Referencing-in-Array Scheme for RRAM-based CIM Architecture

Singh, Abhairaj; Bishnoi, Rajendra; Joshi, Rajiv V.; Hamdioui, Said

DOI
10.23919/DATE54114.2022.9774571

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 2022 Design, Automation and Test in Europe Conference and Exhibition, DATE 2022

Citation (APA)
Singh, A., Bishnoi, R., Joshi, R. V., & Hamdioui, S. (2022). Referencing-in-Array Scheme for RRAM-based CIM Architecture. In C. Bolchini, I. Verbauwhede, & I. Vatajelu (Eds.), *Proceedings of the 2022 Design, Automation and Test in Europe Conference and Exhibition, DATE 2022* (pp. 1413-1418). Article 9774571 IEEE. https://doi.org/10.23919/DATE54114.2022.9774571

Referencing-in-Array Scheme for RRAM-based CIM Architecture

Abhairaj Singh[1], Rajendra Bishnoi[1], Rajiv V. Joshi[2] and Said Hamdioui[1]

[1]Computer Engineering Laboratory, TU Delft, The Netherlands: (a.singh-5, r.k.bishnoi, s.hamdioui)@tudelft.nl
[2]IBM Thomas J. Watson Research Centre, Yorktown Heights, NY 10598 USA: rvjoshi@ibm.us

***Abstract*—Resistive random access memory (RRAM) based computation-in-memory (CIM) architectures are attracting a lot of attention due to their potential in performing fast and energy-efficient computing. However, the RRAM variability and non-idealities limit the computing accuracy of such architectures, especially for multi-operand logic operations. This paper proposes a voltage-based differential referencing-in-array scheme that enables accurate two and multi-operand logic operations for RRAM-based CIM architecture. The scheme makes use of a 2T2R cell configuration to create a complementary bitcell structure that inherently acts also as a reference during the operation execution; this results in a high sensing margin. Moreover, the variation-sensitive multi-operand (N)AND operation is implemented using complementary-input (N)OR operation to further improve its accuracy. Simulation results for a post-layout extracted 512x512 (256Kb) RRAM-based CIM array show that up to 56 operand (N)OR/(N)AND operation can be accurately and reliably performed as opposed to a maximum of 4 operands supported by state-of-the-art solutions, while offering up to 11.4X better energy-efficiency.**

I. Introduction

Resistive random access memory (RRAM)-based computation-in-memory (CIM) architectures have been vastly explored to perform boolean binary logic (BBL) operations for applications related e.g., to artificial intelligence and big data in an energy-efficient manner [1]. However, RRAM devices suffer from non-idealities such as variations, resistance drift and read disturb [2, 3]. These limitations, along with CMOS variations and wire parasitics, lead to inaccurate, unreliable and energy-inefficient CIM-based BBL operations. Therefore, energy-efficient circuit solutions that realize accurate and reliable CIM-based BBL operations are required.

Several works on CIM-based BBL operations have been reported, but they weakly address the aforementioned challenges. Such works can be classified into two classes: 1) non-stateful or read-assisted logic- where the RRAM state is unaltered. 2) stateful logic- where the RRAM state is altered in order to perform BBL operations. Most of the non-stateful architectures are based on single-ended sensing mechanism where multi-row (operands) read operations are performed using dedicated reference signals [4–8]; the signals enable the selection of the BBL operations to be performed (e.g., (N)AND, (N)OR) and associated number of operands. However, this results in complex periphery circuitry with limitations. First, the sense amplifier (SA) employed incur more area and consume more energy as they need to generate several reference schemes controlled using multi-MUXs. Second, due to the static nature of their referencing schemes, these architectures have a small read margin, especially in the presence of process variation and wire parasitics; this reduces the computing accuracy and limits the maximum number of operands. Third, (N)OR and highly sensitive (N)AND operations are executed in a similar way, implying that the (N)AND operation becomes the bottleneck which significantly impacts the overall performance as well as limits the scalability of such CIM architectures [9, 10]. On the other hand, few non-stateful logic solutions have adopted differential sensing scheme [11–14]; however, the use of external references implies that they face the aforementioned issues as well. Other stateful logic solutions require programming of RRAM devices [15–19]; they result into a high energy consumption as well as reliability and endurance issues. In short, there still a need of cost effective circuit-level solutions that does not only provide accurate and energy-efficient BBL operations, but also push the limit of the maximum allowed number of operands.

