IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 March 2024, accepted 28 March 2024, date of publication 3 April 2024, date of current version 11 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384752

RESEARCH ARTICLE

CMOS-Based Single-Cycle In-Memory XOR/XNOR

SHAMIUL ALAM[1], (Graduate Student Member, IEEE), JACK HUTCHINS[1], NIKHIL SHUKLA[2], (Member, IEEE), KAZI ASIFUZZAMAN[3], (Member, IEEE), AND AHMEDULLAH AZIZ[1], (Senior Member, IEEE)

[1]Department of Electrical Engineering and Computer Science, University of Tennessee Knoxville, Knoxville, TN 37996, USA
[2]Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA
[3]Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

Corresponding author: Ahmedullah Aziz (aziz@utk.edu)

This work was supported in part by seed funding from the AI Tennessee Initiative, University of Tennessee Knoxville; and in part by the U.S. Department of Energy, UT-Battelle LLC, under Contract DE-AC05-00OR22725. The work of Shamiul Alam was supported by the Science Alliance, a Tennessee Higher Education Commission Center of excellence administered by The University of Tennessee-Oak Ridge Innovation Institute on behalf of The University of Tennessee Knoxville. The work of Nikhil Shukla was supported in part by the National Science Foundation under Grant 2132918.

ABSTRACT Big data applications are on the rise, and so is the number of data centers. The ever-increasing massive data pool needs to be periodically backed up in a secure environment. Moreover, a massive amount of securely backed-up data is required for training binary convolutional neural networks for image classification. XOR and XNOR operations are essential for large-scale data copy verification, encryption, and classification algorithms. The disproportionate speed of existing compute and memory units makes the *von Neumann* architecture inefficient to perform these Boolean operations. Compute-in-memory (CiM) has proved to be an optimum approach for such bulk computations. The existing CiM-based XOR/XNOR techniques either require multiple cycles for computing or add to the complexity of the fabrication process. Here, we propose a CMOS-based hardware topology for single-cycle in-memory XOR/XNOR operations. Our design provides at least 2× improvement in the latency compared with other existing CMOS-compatible solutions. We verify the proposed system through circuit/system-level simulations and evaluate its robustness using a 5000-point Monte Carlo variation analysis. This all-CMOS design paves the way for practical implementation of CiM XOR/XNOR at scaled technology nodes.

INDEX TERMS Artificial intelligence, compute-in-memory, encryption, verification, XOR, XNOR.

I. INTRODUCTION

Academia and industry are pushing their last strides in keeping Moore's law alive, demonstrated by IBM's 2 nm process technology [1]. However, as the available bandwidth between the processor and main memory is not growing commensurately with the advancements in compute units, the well-known 'memory wall' [2] is becoming one of the toughest challenges for engineers in this exascale (big data) computing era. The issue with handling this massive data load is getting more acute with unprecedented progress in machine learning and artificial intelligence (AI) applications. These data-intensive applications require frequent access to memory and hence, von Neumann and memory wall bottlenecks become more pronounced. As a result, the use of conventional von Neumann architectures in these applications leads to negative impacts on energy efficiency, performance, latency, scalability, complexity, and data movement overhead [3]. Recent reports by *Google* have shown that a significant portion of their data center workload is performing bulk data movement and about 20-42% of the energy is required to drive the data bus connecting the compute and memory units [4], [5]. Surprisingly, these data-intensive applications are often not inherently complicated. Rather, they rely on simple logic operations at a massive scale. As an alternative, compute-in-memory (CiM) has garnered attention in the research community [6], [7], [8]. CiM not only dramatically reduces the data movements, but also takes advantage of large internal memory bandwidth and enables massive parallelism to improve latency. In addition to the endeavor to improve the

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Crippa.

architecture, device engineers are exploring next-generation memory technologies as the mainstream CMOS memories are approaching the scaling limit [9], [10], [11], [12]. The emerging memories are expected to provide a faster yet more energy-efficient solution in a compact footprint. Combining the best of both worlds, several CiM architectures have been proposed in recent years with emerging memory devices [13], [14], [15]. However, with exponentially increasing data volume, customized solutions are needed for optimized performance in application-specific scenarios.

With the advent of cloud computing, consumer computer applications are gradually finding their way into virtual machines rather than physical devices, thereby leading to more data in data centers. Keeping this ever-increasing data in a secured backup is a challenging task in terms of performance, energy, and memory. While intelligent and efficient algorithms were proposed for bulk data movement in data centers using row-level cloning [16], integrity verification of the copy procedure is also extremely important. Moreover, in the age of cybersecurity and identity theft, data encryption is equally crucial. Having such securely backed-up data is essential for big data applications like image classification. XOR/XNOR operations are essential for the above-mentioned applications.

