fig.8b).

The input filter size determines the number of MAV and the size of output activations. The size of output activations determine the latency for Phase 1 and Phase 3. The size of output activations coupled with the square of kernel size determines the Phase 2 latency. The size of output activations decrease as the size of the kernel increases for a given input activation size. However, the effect of increase in kernel size compensates the effect of decreased output activation. Thus, the speedup is limited by the Phase 2 computation time as the kernel size increases. This results in a saturation in speedup to a value close to 1 as the kernel size increases, as described in Fig.8d), e).

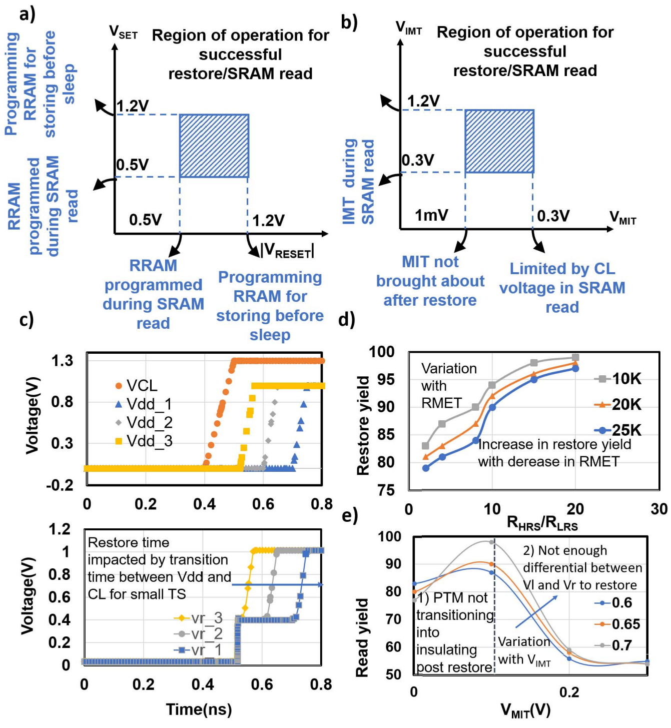


Fig. 10. a) Design considerations on PTM parameters like V_{IMT} and V_{MIT} b) Design considerations on RRAM parameters like V_{SET} and V_{RESET} c) Impact of transition between CL and Vcc impacting the restore time d) Restore yield as a function of R_{MET} and ratio of R_{HRS} to R_{LRS} e) Read yield dependence on V_{IMT} and V_{IMT}.

Fig.8f) and g) shows the impact of data bus width and quantization of input activations, weights on speedup and energy efficiency. The data bus width relates to the number of bits that can be parallelly read from/written onto the storage sub-array. Increase in bus width reduces the speedup because of the reduction in latency for Phase 1 and Phase 3. Similarly, the increase in quantization impacts all the phases, with more impact being on the data transfer phases because of the sheer amount of data movement that is required. Thus, speedup increases with increase in bit-width and decrease in bus-width.

VIII. Design Considerations

A. NVM Mode

In the case of 2T-2R-2S mode in 6T-2R-2S, the PTM parameters such as V_{IMT} and V_{MIT} need to be optimized for successful operation. V_{IMT} maximum constraint is imposed by the inability to program during program. Similarly, the maximum constraint on V_{MIT} of PTM is governed in such a way that it doesn't transition into insulating state, thus having a large drop across the RRAM and hence causing write failure. At the same time, the voltage should be high enough that the PTM transitions into insulating state after the programming is complete(Fig.9a). RRAM programming is