This paper proposes a reference-in-array circuit-level scheme that accurately and reliably perform BBL operations with up to 56 operands in a single cycle, while realizing up to 11.4X energy-efficiency compared to state-of-the-art CIM-based BBL solutions. The key contributions of the paper are:

- Introduces a differential sensing scheme by arranging cells in a complementary structure and a row of dummy bitcells in such a way that it inherently acts as a reference to enable a high sensing margin.
- Implements highly variation-sensitive (N)AND operation using complementary-input (N)OR operation to further enhance the accuracy and energy-efficiency.
- Presents two design-for-reliability techniques to accurately perform up to 10^8 consecutive BBL operations on the same operands or data-bits.
- Reports simulation results and comprehensive comparison using a post-layout extracted 512x512 (256Kb) RRAM-based CIM array.

The rest of the paper is organized as follows. Section II covers the fundamentals and challenges related to BBL operations using RRAM-based CIM. Section III presents the proposed scheme, followed by results in Section IV. Finally,

This work was supported in part by the EU H2020 grant "DAIS" that has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007273.

978-3-9819263-6-1/DATE22/©2022 EDAA

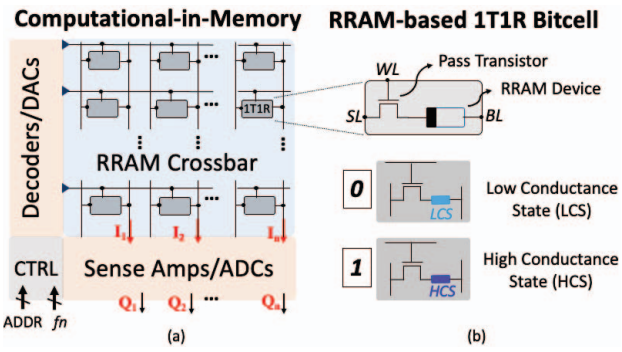


Fig. 1: (a) CIM architecture (b) Typical RRAM-based bitcell design.

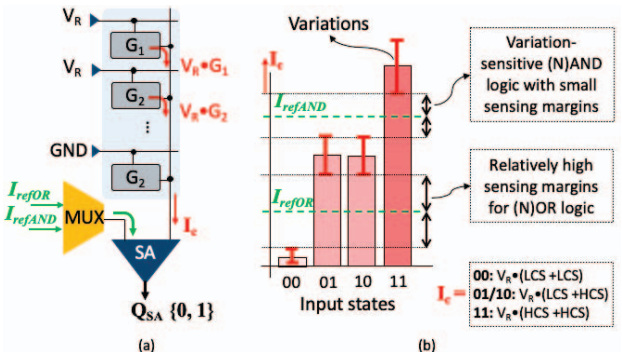


Fig. 2: (a) Working of (N)AND, (N)OR operation; (b) Comparison of the relative sensing margins of (N)AND and (N)OR logic operation.

Section V concludes the paper.

II. Background

A. RRAM-based CIM architecture

CIM architecture performs computation within the memory. RRAM-based CIM architecture is shown in Fig. 1a. CIM comprises of RRAM-based crossbar memory, address decoders and periphery circuitry to support computation within the memory. CIM performs BBL operation 'f_n' on the operands with addresses '$ADDR$' using row drivers and customized SAs in the periphery; note that these are replaced by digital-to-analog (DACs) and analog-to-digital converters (ADCs), respectively, for analog multiply-and-accumulate operations. Fig. 1b shows the RRAM-based one-transistor-one-resistor (1T1R) bitcell. An RRAM device is based on the reversible formation of a conductive filament delivering low (LCS) and high conductance states (HCS) [3]. Data-bits '0' and '1' are represented as LCS and HCS of the RRAM, respectively.

B. BBL operations and their challenges

CIM-based BBL operations are illustrated in Fig 2(a). The underlying concepts for read-assist logic designs are: 1) The interaction of voltage V_R and bitcell conductance G, in accordance with Ohm's law, resulting in the current $V_R \times G$ per bitcell; 2) Accumulation of these currents into a column current I_c, in accordance with Kirchhoff's current law; and 3) comparison of the current or voltage drop with an appropriate reference (selected by a MUX) using a customized SA. Therefore, BBL operations can be performed simultaneously in all the activated columns and practically operate at O(1) time complexity, thereby achieving massive parallelism.

Read-assisted BBL operations suffer from the presence of variations which affect the overall performance and computational accuracy. The conventional approach is based on a single-sided sensing mechanism, implying that array current/voltage output is compared against a reference which offers small sensing margin for computation [4–9]. Additionally, as shown in Fig. 2(b), the inherent small margin associated with (N)AND operation as compared to (N)OR operation further limits the scalability in terms of operand size [6, 9]. Moreover, the crossbar size scalability is limited due to the influence of wire parasitics that further degrades the performance and the accuracy of the operations. On top of that, an RRAM device also suffers from the accumulated effect of large number of read operations that can lead to significant conductance change (conductance drift) or unwanted bit-flip (read disturb) [3, 20]. Existing solutions to deal with all these challenges are mainly based on large and complex SAs and reference generators; however they degrade the overall efficiency of the computing [4–9, 21–24]. Supporting multi-operand operations further aggravates the above aforementioned challenges [8, 21]. Multi-operand can alternatively be performed by investing in aggregating multiple two-operand (N)AND or (N)OR operating cycles effectively in the periphery [9]. This, however, requires complex periphery circuits implying additional area and power consumption, and reduced computational throughput.

