Time-multiplexed In-memory computation scheme for mapping Quantized Neural Networks on hybrid CMOS-OxRAM building blocks

Sandeep Kaur Kingra*, Vivek Parmar*, Manoj Sharma and Manan Suri

Abstract—In this work, we experimentally demonstrate two key building blocks for realizing Binary/Ternary Neural Networks (BNNs/TNNs): (i) 130 nm CMOS based sigmoidal neurons and (ii) HfO$_2$ based multi-level (MLC) OxRAM-synaptic blocks. An optimized vector matrix multiplication (VMM) programming scheme that utilizes the two building blocks is also presented. Compared to prior approaches that utilize differential synaptic structures, a single device per synapse with two sets of READ operations is used. Proposed hardware mapping strategy shows performance change of $<$5% (decrease of 2-5% for TNN, increase of 0.2% for BNN) compared to software-based implementation with significant memory savings in the order of 16-32x for classification problem on Fashion MNIST (FMNIST) dataset.

Impact of OxRAM device variability on the performance of Hardware BNN/TNN is also analyzed.

Index Terms—OxRAM, VMM, TNN, BNN, Neuromorphic, In-Memory Computing

I. INTRODUCTION

With advent of artificial intelligence (AI), optimizations of compute architectures to reduce energy and latency have received significant attention [1]. CMOS-based von-Neumann architectures suffer limitations in terms of leakage energy (with scaling) as well as memory bottlenecks [2]. To address the same, novel concepts such as in-memory computing (IMC) [3] as well as quantized low-precision computation [4] have been successfully demonstrated for AI applications. Recent research in quantized neural networks (QNN) focusing on low-precision compute i.e. Binary/Ternary Neural Networks (BNN/TNN) [4]–[7] has created an opportunity to realize computational benefits by exploiting inherent analog computing capabilities of emerging resistive memories [8], [9]. Some of the NVM (non-volatile memory) technologies explored for IMC applications for analog multiplication include: Flash [10], [11], RRAM (resistive random access memory) [3], [12]–[15], and MRAM (magnetoresistive RAM) [16]–[19].

OxRAM based XNOR bitcells provide following advantages: (i) less area and non-volatility compared to SRAM ($\approx$150 F$^2$ per bitcell), (ii) lower operating voltages and faster memory access time compared to Flash, and (iii) lower fabrication cost, reduced area and write energy compared to MRAM.

BNNs assume only binary values (`+1`, `-1`) while TNNs assume ternary values (`+1`, `0`, `-1`) for both synaptic weights and neuronal activations. BNN based on differential synapses have been explored demonstrating error resilience to programming failures [12], [14]. Recent studies have also shown TNNs realized by using similar BNN hardware that offer sparsity benefits [6]–[8]. Recently some groups have investigated incorporation of device non-idealities during training to further improve error resilience of the final implementation [19]. However, this requires increased computational resource during training to account accurately for hardware variability. Training using conventional algorithms while resulting in less accuracy may still result in tolerable error resilience at lower computational costs. In this study, we propose BNN/TNN implementation using fabricated 1T-1R device-based synapses (Fig. 1(a)) by exploiting MLC [20] OxRAM programming and CMOS sigmoid neuron (on-chip) for classification on Fashion MNIST (FMNIST) dataset [21]. Key contributions of the work are listed below:

1) Demonstration of a batchnorm-free end-to-end implementation (synapse and neuron on same die) of CMOS-OxRAM based hardware QNNs (BNN/TNN).
2) Time-multiplexed VMM computation using single device per synapse and sequentially-applied input activation.
3) Performance comparison of QNNs based on BNN/TNN as well as impact of MLC state distributions using FMNIST dataset.
4) Proposed hardware mapping strategy for QNN shows performance change of $<$5% (decrease of 2-5% for TNN, increase of 0.2% for BNN) compared to software-based implementation.

The manuscript is organized as follows: Section II describes key concepts such as BNN/TNN along with hardware implementations, Section II presents details of fabricated circuits (sigmoid neuron and 1T-1R synapse) and device characterization results, Section IV describes proposed VMM computation scheme for BNN/TNN realization, Section V describes key results based on network simulations and finally Section VI provides concluding remarks.