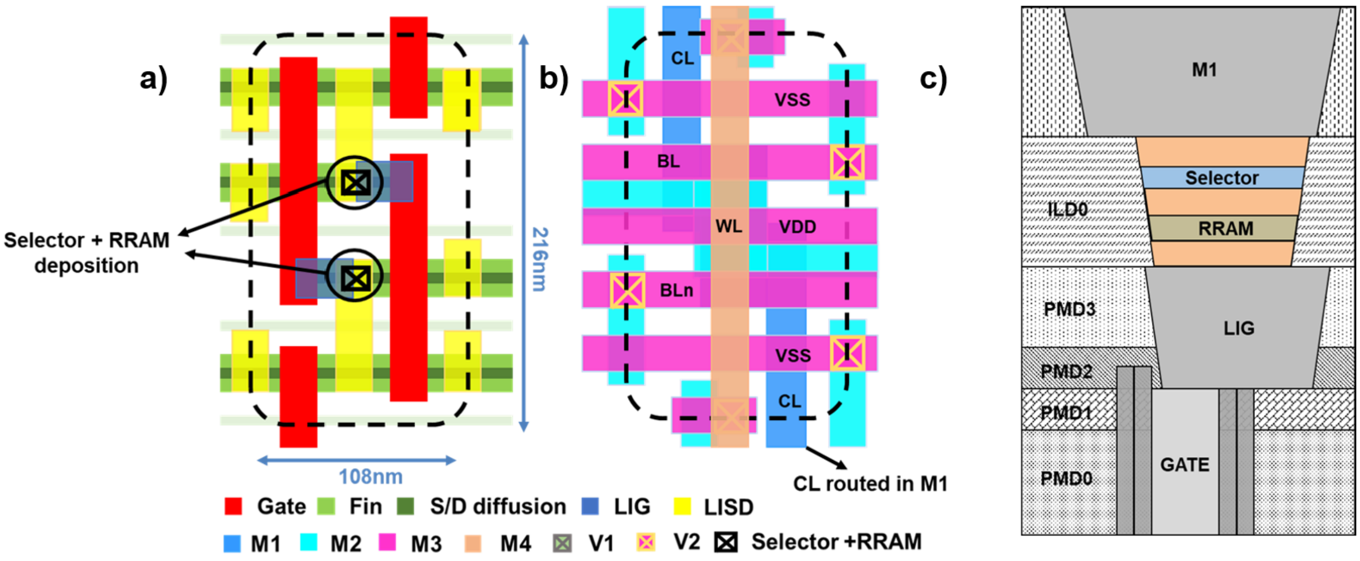


Fig. 11. Layout of 6T2R2S for the a) front end of line, b) back end of line using ASAP 7nm PDK c) heterogeneous integration of the selector and RRAM.

governed by the SET and reset voltages of RRAM. Assuming the bipolar RRAM case, the constraints of V_{SET} and V_{RESET} are similar. V_{SET} should be sufficiently large to avoid accidental programming during read. The minimum value of V_{SET} is limited by RRAM not getting programmed during write, as shown in Fig.9b. There is a direct relationship between V_{SET} and R_{MET} of the PTM. In the case of V_{SET} or V_{RESET} being small, the range of allowed values of R_{MET} is high because even a large drop across PTM can program the RRAM. As V_{SET} increases, R_{MET} needs to be sufficiently small enough to program RRAM, thus limiting the range of values of R_{MET}(Fig.9c).

Read time is governed by the combination of ratio of R_{HRS} to R_{LRS} and the switching time of the PTM. For smaller values of PTM switching time, the read time is limited by the ratio of R_{HRS} to R_{LRS} and switching time becomes the limiting factor for small values of R_{HRS} to R_{LRS} ratio(Fig.9d).

Program yield analysis is identified by performing 10^5 Monte Carlo simulations by including 1 sigma variation of 30mV for the transistors and 1σ variation of 5% of the parameters for the RRAM and PTM parameters. The program yield is dependent on the metallic resistance of the PTM and the PTM switching time. As R_{MET} increases, yield starts decreasing(Fig.9e) because of the increase in voltage drop across the PTM, thus increasing the risk of RRAM not getting programmed. As switching time increases, there is a potential issue of the PTM not transitioning into metallic state in the given program time. Similarly, the constraints on read yield is dependent on the ratio of R_{HRS} to R_{LRS} and the R_{MET}. As the ratio of resistance increases, the read yield starts increasing because of the increase in bitline differential observed. R_{MET} plays a role in case of determining the yield when the HRS to LRS resistance ratio of RRAM is low. This is because of PTM resistance contributing to the bitline differential, thus reducing the net effect of RRAM difference(Fig.9f). The effect of R_{MET} affecting read yield is not observed as the ratio increases.