III. Proposed CIM Accelerator

Here, we first present an overview of our proposed scheme, followed by its implementation details.

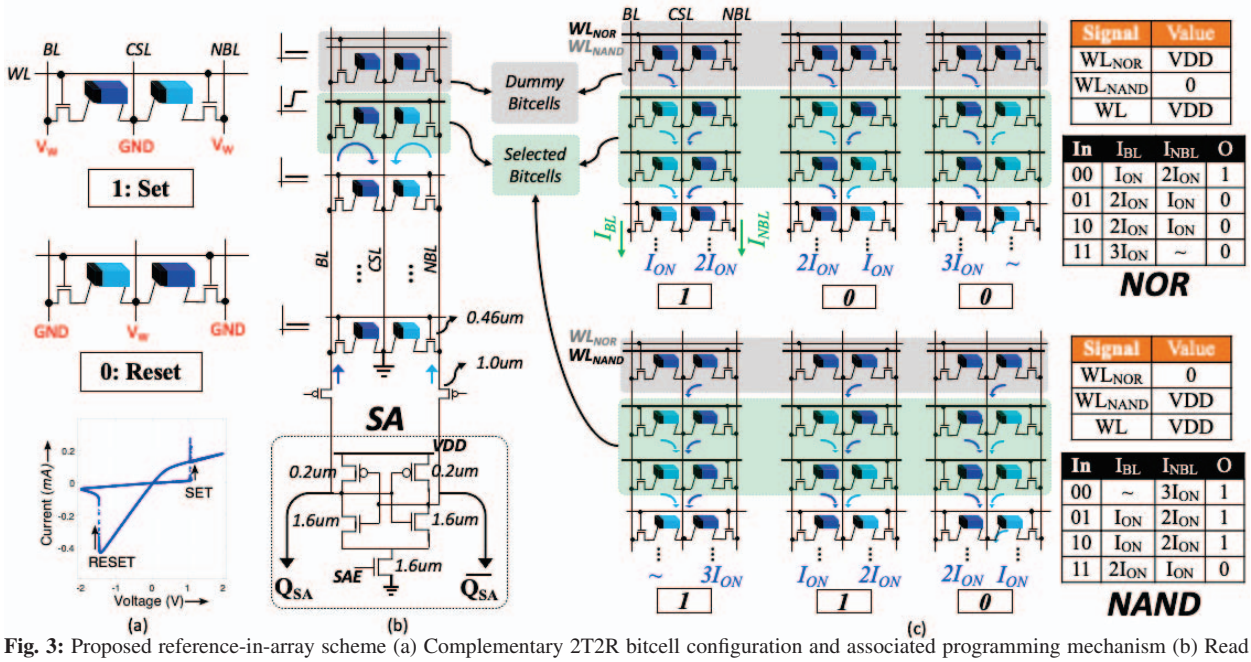
A. Overview

We use a bitcell design which has two transistors and two RRAM devices (2T2R), arranged in such a way that the two RRAM devices always have the opposite values as shown in Fig. 3(a); the complementary cell structure enables the use of differential sensing, resulting in a high sense margin (∼2x compared to a single-ended sensing). We also realize NAND operations using NOR logic design by exploiting the De-Morgan's law[1]; this completely eliminates the sensing bottleneck of NAND operations. We introduce an extra dummy cell per column that acts a reference to perform BBL operations in a differential manner. The resulting scheme does not only deal with the process variation naturally, but it also simplifies the overall sensing mechanism as our scheme does not need any external reference signals, nor any other controlling circuits.

B. Proposed CIM Architecture

Next the different aspects of the proposed architectures are explained.

[1]De-Morgan's law states that the NAND gate is equivalent to an OR gate with inverted inputs



Signal	Value
WL_{NOR}	VDD
WL_{NAND}	0
WL	VDD

In	I_{BL}	I_{NBL}	O
00	I_{ON}	$2I_{ON}$	1
01	$2I_{ON}$	I_{ON}	0
10	$2I_{ON}$	I_{ON}	0
11	$3I_{ON}$	~	0

Signal	Value
WL_{NOR}	0
WL_{NAND}	VDD
WL	VDD

In	I_{BL}	I_{NBL}	O
00	~	$3I_{ON}$	1
01	I_{ON}	$2I_{ON}$	1
10	I_{ON}	$2I_{ON}$	1
11	$2I_{ON}$	I_{ON}	0

Fig. 3: Proposed reference-in-array scheme (a) Complementary 2T2R bitcell configuration and associated programming mechanism (b) Read operation performed in a column within the CIM core. (c) Working of the proposed two-operand NOR and NAND logic designs.