(c) Copyright 2022 IEEE. This work was supported in part by SERB-CRG/2018/001901 and IITD-FIRP grant. S. K. Kingra, V. Parmar, M. Sharma and M. Suri are with the Indian Institute of Technology Delhi, New Delhi.

* S. K. Kingra and V. Parmar contributed equally for the manuscript.

Corresponding Authors: Manan Suri (manansuri@ee.iitd.ac.in).

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
II. PRIOR ART

QNNs enable an efficient mapping of neural network computations to analog hardware by restricting precision \[1\] while simultaneously offering significant resilience to errors \[23\]. Extreme forms of QNN investigated in literature include BNN (2 states/1-bit) and TNN (3 states/2-bit). BNNs are realized in digital hardware in form of XNOR gates followed by popcount circuits. TNNs add ‘0’ state and replace XNOR with Gated-XNOR \[5\]. Gating is realized by checking non-zero status of both inputs thus avoiding unnecessary computation of XNOR operations. This further offers benefits in terms of sparsity (for storage) as well as pruning (for computations) in comparison to BNN. The key difference exists between both in terms of activation functions employed as shown below in Eq. \(1\) and Eq. \(2\).

\[
\begin{align*}
    Act_{\text{binary}}(x) &= \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (1) \\
    Act_{\text{ternary}}(x) &= \begin{cases} +1, & x > r \\ -1, & x < -r \\ 0, & x \leq \text{abs}(r) \end{cases} \quad (2)
\end{align*}
\]

BNN hardware implementations based on In-Memory Computing (IMC) typically employ usage of differential weights and inputs to realize dual polarity (‘+1’, ‘-1’) in hardware \[7\], \[12\], \[14\]. Recent studies have also shown TNNs realized by using similar hardware that offer sparsity benefits \[6\]– \[8\] by employing a 3rd state combination for the synapse (both devices in HRS) and activations (all 0’s). This leads to savings in energy dissipation for computation. Some studies have explored benefits of employing sigmoid neurons with binarized weights to improve accuracy \[18\]. However, this can potentially lead to implementation complexities since the resulting output may not conform to the signal requirement of BNN/TNN as sigmoid can result in an analog value (see Fig. 2). To perform computation for all layers directly based on output of previous layer it becomes essential to use activation functions described in Eq. \(1\) and Eq. \(2\) for all the intermediate layers.

III. FABRICATED CMOS-OXRAM CIRCUITS

A single resistive memory cell comprises of 1T-1R (1 transistor-1 resistor), where NMOS transistor acts as the selection element in the array. TiN/HfO\(_2\)/Ti/TiN OxRAM devices (referred as R in 1T-1R), having 300 nm cross point diameter are monolithically integrated on top of 130 nm CMOS baseline process (see Fig. 1(a)). The basic cell structure and device stack are shown in Fig. 1(b). The experimental test setup used in this study is shown in Fig. 1(c). In Fig. 1(d), the layout of 1T-1R bitcell is shown highlighting the OxRAM device.
layers. Experimentally measured NMOS selector characteristics ($I_D$ vs $V_{DS}$, $I_D$ vs $V_{GS}$) are shown in Fig. 5. The forming procedure is required on pristine OxRAM devices before executing SET/RESET programming operations. DC-IV curves showing FORMING, SET and RESET switching actions of 1T-1R OxRAM devices are presented in Fig. 5(a). Fig. 5(b) illustrates pulse switching behavior of the 1T-1R structure. For measuring the device resistance state, a 10 µs read pulse with $V_{read} = 0.2$ V is used. MLC programming of the 1T-1R structure is achieved (see Fig. 5) by modulating $V_{stop}$ for HRS (High Resistance State) and $I_{comp}$ (compliance current) for LRS (Low Resistance State). In Fig. 5(a), by sweeping $V_{stop}$ from -0.9 V to -1.2 V (applied at top electrode), different HRS region states are realized. For MLC in LRS region, $V_{gate}$ of NMOS selector is tuned to control $I_{comp}$. In Fig. 5(c), by sweeping the $V_{gate}$ from 0.8 V to 1.5 V, multiple LRS states are realized. To realize hardware BNN/TNN, we selected three distinct resistive states in HRS/LRS region. These well separable programming states are selected based on programming reproducibility. Fig. 5(b,d) presents the HRS/LRS distribution for selected 1T-1R bitcell states including C2C/D2D (C=cycle, D=device) variability. As observed, the states are well separable, thereby offering better reliability for BNN/TNN realization.