Here, we propose a ubiquitous system to achieve single-cycle in-memory bitwise XOR/XNOR operation using modified peripheral sensing circuitry. The novel contributions of this paper are-

1) Designing an All CMOS-based hardware topology for single-cycle in-memory XOR/XNOR operations.

2) Developing a rigorous HSPICE simulation framework and verifying the functionality of in-memory XOR/XNOR operations through transient simulations.

3) Highlighting the effects of external variations on the design through rigorous Monte-Carlo simulations.

4) Comparing the proposed design with the existing approaches in terms of latency.

5) Demonstrating the speedup advantage of the proposed design in implementing XNOR-Net neural network.

The rest of the manuscript is arranged as follows. We discuss the motivation and principle of in-memory XOR/XNOR in section II. We then present our design methodology and the simulation framework (section III). Sections IV and V present the timing simulations and variation analysis, respectively. Section VI presents a comparison with the existing approaches in literature.

II. MOTIVATION FOR SINGLE-CYCLE IN-MEMORY XOR/XNOR

Bulk data copy is such an expensive process (in terms of memory usage and energy demands), that there has been a separate hardware-level instruction set for it since the introduction of Intel IA-32 architecture [17]. In cutting-edge memory chips, an entire row of data is copied from the memory array to the corresponding row buffer, then to the destination row, and finally, a validation is performed to verify the successful copy [18]. This multi-cycle copy and verification procedure is a major concern.

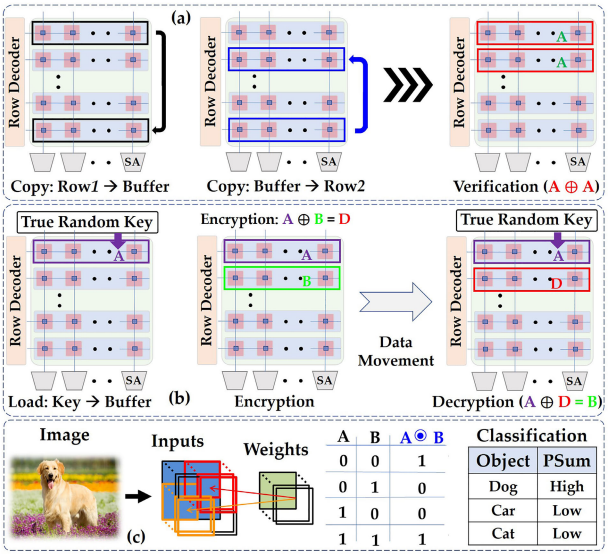

FIGURE 1. A system level view in commercial memory products, where the memory cells are banked, will help understand the latency minimization for the proposed CiM XOR in (a) verification of copied data and (b) data encryption/decryption. (c) CiM configuration can also be used to deploy binary CNN to image classification problem which is essentially an XNOR operation.

For the validation process, parity checking is the most commonly used algorithm and for that, XOR operations between the bits copied from and to the memory cells are performed. A logical '0' XOR output indicates a successful copy operation (Fig. 1(a)). In addition to having back-ups, it is also important to ensure its security. Fortunately, the in-memory XOR operation is perfectly suited for data encryption (Fig. 1(b)). Among the known techniques for ciphers, XOR is the most trustworthy and unbreakable if the key used is a true random number.

Therefore, the significance of performing such XOR/XNOR operations within the memory block (CiM implementation) is well understood. Now, if each of these XOR operations is itself a multi-cycle process, the latency will take a serious hit. All the in-memory XOR operations previously demonstrated take more than one cycle except for one proposed in [14], which too is a memristor-only CMOS non-compatible design, for which the design space will be too complicated. To the best of our knowledge, ours is the first CMOS-compatible in-memory XOR that operates in a single-cycle. We propose a simple all-CMOS-based peripheral circuit design, slightly modifying the sensing circuitry to employ CiM XOR for superior performance in bulk data operations. On top of that, this modification in peripheral circuitry can also be used in binary neural networks like image classification problems, which is essentially an XNOR operation (shown in Fig. 1(c)). Thus, to gain excellent capacity and speed in an in-memory system, the proposed system can be put into use.