1) Cell structure: The proposed architecture is based on 2T2R bitcell configuration as shown in Fig. 3(a). Data-bit '1' is represented by dark blue (HCS) and the complementary data-bit as light blue (LCS), and the pass transistors connect these RRAMs to bitline (BL) and negative bitline (NBL). The bitcell has a wordline (WL) and a common select line (CSL), connecting the top electrode of one RRAM to the bottom electrode of the other. Fig. 3(a) shows that programming such a bitcell is not different from programming a 1T1R bitcell, where CSL is connected to GND and both bitlines are supplied with the write voltages (V_W) for SET and vice-versa for RESET. Also, the figure shows the current-voltage (IV) characteristics of an RRAM device.

2) Memory single read operations: Fig 3(b) shows how a read operation is performed. The bitline pair is pre-charged to voltage supply (VDD) and CSL is connected to ground (GND) before the start of the cycle. WL activation enables the bitlines to discharge through complementary RRAM devices, creating a differential BL/NBL voltage $(\Delta V{=}V_{BL}{-}V_{NBL})$ which is sensed using a SA. We use cross-coupled SA that involves positive feedback loop to amplify the input differential voltage.

3) Two-operand BBL operations: Fig. 3(c) illustrates logic operations performed using a dummy row of 2T2R bitcells; the top part of the figure illustrates the NOR operation and the bottom part the NAND operation. The idea is to bias the BL(NBL) side with an ON current while performing NOR(NAND) operation. The dummy cells require two modifications; they need a) to have both RRAM devices of the cell in the HCS, and 2) to get two independent WLs, namely WL_{NOR} and WL_{NAND}, connected to the BL and NBL-sided pass transistors, respectively. The one-time configuration of dummy cells requires individual RRAM device programming to HCS by selecting the dedicated WLs one at a time. To perform a NOR function, two rows storing operands are activated along with WL_{NOR} in the dummy row. For simplicity, we assume ON current to be I_{ON}, and OFF current to be zero. In case both operands are in the RESET state (00); the BL/NBL discharge currents are $I_{ON}/2I_{ON}$, resulting in '1' as NOR result. In case the two operands are in different states (01/10); the BL/NBL discharge currents are $2I_{ON}/I_{ON}$, implying a value '0'. In case both operands are in the SET state (11); the BL/NBL discharge currents are $3I_{ON}/0$, leading to a value '0'. In a similar manner, the distribution of BL/NBL currents for NAND can be derived for each of the above cases; the results are shown in the table included in the bottom part of Fig. 3(c).

Fig. 4a shows the simulation results of our two-operand NAND and compares it with the conventional approach based on 1T1R and sensing mechanism with a fixed reference[4–6]. In the figure, WL_{conv} represents the array wordline timing used for the conventional case; while WL the wordline behavior of one of accessed cells (operands), and WL_{NAND} the wordline of the dummy cell of our scheme (see Fig. 3(b)). The operating cycle time depends on the time needed to develop the required ΔV on the bitline pair (assuming $\Delta V{=}40mV$) to accurately differentiate the operand states. For the conventional scheme, the figure shows the BL discharge for the cases 01 and 11, as well as the reference (BL_{ref}); note that a total of $\Delta V{=}80mV$ is generated between the 01 and 11 states. On the other hand, our proposal (making use of differential self-referencing scheme) takes much less time to generate the required $\Delta V{=}40mV$ between BL and NBL; hence, enabling not only wide margins but also fast operations.

4) Multi-operand BBL operations: Fig. 5(a) illustrates the multi-operand NOR and NAND operations. Each 2T2R array bitcell has now two independent WLs; one used for NOR operations and act on data (WL_D), one used for NAND op-