In case of 8T-2R bitcell, the ratio of width of access transistor to the width of selector transistor(XSL) determines the minimum voltage required at CL to program the RRAM into LRS. As the ratio increases, the negative voltage at CL reduces subsequently being limited by the SET voltage of RRAM(Fig.9g). Furthermore, the read yield is governed by the R_{HRS} to R_{LRS}, assuming all MOSFET's are 1fin devices. The read yield reaches around 100% for a ratio of 4(Fig.9h)

B. SRAM-RRAM Hybrid Mode

In the case of SRAM-RRAM hybrid mode, the design considerations of the RRAM and PTM parameters are more stringent as shown in Fig.10a) and b) respectively. V_{SET} and V_{RESET} of RRAM have similar design considerations as in the case of 2T-2R-2S mode. The minimum value of V_{SET} is limited by the accidental programming of RRAM during SRAM read, with the maximum value of V_{SET} limited by the risk of not storing the bitcell contents onto the RRAM before the bitcell goes to sleep state, assuming that similar voltages are used for storing the bitcell contents referred to as state store operation [11]. In case of the PTM, the minimum value of V_{IMT} is limited by the potential transition to metallic state during SRAM read that might lead to sneak path current and programming of RRAM during SRAM read. The maximum value of V_{IMT} is due to the inability to program the RRAM during state store operation. Furthermore, V_{MIT} is bounded on the lower end by the PTM not transitioning into metallic state post-restore and by the inability to have sufficient voltage difference between the bitcell storage nodes because of the PTM on the side with RRAM in LRS state transitioning into insulating state. The time difference between the arrival of CL and arrival of bitcell Vcc impacts the restore time (Fig.10c).

Furthermore, the restore yield is limited by the ratio of R_{HRS} to R_{LRS} of RRAM (Fig.10d). The restore yield is a strong function of the R_{MET} of PTM. As the R_{MET} increases, the effect of difference in RRAM's HRS to LRS ratio is nullified, thus not having enough differential at the storage nodes for the cross-coupled inverter to restore the bitcell storage values correctly. As the ratio of RRAM HRS to LRS is increased, the effect of variation in restore yield caused by increase in PTM's R_{MET} is overcome by the increased RRAM's R_{HRS}. Read yield on the other hand is a strong function of V_{IMT} and V_{MIT}. Read yield reduction in this analysis takes into account the effect of RRAM not restoring the values correctly and the SRAM read failures as well. For small values of V_{MIT}, PTM does not transition into insulating state post restore, thus causing the PTM to interfere with the SRAM read, resulting in low read yield. As the V_{MIT} increases, there is not enough differential between VL and VR to restore the contents,' as the PTM on the side where the RRAM is in LRS state has the risk of undergoing MIT transition before developing sufficient voltage difference between the VL and VR nodes. This further results in reduction of read yield (Fig.10e).

IX. Layout Studies

Layout studies were performed and the effect of additional RC due to the integration of 2R-2S have already been included in the simulations presented in the paper. Here we provide a summary of the approach to design and study the layouts. Layout of the baseline SRAM and proposed 6T-2R-2S are implemented in Cadence Virtuoso design environment using 7nm open source predictive technology PDK ASAP7. [18].

X. Conclusion

This paper makes use of the 6T bitcell with the addition of RRAM and PTM as technology assist to reduce the V_{MIN} of the bitcell, proposed in [11]. In this paper, we explore NVM, SRAM-RRAM hybrid modes of operation. 2T-2R-2S/2T-2R mode is realized by exploiting only the differential NVM portion of the bitcell. Furthermore, this paper analyzes the feasibility of performing the above-mentioned operations in case of 8T2R bitcell as well, which uses the additional transistors instead of using selector devices. SRAM-RRAM hybrid mode makes use of SRAM's read port access transistor to fasten the read time. The design space exploration with respect to different parameters of PTM and RRAM is studied in the case of NVM and SRAM-RRAM hybrid mode. As part of circuit-system co-design, the performance of the 6T-2R-2S bitcell is measured in terms of a 8KB set-associative cache. Furthermore, the 2T-2R-2S/2T-2R mode coupled with SRAM only mode is used to perform in-situ compute in memory application for performing dot product. XNOR operation, widely used in performing Multiply-and-accumulate operation in binarized neural networks is presented using the 2T-2R-2S and SRAM only mode of operation. The proposed CIM design is further tested using CIFAR-10 dataset on custom CNN and this shows an accuracy of 76% as opposed to software accuracy of 82%. This is followed by a system level study analyzing the performance of the proposed design in terms of speedup and energy efficiency in contrast to the baseline design. The results indicate a speedup of 3x is possible, assuming best case scenario for the baseline design of 1T1R based CIM architecture.