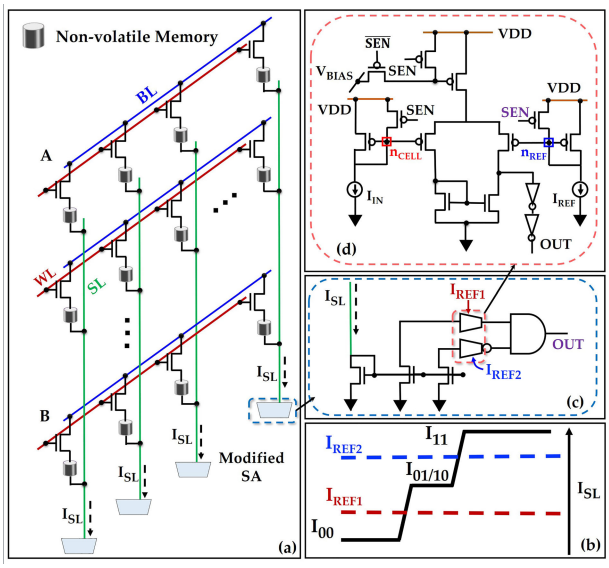

FIGURE 2. (a) Non-volatile memory array with modified sense amplifiers. (b) Mechanism of choosing reference currents. Schematic of (c) the modified SA for in-memory XOR/XNOR and (d) a current sense amplifier.

III. DESIGN METHODOLOGY & SIMULATION FRAMEWORK

For a conventional memory array comprised of access transistors and memory cells, the sense line (SL) currents are collected and sensed via a current-based sense amplifier at the periphery (Fig. 2(a)). In our work, we utilize the current-based sense amplifier (CSA) reported in [19] as the building block for the modified peripheral circuitry to realize the in-memory XOR/XNOR. Here, we use a ReRAM as the NVM cell, but the peripheral circuit modification (all CMOS) to realize the in-memory XOR/XNOR operation is a memory-agnostic design. Irrespective of the memory used, when in computation mode, two-word lines (WL) are asserted in a single sensing line to select the memory cells that will undergo the XOR/XNOR operation. The current contribution of the two selected cells along with the unselected ones of that column is fed into the modified SA. The modified SA consists of a current mirror to copy the SL current, two current-based SAs (CSA), one inverter, and one AND gate as shown in Fig. 2(c). Fig. 2(d) shows the circuit schematic of each CSA used in the SA. The SL current being fed into the two CSAs sets a gate voltage through the current mirror circuit. This set voltages then being compared to the reference voltages, produce binary outputs. As for XOR/XNOR operations, two different reference current levels are being used, they will produce two different logic outputs. These two different logic outputs, one negated through an inverter and the other one intact, fed into the AND gate, give out XOR/XNOR logic. Here it is noteworthy that, the complementary reference current level is set for two CSAs for giving out XOR/XNOR logic output. Different levels of SL current corresponding to different logic conditions along with the reference currents are shown in Fig. 2(b). It can be seen from the illustration that reference current levels are set in between the I_{00} and I_{11} current levels. The reference currents are set in such an intelligent way that an AND operation of the outputs of two CSAs gives out the desired XOR/XNOR result. The sense amplifiers being exactly similar in construction in a CMOS process separates the two extreme cases of both the selected cells storing either '0' or '1' using two reference currents ($I_{00} < I_{REF1} < I_{01}\&I_{01} < I_{REF2} < I_{11}$). This slight modification in peripheral sensing circuitry allows normal memory mode operation as well as single cycle XOR/XNOR operation, which can be very crucial in certain specific application scenarios. Not only that, but this design can also be used to implement other logic operations like AND/NAND, OR/NOR, etc. by carefully choosing the two reference current levels.

In this work, a rigorous SPICE simulation is done for the CiM provision in the memory array. For simulation, a phenomenological compact model of resistive RAM (ReRAM) is used as the non-volatile memory (NVM). The model is calibrated and matched with the experimental data for the Cu/HfO2/Pt stack published in [20]. The low resistance state (LRS) and the high resistance state (HRS) are set at 10 kΩ and 3 GΩ, respectively. 14 nm PTM (Predictive Technology Model) [21] transistors are utilized to simulate the CMOS transistors (FinFETs) used in the memory array and peripheral circuitry. A detailed Monte-Carlo variation analysis is also shown to determine the limitation of the effect of variation on the number of allowed rows in the memory array along with sense margins for the successful operation.

IV. FUNCTIONAL VERIFICATION

Upon setting up the simulation framework, functional verification was performed for the in-memory XOR/XNOR operation in HSPICE. The memory array functions as expected in the memory mode, allowing successful write operations shown in Fig. 3. In the memory mode of operation, the bit lines (BL) are kept precharged and the access transistors are turned on for the selected cell applying suitable biases to the

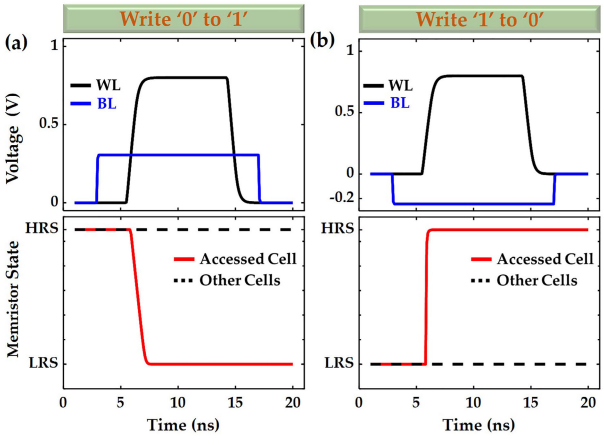

FIGURE 3. (a) Write '0' → '1' (HRS → LRS) and (b) '1' → '0' (LRS → HRS) operations upon applying suitable WL and BL biases.