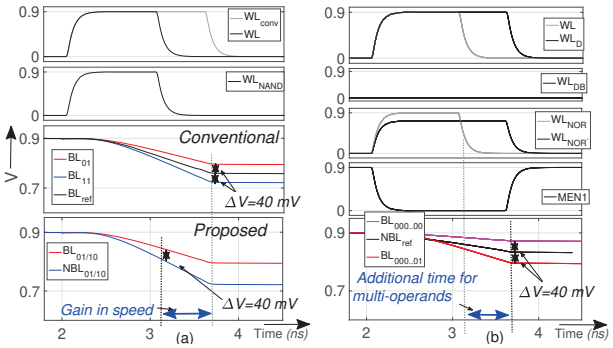
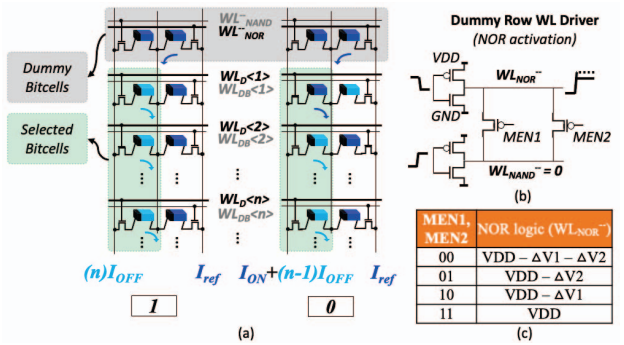


Fig. 4: Timing diagrams (a) to illustrate performance gain for two-operand NAND operation (b) multi-operand NOR operation.



MEN1, MEN2	NOR logic (WL_{NOR^-})
00	VDD – ΔV1 – ΔV2
01	VDD – ΔV2
10	VDD – ΔV1
11	VDD

Fig. 5: (a) Proposed multi-operand NOR operation. (b) Modified dummy WL driver to degrade WL_{NOR^-} or $WL_{NAND\_}$ signal, and (c) Configurable MEN1, MEN2 switches and associated WL_{NOR^-}.

erations and act on complementary of data (WL_{DB}). During the NOR multi-operand operation, BL is discharged through the RRAM devices on the left side of the column (storing data), while during the NAND operation, NBL is discharged through the RRAM devices on the right side of the column (storing complement data). The dummy bitcell (per column) is used to enable the use of differential SA; it provides the reference current I_{ref}, and it has also two separate WLs; one selected during NOR operation and one during NAND operation. During the NOR operation, the BL/NBL = BL/I_{ref} discharge currents depend on the state of the selected operands. The value of I_{ref} has to guarantee the correct NOR operation for all states including the states with minimum difference in the BL discharge currents; these states are "all RESET (0)" (resulting in a BL discharge current of $n * I_{OFF}$) and "one SET (1) and $n-1$ RESET" (resulting in a BL discharge current of I_{ON}+$(n-1)$*I_{OFF}). Hence, I_{ref} has to be ideally at the middle of these two discharging currents; if we assume that $I_{ON} >> I_{OFF}$, then ideally I_{ref} should be about $I_{ON}/2$.

The dummy bitcell is used to generate I_{ref} ; it is controlled with a dummy wordline WL_{NOR^-} driven by reduced voltage levels. Such a voltage can be generated using the PMOS-based bleeder circuit shown in Fig. 5(b). The PMOS devices enabled by MEN1 and MEN2 are used to provide the flexibility of generating different values of WL_{NOR^-} (hence of I_{ref}). Therefore, some calibration can be done if needed. The configuration of Fig. 5(b) allows the generation of three reduced voltage levels.

Similar analogy like the above can be used for multi-operand NAND. In this case, the WL_{NAND^-} should be driven with a reduced voltage ensuring the discharging current of NBL= I_{ref}, which has to be around $I_{ON}/2$ as well. During this operation, the difference in discharge currents BL/NBL= I_{ref}/NBL will ensure the correct operation through the SA.

Fig. 4b illustrates the working of 10-operand NOR operation and shows that additional time it required, as compared to the two-operand NOR operation of Fig. 4a. In the figure the same signal naming is used as that of Fig. 5(a). To show the comparison with two operand NOR operation, WL (representing one of the accessed bitcell wordlines) and WL_{NOR} (representing the dummy row wordline) for this two-operand operation are included. The results are shown for two cases resulting in minimum discharge of the BL: a) all the operands are set to 0 (00..00), b) all operands as set to 0 except one operand is set to 1 (00..01). To reach the required minimum ΔV=$40mV$ between BL and NBL_{ref}, the duration of WL_D should be extended to give enough time for BL discharging though dummy cell selected by WL_{NOR^-}. Hence, increasing the number of operands to 10 while realizing robust operation comes at the cost of additional latency of 40%.

To ensure accurate execution of multi-operand operations in large-sized crossbars, both the *number* of operands and the *RC parasitic* of the wordlines have to be taken into consideration. The higher the number of operands, the smaller the sensing margin for a fixed wordline duration; the farther the bitcell accessed by the wordline signal, the larger the wordline delay and voltage degradation reaching that bitcell, and hence, the slower the corresponding bitline discharge. The cumulative effect is the degradation of signal margins.