Using the same baseline 130 nm CMOS we designed and fabricated 6T-sigmoid neurons (Fig. 6). Fig. 6(a) presents the circuit schematic for the CMOS sigmoid neurons that acts as current-voltage conversion circuit. Here, $I_{input}$ current acts the input signal to the circuit and output is realized as $V_{neuron}$. Ideal, sigmoid function has output as defined in Eq. (3),

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}$$

Experimentally characterized sigmoid neuron’s transfer function characteristics are shown in Fig. 6(b). Here a slight offset in the $V_{neuron}$ output is observed and it is represented as Eq. (4) where $I_{input}$ is specified in µA. The layout of the sigmoid neuron is shown as inset in Fig. 6(b).

$$V_{neuron}(I_{input}) = \frac{1.5156}{1 + e^{-(input - 1.56)}} + 0.1$$

### IV. VMM Operation Mapping

In this study, compared to prior work BNN/TNN are realized by exploiting analog MLC states of the OxRAM device and a two-stage input mapping to perform computations. While implementing analog VMM using resistive memory arrays, output is realized as current integrated along the column [9]. Sigmoid neuron translates this integrated current ($I_{input}$) into output voltage ($V_{neuron}$). Neuron gain can be fine-tuned by carefully scaling the $(\frac{W}{L})$ of the 6 transistors.
Simulated S/H-based time-multiplexed differentiation circuit for validation of proposed methodology. The circuit is simulated with resistance values based on experimental characterized 1T-1R data along with 130 nm CMOS technology.

**Fig. 9.** Input and output waveforms based on SPICE simulations of time-multiplexed differential circuit validating proposed methodology. $V_{out}$ is high in case output of $S/H_2$ exceeds output of $S/H_1$. $V_{in}$ and $V_{clk}$ are pulses of amplitude 1.4V. Since clock signals are applied at the gate terminal of PMOS transistor, output is sampled on the input lines when $V_{clk}$ is low. Similarly, final output voltage $V_{out}$ is sampled when $V_{clk,diff}$ is low.

For performing vector multiplication, synaptic weights are mapped as OxRAM device conductance (2 states for BNN, 3 states for TNN) while input activations are applied as $V_{gate}$ during two consecutive READ operations at time instances $t_0$ and $t_1$ (Fig. 7). To map inputs with opposing polarities i.e. (+1’, -1’), input vector is split into two components i.e. +ve component and –ve component. For +ve component: $V_{gate,+}=1.2$ V is applied for input $\geq 0$ (i.e., ‘+1’) and $V_{gate,-}=0$ V is applied for input $< 0$ (i.e., ‘-1’). For –ve component, the vice-versa signals are applied for input ‘+1’ and ‘-1’. For ternary input value ‘0’, $V_{gate}=0$ V is applied for both input vector components. READ operation for each input vector component generates a column-wise integrated current ($I_{+0}$ at $t_0$, $I_{-0}$ at $t_1$ for column 0). Since both components are applied at consecutive time instances, Sample & Hold (S/H) circuits are used to store the integrated current outputs at each time instant. Once both vector components have been applied, S/H outputs are fed to a comparator. The comparator output is then applied to an activation function (here CMOS sigmoid neuron which is used for last layer).

To validate the proposed S/H based circuit, extensive circuit simulations are performed. The simulated circuit used for validation is shown in Fig. 8. Corresponding input and output waveforms captured from SPICE simulations are shown in Fig. 9. Input in the form of integrated current ($I_{+0}$/$I_{-0}$) from a column is first converted to voltage $V_{sum}$ which is captured using phase-delayed clocks ($V_{clk,S/H_1}$/$V_{clk,S/H_2}$) by two S/H circuits ($S/H_1$, $S/H_2$). The output voltage ($V_{S/H, out}$) from each S/H circuit is applied at the input of a differential amplifier with gain adjusted based on column length and resistance distribution of the MLC devices. This output can now be fed to any activation function circuit (binary, ternary, sigmoid) to compute the neuron output. As shown in Fig. 8, the proposed S/H-based differential circuit requires 3 op-amp blocks. The area of a single op-amp is estimated to be 45.26 $\mu$m$^2$ thus leading to an overall area cost of $\sim 136$ $\mu$m$^2$. Total energy dissipation of the S/H-based differential circuit for a single column operation is estimated to be 83 pJ with a latency of 50 ns. Though there is an increase in the area for the
periphery circuits with the proposed approach, the energy cost is not as significant. Prior studies have shown energy cost of programming a single OxRAM device to be \( \sim 200-400 \) pJ \[24\]. Since the proposed approach enables usage of a single device per synapse, it reduces programming energy cost by half. The area cost of the periphery is already a well-established limitation even with approaches using two devices per synapse that have been addressed using schemes such as multiplexed sensing \[14\]. While the proposed approach shows marginal limitations in terms of area, it still shows significant energy saving in scenarios where weight re-programming is performed at regular intervals (\( \sim 2 \times \)).