WLs and SLs. 0.4 V (-0.15 V) is applied to the corresponding BL for writing '1' ('0') into the memory cell, as per the non-volatile memory material we are using from [20]. Later, when WLs are asserted, the accessed cell gets the write voltage applied to the BL. The biasing scheme for write operations is designed in such a way that the half-accessed and unaccessed cells are not accidentally disturbed. Also, reading from the memory cell, we propose to use the same SA designed for the in-memory XOR/XNOR operation to make the peripheral circuitry universal for both memory and compute mode.

To demonstrate the successful operation with our design, we simulated a 3 × 3 array shown in Fig. 2(a). Here, all the bit lines (BL) are pre-charged with a 100mV supply. After the WLs corresponding to two computing rows are asserted, current starts to flow through the memory cells. Fig. 4(a) shows the biasing scheme for the in-memory operations. Now, based on the assumed memory states for the accessed cells, different current levels are obtained in the SLs. The SL current levels for different combinations of memory states in the columns are well distinguishable as shown in Fig. 4(c). Considering the unaccessed cells in HRS, the SL currents are obtained as 100 pA, 7.87 μA, and 15.7 μA for '00', '01'/'10', and '11' logic combinations in the accessed cells, respectively. The reference current levels of the sense amplifiers need to be carefully set in based on these numbers.

For verifying the XOR operation, we set the reference currents as $I_{REF1} = 4\ \mu$A and $I_{REF2} = 12\ \mu$A. When the SEN (Sense Enable) is enabled, the CSAs sense the current levels and result in logic '1' or '0' depending on the SL currents and the reference currents (Fig. 4(c)). As seen, the output of the XOR operation becomes logic '1' only for '01'/'10' logic combination. Note, the SL currents are readily available in the sense amplifiers when WLs and BLs are asserted. Therefore, the XOR operation only requires a single cycle. However, for XNOR operation, the reference currents are set in the exact opposite fashion ($I_{REF1} = 12\ \mu$A and $I_{REF2} = 4\ \mu$A) which also requires single-cycle.

V. VARIATION ANALYSIS

It is seen in Fig. 4(c) that the SL currents are well-distinguishable for different memory combinations in the cells in a single column. However, a quantitative analysis was performed to full-proof the robustness of the design. Even when a cell is not accessed (WL not asserted), a small leakage current flows through those cells: 28 pA for HRS and 774 pA for LRS. The leakage currents through the unaccessed cells contribute to the SL current of the column, which causes a risk of identifying the SL current of one logic combination as another. Therefore, the leakage current (depending on the LRS and HRS values) puts a restriction on the maximum number of rows allowed in an array. Also, average power consumption and area are two very important parameters that directly affect the scaling of the memory system. Fig. 5(a) and 5(b) show the effects of a number of fins on the power consumption and area of the CSA and the effects of HRS and

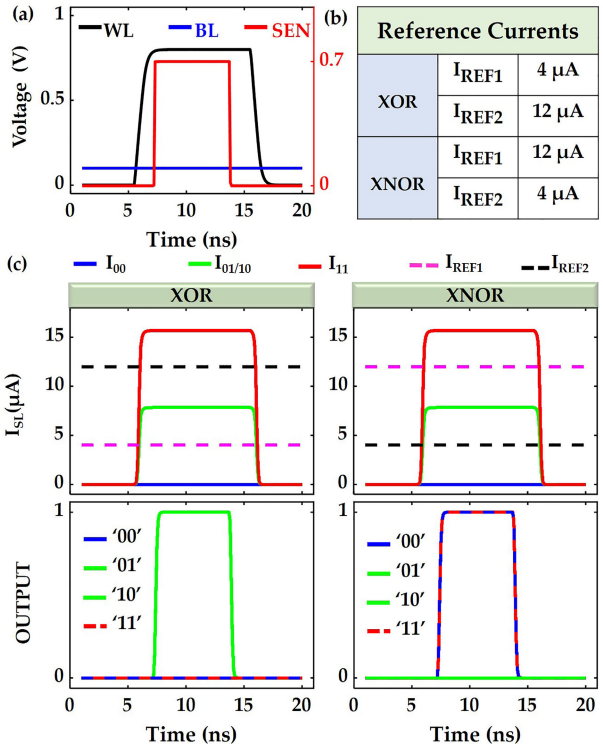

FIGURE 4. (a) The application of required voltages to WLs, BLs, and SEN. (b) Reference current levels chosen for XOR and XNOR operations. (c) SL currents and logic outputs of XOR and XNOR operations.