Fig. 6 shows the worst case sensing margins ΔV developed for multi-operand NAND operation (for conventional scheme as well as for our proposed scheme) operating at a fixed wordline duration where the operations are performed for different columns (from 32 to 512), and by selecting the appropriate values for the operands resulting the the minimum sensing margin. The notation $m(opr)$ denotes m-multi operand operation where opr operands are set to 1. E.g., 10(10) denotes the 10 multi-operand operation where all operands are set to 1. For the conventional approach, $3 \leq m \leq 10$, and the the interleaved horizontal lines are the corresponding reference signals for which changes depending on m. It can be seen that the reference signals in the conventional scheme fail to differentiate the operand states beyond a certain number operands ($m > 4$) and/or beyond a certain column (which is m dependent). Hence, in these cases, the multi-operand operation fails (shown in light red color in the figure). On the other hand in our proposed scheme, the reference is fixed irrespective of the number of operands; it is set to the middle of discharging voltages of two cases: 10(0) and 10(1), meaning 10 multi-operand NAND operation where none of the operands is set to 1 (i.e., all in RESET) and only one operand is set to 1, respectively. Clearly our proposal enables large m even beyond 10. Moreover, as the dummy row generating the references experiences similar degradation as the accessed operands, this can reliably provide undeterred sensing margins.

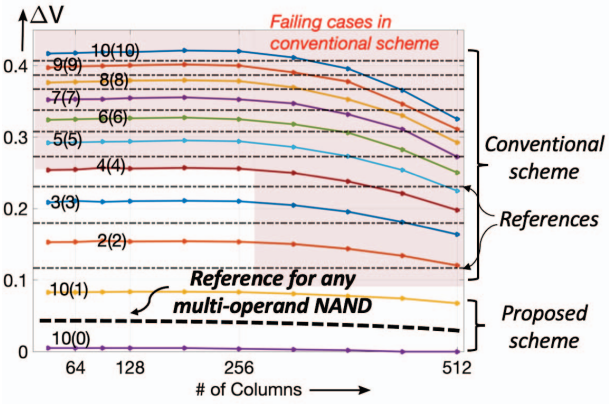


Fig. 6: (a) Reduced signal margins due to operand size increase and RC parasitics. Proposed scheme to circumvent the erroneous region.

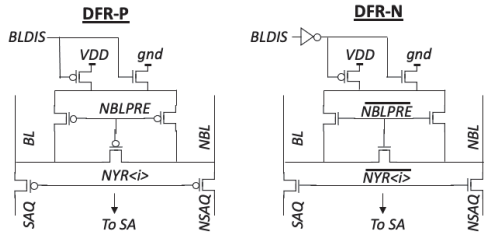


Fig. 7: Design for reliability based on PMOS and NMOS device.

In summary, significant operand scalability improvement for BBL operations is achieved.

C. Design Optimizations

RRAM devices suffer from device degradation, resistance drift and eventual unwanted bit-flip. These are accelerated when the RRAMs are under voltage stress; the higher the voltage, the faster the degradation [3]. To suppress and/or to slow the degradation, one can reduce the voltage of the bit line precharge from VDD to a lower level, resulting in small voltage ($V_{BL/NBL} - V_{CSL}$) across RRAM devices (see Fig. 3b). Fig. 7 presents two possible circuit solutions to generate such reduced precharge voltage using VDD voltage supply: one based on a PMOS device (DFR-P) and one based on NMOS device (DFR-N). For DFR-P, the bitline pair is precharged to VDD, then discharged through PMOS to VSS+VthP; while for DFR-N the bitline pair is pre-discharged to VSS, then charged through NMOS to VSS-VthN (VthP and VthN are PMOS and NMOS voltage thresholds, respectively). For both designs, BLDIS signal remains active while keeping NBLPRE=0 for a configurable amount of time to ensures equal voltages at the bitline pair before the start of the active cycle. NYR signal connects the bitline pair to the SA.

IV. Simulation Setup & Results

A. Setup

Table I summarizes the design specifications used for our circuit-level analysis. SA is designed to accurately sense a minimum ΔV of 40mV. Fig. 8 shows the layout view of the HfO_2/TiO_x RRAM-based conventional 1T1R bitcells and our 2T2R bitcell. Pass transistor (NMOS) is 540 nm/40 nm

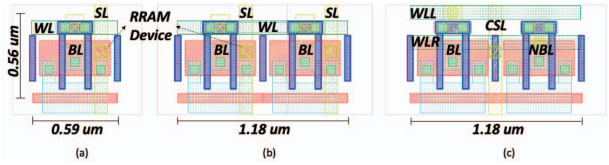


Fig. 8: Layout footprint for (a) conventional 1T1R bitcell (b) two 1T1R bitcells and (c) 2T2R bitcell in our proposed scheme.

and it typically provides a resistance of 1.3 $K\Omega$. Vertical (Horizontal) wire resistance adds up to 0.4 Ω (0.8 Ω) and total capacitance adds up to 0.3 fF (0.6 fF) per unit bitcell.