References

[1] E. Morifuji, T. Yoshida, M. Kanda, S. Matsuda, S. Yamada, and F. Matsuoka, "Supply and threshold-voltage trends for scaled logic and SRAM MOSFETs," *IEEE Trans. Electron Devices*, vol. 53, no. 6, pp. 1427–1432, Jun. 2006.

[2] F. Hamzaoglu *et al.*, "Dual-V_T SRAM cells with full-swing single-ended bit line sensing for high-performance on-chip cache in 0.13 μm technology generation," in *Proc. Int. Symp. Low Power Electron. Design*, 2000, pp. 15–19.

[3] W. Wang *et al.*, "Nonvolatile SRAM cell," in *IEDM Tech. Dig.*, Dec. 2006, pp. 1–4.

[4] A. Lee *et al.*, "RRAM-based 7T1R nonvolatile SRAM with 2x reduction in store energy and 94x reduction in restore energy for frequent-off instant-on applications," in *Proc. Symp. VLSI Technol. (VLSI Technology)*, Jun. 2015, pp. C76–C77.

[5] P.-F. Chiu *et al.*, "A low store energy, low VDDmin, nonvolatile 8T2R SRAM with 3D stacked RRAM devices for low power mobile applications," in *Proc. Symp. VLSI Circuits*, Jun. 2010, pp. 229–230.

[6] P. Kolar *et al.*, "A 32 nm high-K metal gate SRAM with adaptive dynamic stability enhancement for low-voltage operation," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 76–84, Jan. 2011.

[7] J. Chang *et al.*, "12.1 A 7 nm 256 Mb SRAM in high-K metal-gate FinFET technology with write-assist circuitry for low-V_{MIN} applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 206–207.

[8] Y. Wang *et al.*, "Dynamic behavior of SRAM data retention and a novel transient voltage collapse technique for 0.6 V 32 nm LP SRAM," in *IEDM Tech. Dig.*, Dec. 2011, p. 32.

[9] S. Yamamoto, Y. Shuto, and S. Sugahara, "Nonvolatile SRAM (NV-SRAM) using functional MOSFET merged with resistive switching devices," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2009, pp. 531–534.

[10] S.-S. Sheu *et al.*, "A ReRAM integrated 7T2R non-volatile SRAM for normally-off computing application," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2013, pp. 245–248.

[11] S. S. Teja Nibhanupudi and J. P. Kulkarni, "High density NV-SRAM using memristor and selector as technology assist," in *Proc. Int. Symp. VLSI Technol., Syst. Appl. (VLSI-TSA)*, Apr. 2019, pp. 1–2.

[12] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," 2016, *arXiv:1602.02830*.

[13] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, and H.-S.-P. Wong, "Verilog–A compact model for oxide-based resistive random access memory (RRAM)," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices (SISPAD)*, Sep. 2014, pp. 41–44.

[14] S. S. T. Nibhanupudi, S. R. S. Raman, and J. P. Kulkarni, "Phase transition material-assisted low-power SRAM design," *IEEE Trans. Electron Devices*, vol. 68, no. 5, pp. 2281–2288, May 2021.

[15] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, Jul. 2012.

[16] S. Xie, C. Ni, A. Sayal, P. Jain, F. Hamzaoglu, and J. P. Kulkarni, "16.2 eDRAM-CIM: Compute-in-memory design with reconfigurable embedded-dynamic-memory array realizing adaptive data converters and charge-domain computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 248–250.

[17] A. Boroumand *et al.*, "Google workloads for consumer devices: Mitigating data movement bottlenecks," *ACM SIGPLAN Notices*, vol. 53, pp. 316–331, Mar. 2018.

[18] L. T. Clark *et al.*, "ASAP7: A 7-nm FinFET predictive process design kit," *Microelectron. J.*, vol. 53, pp. 105–115, Jul. 2016.