To validate VMM computation methodology, extensive numerical simulations are performed for a \( 4 \times 4 \) 1T-1R array. Simulated voltage-output resulting of sigmoid neurons, \( V_{\text{neuron}} \), (operating in the linear region) from the array is shown in Fig. 10(a,b) for HRS based weight mapping (i.e. the device state mapped on OxRAM array is chosen from Fig. 5(b)) and Fig. 10(c,d) for LRS based weight mapping (i.e. the device state mapped on OxRAM array is chosen from Fig. 5(d)). Popcount denotes mathematical product of binary/ternary precision vectors utilizing \((+1,-1)/(+1,0,-1)\) as weights and activations. As shown in Fig. 10, TNNs show popcount output at all integer discrete levels (refer Fig. 10(b,d)). However, BNN discretizes output further to multiples of 2 (see Fig. 10(a,c)). As a result, TNN output is more sensitive to the device variability as compared to BNN.

V. NETWORK SIMULATIONS

To validate the proposed hardware, CNNs with multiple (binary/ternary) precisions are trained using QKeras \[25\] framework on the FMNIST dataset based on training algorithms discussed in \[26\]–\[28\]. Using thermometric encoding \[29\] grayscale images are transformed into 8 channels of binary images (refer Fig. 11(a)). Network based on LeNet architecture \[30\] used for the study is shown in Fig. 11(b). Training evolution curves for BNN/TNN (in terms of loss) are shown in Fig. 11(c). Due to batchnorm-free training, optimization converges within few epochs and saturates.

Network simulations incorporating experimentally measured OxRAM device conductance (including C2C/D2D variability) are performed by mapping trained weights to MLC conductance state. Weight distributions for BNN/TNN in terms of HRS device conductance are shown in Fig. 12(a,b) and for LRS device conductance in Fig. 12(c,d) respectively. A clear separable distribution for each selected MLC state can be observed in Fig. 12. HRS region states exhibit lesser current dissipation (compared to LRS region states) during READ operation for inference, leading to better energy efficiency. Accuracy results for simulated and ideal networks (full-precision and QNN) are summarized in Table I. Compared to full-precision networks, the ideal BNN and TNN show an accuracy drop of \( \sim 8\% \) and \( \sim 6\% \) while offering memory savings of \( 32 \times \) and \( 16 \times \) respectively. In case of the proposed OxRAM-based BNN implementation, there is a marginal increase in the performance compared to ideal BNN that can be attributed to the imbalance resulting from synaptic weight mapping i.e. unlike symmetric ‘+1’/-1’ weights, they are realized as positive conductance values relating to LRS/HRS distributions of the devices. For the proposed TNN, accuracy drop of 2% is observed compared to an ideal TNN in case of HRS based MLC state mapping and 4% in case of LRS based MLC state mapping. This can be attributed to better separability (i.e. dissipation).
relative window between states ‘-1’ and ‘0’) in distributions exhibited for HRS MLC states as compared to LRS MLC states. To illustrate impact on class-wise accuracy (for 10 classes in FMNIST), confusion matrices for BNN/TNN for HRS region state mapping are shown in Fig. 13(a,b) respectively. Similarly, confusion matrices for BNN/TNN for LRS region state mapping are shown in Fig. 13(c,d) respectively. It can be observed that similar classes such as (0, 2, 4, 6) with matching features exhibit more false positives compared to other classes. False positive count increases even further when using LRS MLC states with TNN which can be attributed to sensing region overlap as observed in Fig. 10.