The simulation and comparisons with the conventional schemes are extracted using the same setup; e.g., technology node, SA design, RRAM bitcell, wire parasitics. In this way, all penalties are considered in terms of latency (power is assumed to be nearly the same).

B. Circuit-level Results and Comparisons

Fig. 9a shows the way the bitline is discharged for a m-operand NAND operation for $2 \leq m \leq 9$ where they all are in SET state (worst case); ΔV=40mV is the minimum sensing margin required by the SA (the reference lines are not included in the figure for clarity). Note that reaching the required sensing margin becomes impossible after 4 operands irrespective of additional discharge time. Hence, conventional schemes can support only a limited number of operands.

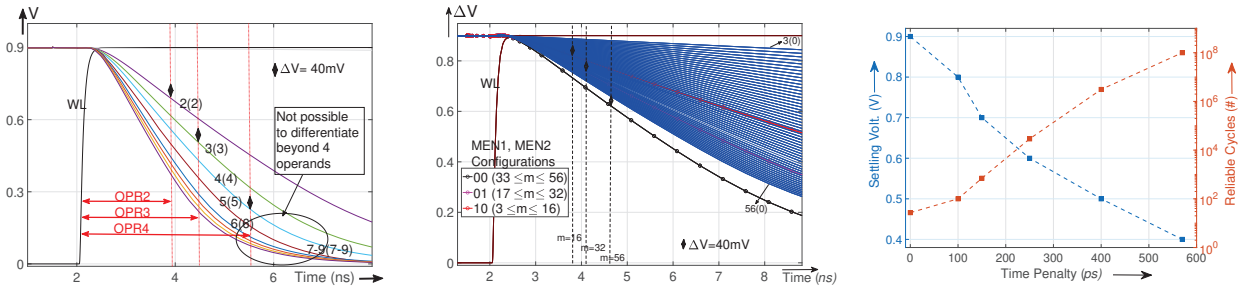
Fig. 9b illustrates how our proposed scheme can make use of signals MEN1 and MEN2 (see Fig. 5b) to better configure I_{ref} in order to maximize the number of operand m for NAND operation; the three configurable reference signals ensure $\Delta V = 40mV$ signal margin for the worst case scenarios; for our scheme, this is the case when all operands are 0 (RESET), which is the opposite of the conventional approach. The figure shows the discharge of the bitline for $3 \leq m \leq 52$; the wordline is activated for a long duration to capture different timing requirements to perform operations up to 56 operands. Note that the maximum achievable m is limited by the fact that a RESET state has a finite bitline current. Therefore, a higher HCS/LCS ratio can enable more number of operands.

Fig. 9c shows the significant impact of using DFR-P of Fig. 7 in reducing the conductance drift and the probability of eventual bit-flip during consecutive multi-operand BBL operations; this comes at the cost of timing penalty which depends on the targeted BL precharge (settling) voltage. The more timing penalty we tolerate, the slower is the RRAM device degradation and the more consecutive reliable operations.

Table II summaries the results and shows the comparison with the state-of-the art for different metrics. It is worth noting

Parameters	Specifications
CIM Array	512x512 (256Kb)
RRAM Device	HfO_2/TiO_x [3]
HCS/LCS	100 $K\Omega$ / 3 $K\Omega$
RRAM Variation	20 % parametric
Voltage supply	0.9 V with ±10 % variations
CMOS (variations)	SVT, 40 nm TSMC (3 σ)
Temperature	-40°C to 125°C

TABLE I: Design parameters.



(a) **(b)** **(c)**

Fig. 9: (a) BL discharge behavior for increasing number of NAND operands for conventional scheme (b) Multi-operand operations up to 56 operands can be performed by configuring MEN1 and MEN2 signals accordingly. (c) Reliable read cycles due to settling BL/NBL voltages associated with timing penalty using DFR-P design.

that although our 2T2R bitcell takes 2X area compared to the 1T1R bitcell, the complete CIM core of similar capacity is estimated to have 1.67X area owing to our smaller periphery.