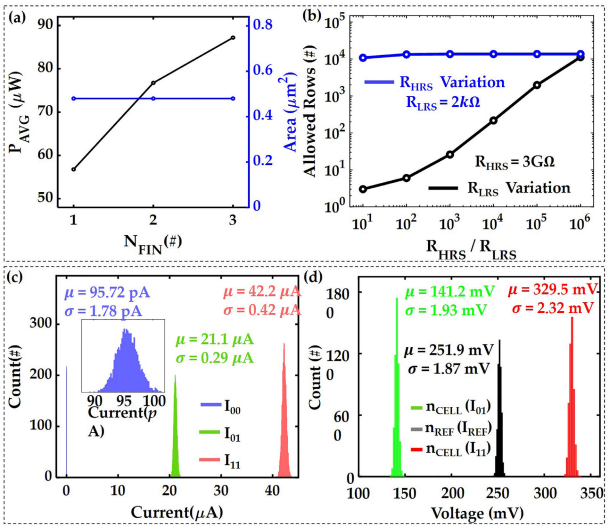

FIGURE 5. (a) Effect of number of fins of the transistors on the CSA circuit and (b) memristor on/off ratio on the array size. Histogram plots of (c) the current distributions and (d) voltages of n_{CELL} and n_{REF} nodes set by the distributions in input and reference current levels, respectively.

LRS values on the maximum number of rows in the array, respectively. In Fig. 5(b), we show the effects of variation in both HRS and LRS separately which shows that the variation in LRS affects more significantly compared to that in HRS. With a fixed HRS, when we vary the LRS by changing the HRS/LRS ratio (black line in Fig. 5(b)), we observe that a larger HRS/LRS ratio results in higher scalability. This

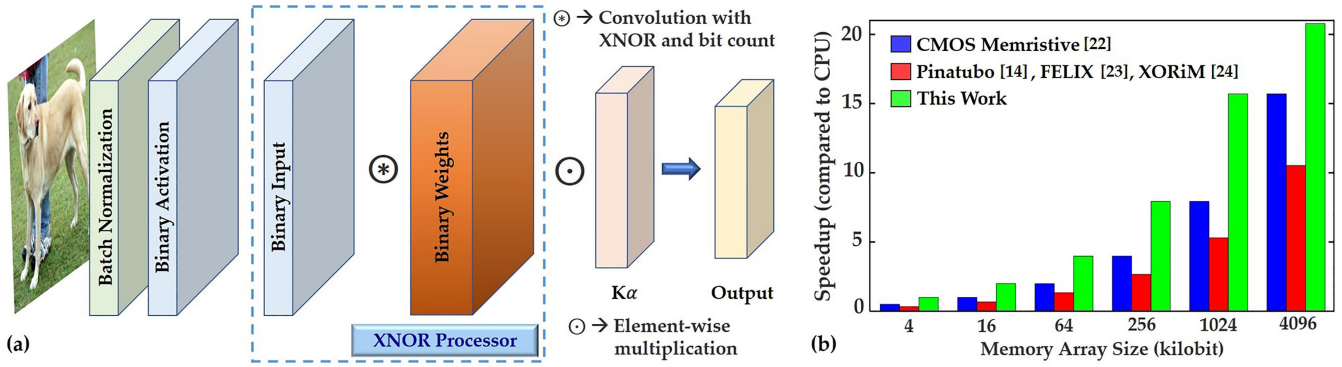

FIGURE 6. Comparison of our design with the existing works based on the implementation of a XNOR-based CNN.

analysis not only lets a designer be aware of the size limitation of the memory array but also opens up a new window of research from the perspective of the material choice.

Furthermore, a rigorous 5000-point Monte-Carlo simulation is performed to ensure that different current levels are well-distinguishable even with the process variations. In our variation analysis, we consider a Gaussian distribution for LRS and HRS with a mean value of 10 kΩ and 3 GΩ (respectively) and a 3σ variation of 10% of the mean value. We also consider a variation in the threshold voltages of the transistors with a standard deviation of 25 mV. The results are shown in Fig. 5(c) and 5(d). Fig. 2(d) shows the schematic of a conventional current sense amplifier with different important nodes marked. The distribution in SL currents shown in Fig. 5(b) leads to a distribution of voltages at the n_{CELL} node of the sense amplifier. Finally, the digital output at the OUT node is obtained based on the difference between the voltages set at n_{CELL} and n_{REF} nodes.