Design Metrics	Conv. [4–8]	Cascade [9]	Proposed *(Standard)*	Proposed *(DFR-P)*
Max. Ops/cycle	4	2	56	17
Latency (ns)*	4.2/4.2	4.4/59.2	2.1/5.2	2.7/5.8
Energy (pJ)*	0.56/0.56	0.58/18.4	0.29/1.62	0.32/1.67
Energy/Op (fJ)*	140/140	145/380	72/29	80/29.8
Voltage supplies	2	2	1	1
Read Disturb Tol.	No	No	No	Yes
Variation Tol.	No	No	Yes	Yes

TABLE II: Comparison of our proposed design with prior techniques. * indicates *four*/maximum operands supported. *Standard* and *DFR-P* are non-optimized and optimized designs for read disturb.

V. Conclusion

This paper has demonstrated how adding a dummy row in a computation-in-memory (CIM) crossbar based on RRAM can enable robust and reliable multi-operand bit wise logic operation. It does not only make the generation of references much simpler, but it also enables wider sensing margins. In addition, the paper has shown how some basic circuits can be integrated with CIM to suppress/slow the degradation and boost the robustness of CIM logic operations while trading off some latency. Comparison results using 512x512 CIM core shows that our proposed design offers an improvement of upto 11.4X in terms of energy-efficiency while performing up to 56 operands in a single cycle.

References

[1] M. A. Lebdeh *et al.*, "Memristive device based circuits for computation-in-memory architectures," in *ISCAS*, 2019.

[2] A. Singh *et al.*, "Low-power memristor-based computing for edge-ai applications," in *ISCAS*, 2021, pp. 1–5.

[3] W. Kim *et al.*, "Multistate memristive tantalum oxide devices for ternary arithmetic," *Scientific reports, 2016*,

[4] S. Li *et al.*, "Pinatubo: Processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *DAC*, 2016.

[5] L. Xie *et al.*, "Scouting logic: A novel memristor-based logic design for resistive computing," in *ISVLSI*, 2017.

[6] S. Jain *et al.*, "Computing in memory with spin-transfer torque magnetic RAM," *IEEE Transactions on VLSI Systems*, 2017.

[7] M. Imani *et al.*, "MPIM: Multi-purpose in-memory processing using configurable resistive memory," in *ASP-DAC*, 2017.

[8] M. Lee *et al.*, "FeFET-based low-power bitwise logic-in-memory with direct write-back and data-adaptive dynamic sensing interface," in *International Symposium on Low Power Electronics*, 2020.

[9] I. Giannopoulos *et al.*, "In-Memory Database Query," *Advanced Intelligent Systems*, vol. 2, no. 12, p. 2 000 141, 2020.

[10] P. C. Santos *et al.*, "Operand size reconfiguration for big data processing in memory," in *DATE, 2017*.

[11] M. Bocquet *et al.*, "In-memory and error-immune differential RRAM implementation of binarized deep neural networks," in *IEDM*, 2018.

[12] L. Wang *et al.*, "Efficient and robust nonvolatile computing-in-memory based on voltage division in 2T2R RRAM with input-dependent sensing control," *TCAS II: Express Briefs, 2021*,

[13] J. Wang *et al.*, "A novel page-forming scheme with ultra-low bit-error-rate and high reliability on a 1Mb RRAM chip," in *ICSICT, 2020*.

[14] Z. Zhou *et al.*, "A new hardware implementation approach of BNNs based on nonlinear 2T2R synaptic cell," in *IEDM, 2018*.

[15] G. Snider, "Computing with hysteretic resistor crossbars," *Applied Physics A*, vol. 80, no. 6, pp. 1165–1172, 2005.

[16] J. Borghetti *et al.*, "Memristive switches enable 'stateful' logic operations via material implication," *Nature, 2010*,

[17] S. Kvatinsky *et al.*, "MAGIC—memristor-aided logic," *TCAS II: Express Briefs, 2014*,

[18] L. Xie *et al.*, "Fast boolean logic mapped on memristor crossbar," in *ICCD, 2015*.

[19] P.-E. Gaillardon *et al.*, "The programmable logic-in-memory (PLiM) computer," in *DATE, 2016*.

[20] M.-F. Chang *et al.*, "Circuit design challenges in embedded memory and resistive RAM (RRAM) for mobile SoC and 3D-IC," in *16th ASP-DAC*, IEEE, 2011, pp. 197–203.

[21] W.-H. Chen *et al.*, "CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors," *Nature, 2019*,

[22] H. Koike *et al.*, "1T1MTJ STT-MRAM cell array design with an adaptive reference voltage generator for improving device variation tolerance," in *IMW, 2015*.

[23] Y. Xie *et al.*, "A logic resistive memory chip for embedded key storage with physical security," *TCAS II: Express Briefs, 2015*,

[24] D. Ly *et al.*, "Novel 1T2R1T RRAM-based ternary content addressable memory for large scale pattern recognition," in *IEDM, 2019*.