VI. COMPARATIVE STUDY

The surge in compute-in-memory research because of the 'memory wall' problem led to many recent publications. Studies have shown that the ReRAM crossbar array can implement logic operations in the crossbar array [22]. However, some of them are not necessarily fitted to the CiM concept as they use the memory technique to implement processing units. They still pay for the expensive data fetching from the memory and are limited by the memory bus bandwidth. Those that implement the in-memory computation, are tailored to do basic logic operations like AND, OR, etc., some to make ADD operations. Our work can be distinguished from those works in terms of bulk data application in an all-CMOS process.

Based on the required operation steps and overhead circuitry, a comparison with the existing relevant works [14], [22], [23], [24], [25] is presented in Table 1. Our work promises the most efficient solution in terms of latency. Also, an all-CMOS design makes it easy to implement.

We also extend the comparison to the application level using XNOR-Net which uses XNOR operation to replace the

TABLE 1. Comparison of our design with the existing works.

Design	Properties		
	Tech.	Additional Transistors	Latency (Cycles)
Pinatubo [14]	CMOS	7	3
FELIX [23]	Crossbar	-	3
CMOS Memristive [22]	CMOS	16	2
XORiM [24]	CMOS	12	3
SiXOR [25]	Memristor	-	1
This Work	CMOS	13	1

computationally complex convolution operations in convolutional neural networks (CNN). XNOR-Net is a CNN that uses binary filters and XNOR operations to decrease memory cost and decrease computational cost by around 58× [26]. Fig. 6(a) shows a single convolutional block of XNOR-Net. In the beginning, XNOR-Net performs batch normalization and then performs binary activation that binarizes the inputs and generates the scaling factors K and a. From there, the XNOR convolution is performed. We propose using our XNOR processor to accelerate this part of the network. After calculating the XNOR convolution, we then perform element-wise multiplication with the scaling factors (K and a) that we calculated before the XNOR operations. While these operations must be done outside of our accelerator, there are far fewer of these operations than XNOR operations, making our approach still viable despite this limitation. The theoretical speedup due to the use of XNOR convolution is given by [26]-

$$S = \frac{cN_W N_I}{\frac{1}{N_O} cN_W N_I + N_I}$$

Here, c is the number of channels, N_W is the width times the height of the filter, N_I is the width times the height of the input of the layer, and N_O is the number of XNOR operations that can be done in a single clock cycle. In [1], $c = 256$, $N_W = 14^2$, and $N_I = 3^2$ were used since layers with these parameters are common in ResNet [27]. While using a CPU, N_O will be 64, which will be our baseline. Fig. 6(b) shows the speedup of our approach compared to XNOR-Net being executed

in CPU. The speedup of this application compared to the CPU is significantly higher for our XNOR Implementation. We also compare our design with the existing works that require two or three cycles for XNOR operation. Additionally, our design scales better for larger array sizes than the existing designs. In addition to XNOR net, our design could also be used for XOR-Net [28], a version of XNOR-Net that uses XOR and reduces the required number of full precision operations significantly. Using this algorithm, we should see similar speedups and scaling as we did with XNOR-Net, though they will be slightly closer to the ideal $S = \frac{N_O}{64}$ speedup since XOR-Net reduces the full precision operations in a layer with our given parameters by 39.84% [28].

VII. CONCLUSION

In this paper, an all-CMOS single-cycle in-memory XOR/XNOR operation is proposed with a slight modification in the peripheral circuitry. The use of the proposed design is not limited to any specific memory technology. It can be used for all the non-volatile memory technologies to make them capable of performing in-memory XOR/XNOR operations in a single-cycle. Our design allows for a reduced number of cycles and a leap in latency performance. For bulk data operations, even an incremental improvement can be tremendously advantageous. This circuit topology has the potential to revolutionize bulk data copy, verification, and encryption process by reducing the number of cycles required to perform XOR/XNOR operations. The proposed system can also be used in modern and upcoming heavy data applications like binary convolutional neural networks for image classification tasks. Since the designed sense amplifier is CMOS-based, it will not face any difficulty in integrating with the existing memory architectures. The only challenge it will face is higher area needed for the sense amplifier which can be justified by the advantages of in-memory computing and single-cycle XOR/XNOR operations.

ACKNOWLEDGMENT

The publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of the manuscript or allow others to do so, for the U.S. Government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] B. H. McCarthy and S. Ponedal, "IBM unveils World's first 2 nanometer chip technology, opening a new frontier for semiconductors," *IBM News Room*, pp. 6–8, May 2021. [Online]. Available: https://newsroom.ibm.com/2021-05-06-IBM-Unveils-Worlds-First-2-Nanometer-Chip-Technology,-Opening-a-New-Frontier-for-Semiconductors

[2] D. Ielmini and H.-S.-P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, Jun. 2018, doi: 10.1038/s41928-018-0092-2.

[3] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2018, pp. 1–8, doi: 10.1109/CICC.2018.8357072.

[4] S. Kanev, J. Pablo Darago, K. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. Brooks, "Profiling a warehouse-scale computer," in *Proc. ACM/IEEE 42nd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2015, pp. 158–169, doi: 10.1145/2749469.2750392.

[5] A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun, E. Shiu, R. Thakur, D. Kim, A. Kuusela, A. Knies, P. Ranganathan, and O. Mutlu, "Google workloads for consumer devices: Mitigating data movement bottlenecks," in *Proc. 23rd Int. Conf. Architectural Support Program. Lang. Operating Syst.*, vol. 18, 2018, pp. 316–331, doi: 10.1145/3173162.3173177.

[6] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "DRISA: A DRAM-based reconfigurable in-situ accelerator," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2017, pp. 288–301.

[7] Q. Deng, L. Jiang, Y. Zhang, M. Zhang, and J. Yang, "DrAcc: A DRAM based accelerator for accurate CNN inference," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6, doi: 10.1109/DAC.2018.8465866.

[8] Q. Dong, M. E. Sinangil, B. Erbagci, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, "A 351 TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7 nm FinFET CMOS for machine-learning applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 242–244, doi: 10.1109/ISSCC19947.2020.9062985.

[9] H.-S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proc. IEEE*, vol. 98, no. 12, pp. 2201–2227, Dec. 2010, doi: 10.1109/JPROC.2010.2070050.

[10] T.-Y. Liu et al., "A 130.7-mm^2 2-layer 32-Gb ReRAM memory device in 24-nm technology," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 140–153, Jan. 2014, doi: 10.1109/JSSC.2013.2280296.

[11] O. Golonzka et al., "MRAM as embedded non-volatile memory solution for 22 FFL FinFET technology," in *IEDM Tech. Dig.*, Dec. 2018, pp. 18.1.1–18.1.4, doi: 10.1109/IEDM.2018.8614620.

[12] D. Reis, K. Ni, W. Chakraborty, X. Yin, M. Trentzsch, S. D. Dünkel, T. Melde, J. Müller, S. Beyer, S. Datta, M. T. Niemier, and X. S. Hu, "Design and analysis of an ultra-dense, low-leakage, and fast FeFET-based random access memory array," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 5, no. 2, pp. 103–112, Dec. 2019, doi: 10.1109/JXCDC.2019.2930284.

[13] D. Reis, M. Niemier, and X. S. Hu, "Computing in memory with FeFETs," in *Proc. Int. Symp. Low Power Electron. Design*, vol. 18, Jul. 2018, pp. 1–6, doi: 10.1145/3218603.3218640.

[14] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *Proc. 53rd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2016, pp. 1–6, doi: 10.1145/2897937.2898064.

[15] W. Kang, H. Wang, Z. Wang, Y. Zhang, and W. Zhao, "In-memory processing paradigm for bitwise logic operations in STT-MRAM," *IEEE Trans. Magn.*, vol. 53, no. 11, pp. 1–4, Nov. 2017, doi: 10.1109/TMAG.2017.2703863.

[16] V. Seshadri, Y. Kim, C. Fallin, D. Lee, R. Ausavarungnirun, G. Pekhimenko, Y. Luo, O. Mutlu, P. B. Gibbons, M. A. Kozuch, and T. C. Mowry, "RowClone: Fast and energy-efficient in-DRAM bulk data copy and initialization," in *Proc. 46th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2013, pp. 185–197.

[17] *Intel® 64 and IA-32 Architectures Software Developer Manuals*. Accessed: Jan. 18, 2022. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html

[18] X. Xin, Y. Zhang, and J. Yang, "Reducing DRAM access latency via helper rows," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6, doi: 10.1109/DAC18072.2020.9218719.

[19] M.-F. Chang, S.-J. Shen, C.-C. Liu, C.-W. Wu, Y.-F. Lin, Y.-C. King, C.-J. Lin, H.-J. Liao, Y.-D. Chih, and H. Yamauchi, "An offset-tolerant fast-random-read current-sampling-based sense amplifier for small-cell-current nonvolatile memory," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 864–877, Mar. 2013, doi: 10.1109/JSSC.2012.2235013.

[20] N. Shukla, R. K. Ghosh, B. Grisafe, and S. Datta, "Fundamental mechanism behind volatile and non-volatile switching in metallic conducting bridge RAM," in *IEDM Tech. Dig.*, Dec. 2017, pp. 4.3.1–4.3.4, doi: 10.1109/IEDM.2017.8268325.

[21] *Arizona State University Predictive Technology Models*. Accessed: Oct. 17, 2021. [Online]. Available: http://ptm.asu.edu/
[22] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman, "Memristor-based IMPLY logic design procedure," in *Proc. IEEE 29th Int. Conf. Comput. Design (ICCD)*, Oct. 2011, pp. 142–147, doi: 10.1109/ICCD.2011.6081389.
[23] S. Gupta, M. Imani, and T. Rosing, "FELIX: Fast and energy-efficient logic in memory," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 2018, pp. 1–7, doi: 10.1145/3240765.3240811.
[24] K. Zou, Y. Wang, H. Li, and X. Li, "XORiM: A case of in-memory bit-comparator implementation and its performance implications," in *Proc. 23rd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2018, pp. 349–354, doi: 10.1109/ASPDAC.2018.8297348.
[25] N. TaheriNejad, "SIXOR: Single-cycle in-memristor XOR," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 5, pp. 925–935, May 2021, doi: 10.1109/TVLSI.2021.3062293.
[26] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 525–542, doi: 10.1007/978-3-319-46493-0_32.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[28] S. Zhu, L. H. K. Duong, and W. Liu, "XOR-Net: An efficient computation pipeline for binary neural network inference on edge devices," in *Proc. IEEE 26th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2020, pp. 124–131, doi: 10.1109/ICPADS51040.2020.00026.

SHAMIUL ALAM (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology, in 2017, and the M.Sc. degree from the University of Tennessee Knoxville, TN, USA, in 2023, where he is currently pursuing the Ph.D. degree in electrical engineering.

Since January 2020, he has been a Research Assistant with the Nanoelectronic Devices and Integrated Circuits (NorDIC) Laboratory. His research interests include device modeling and circuit design for logic, memory, and in-memory computing applications at room temperature and cryogenic environments.

Mr. Alam's awards and honors include the Tennessee's Top 100 Graduate Fellowship, the DAC Young Fellowship, the Graduate Advancement Training and Education (GATE) Fellowship, and the Gonzalez Family Award for Outstanding Graduate Research Assistant.

JACK HUTCHINS received the B.S. degree in computer science from the University of Tennessee Knoxville, in 2022, where he is currently pursuing the M.S. degree in computer engineering.

During his undergraduate studies, he interned with the Oak Ridge National Laboratory (ORNL), where he worked on machine learning techniques to detect power-related issues with the Titan (supercomputer). In 2021, he was an Undergraduate Research Assistant with the Nanoelectronic Devices and Integrated Circuits (NorDIC) Laboratory, where he continues to work during his graduate studies.

NIKHIL SHUKLA (Member, IEEE) received the Ph.D. degree from the University of Notre Dame, in 2017.

He is currently an Assistant Professor with The University of Virginia with a joint appointment with the Department of Electrical and Computer Engineering and the Department of Materials Science and Engineering. His research interests include co-designing new devices, circuits, and computational models to make computing more efficient.

KAZI ASIFUZZAMAN (Member, IEEE) received the bachelor's degree in computer engineering from North South University, Bangladesh, in 2008, the master's degree in electronic design from Lund University, Sweden, in 2013, and the Ph.D. degree in computer architecture from Universitat Politecnica de Catalunya (UPC), Spain, in 2019.

From 2008 to 2009, he worked on IT systems with Shimizu Densetsu Kogyo Company Ltd., (SEAVAC), Japan. He was a Postdoctoral Researcher with Barcelona Supercomputing Center (BSC), Spain. He is currently a Research Scientist with the Oak Ridge National Laboratory (ORNL), USA. His research interests include memory systems in high-performance computing, including performance analysis, optimization of memory subsystems, and analysis of novel and emerging memory technologies.

AHMEDULLAH AZIZ (Senior Member, IEEE) received the B.S. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology (BUET), in 2013, the M.S. degree in electrical engineering from Pennsylvania State University (University Park), in 2016, and the Ph.D. degree in electrical and computer engineering from Purdue University, in Fall 2019.

Prior to beginning his graduate studies, he was with the Tizen Laboratory, Samsung Research and Development Institute, as a full-time Engineer. He was a Co-Op Engineer (Intern) with the Technology Research Division, Global Foundries (Fab 8), NY, USA. He is currently an Assistant Professor with the Department of EECS, University of Tennessee Knoxville. His research interests include mixed-signal VLSI circuits, non-volatile memory, and beyond CMOS device design.

Dr. Aziz received several awards and accolades for his research, including the Outstanding Dissertation Award from the European Design and Automation Association, in 2020; the Outstanding Graduate Student Research Award from the College of Engineering, Purdue University, in 2019; and the Icon Award from Samsung, in 2013. He was a co-recipient of two best publication awards from SRC-DARPA STARnet Center, in 2015 and 2016, and the Best Project Award from CNSER, in 2013.