VI. CONCLUSION

Hybrid CMOS-OxRAM neuronal and synaptic building blocks for realizing low-resource BNN/TNN have been experimentally demonstrated. Through extensive characterization, realization of reliable MLC OxRAM states with low C2C/D2D variability in order to store synaptic weights on a single 1T1R bitcell was validated. A differential read scheme was presented for performing vector multiplication. BNNs/TNNs were simulated exploiting aforementioned MLC OxRAM states utilizing both LRS and HRS states. Simulated network showed performance degradation (2-5%) compared to ideal TNN realized and a 0.18% improvement compared to ideal software BNN highlighting impact of sensitivity to MLC state distributions. While there may be marginal limitations in the proposed approach in terms of area, it still shows significant energy saving in scenarios where weight re-programming is performed (∼2×) at regular intervals.
[16] L. Chang, X. Ma, Z. Wang, Y. Zhang, Y. Xie, and W. Zhao, “Pxnor-bnn: In/with spin-orbit torque mram preset-xnor operation-based binary neural networks,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 11, pp. 2668–2679, 2019.

[17] S. Gao, B. Chen, Y. Qu, and Y. Zhao, “Mram acceleration core for vector matrix multiplication and xnor-binarized neural network inference,” in 2020 International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA), 2020, pp. 153–154.

[18] M. Elbitry, A. Singh, B. Reidy, X. Guo, and R. Zand, “An in-memory analog computing co-processor for energy-efficient cnn inference on mobile devices,” in 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2021, pp. 188–193.

[19] W. A. Misba, M. Lozano, D. Querlioz, and J. Atulasimha, “Energy efficient learning with low resolution stochastic domain wall synapse based deep neural networks,” 2021. [Online]. Available: https://arxiv.org/abs/2111.07264

[20] E. Hsieh, M. Giordano, B. Hodson, A. Levy, S. Osekowsky, R. Radway, Y. Shih, W. Wan, T. F. Wu, X. Zheng, M. Nelson, B. Le, H.-S. Wong, S. Mitra, and S. Mitra, “High-density multiple bits-per-cell 1t4r rram array with gradual set/reset and its effectiveness for deep learning,” in 2019 IEEE International Electron Devices Meeting (IEDM), 2019, pp. 35.6.1–35.6.4.

[21] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017. [Online]. Available: https://doi.org/10.3389%2Ffnins.2019.01383

[22] V. Parmar, B. Penkovsky, D. Querlioz, and M. Suri, “Hardware-efficient stochastic binary CNN architectures for near-sensor computing,” Frontiers in Neuroscience, vol. 15, Jan 2022. [Online]. Available: https://doi.org/10.3389%2Ffnins.2021.781786

[23] V. Parmar, M. Suri, K. Yamane, T. Lee, N. L. Chung, and V. B. Nalik, “Mram-based her resilient quantized edge-ai networks for harsh industrial conditions,” in 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2021, pp. 1–4.

[24] T. Hirtzlin, M. Bocquet, B. Penkovsky, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, and D. Querlioz, “Digital biologically plausible implementation of binarized neural networks with differential hafnium oxide resistive memory arrays,” Frontiers in Neuroscience, vol. 13, Jan 2020. [Online]. Available: https://doi.org/10.3389%2Ffnins.2019.01383

[25] C. N. C. Jr., A. Kuusela, S. Li, H. Zhuang, J. Ngadiuba, T. K. Aarrestad, V. Loncar, M. Pierini, A. A. Pol, and S. Summers, “Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors,” Nat. Mach. Intell., vol. 3, no. 8, pp. 675–686, 2021. [Online]. Available: https://doi.org/10.1038/s42256-021-00356-5

[26] B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, “Minimum energy quantized neural networks,” in 2017 51st Asilomar Conference on Signals, Systems, and Computers, 2017, pp. 1921–1925.

[27] K. Hwang and W. Sung, “Fixed-point feedforward deep neural network design using weights +1, 0, and -1,” in 2014 IEEE Workshop on Signal Processing Systems (SiPS), 2014, pp. 1–6.

[28] F. Li, B. Zhang, and B. Liu, “Ternary weight networks,” 2016. [Online]. Available: https://arxiv.org/abs/1605.04711

[29] D. A. Rachkovskiy, S. V. Slipchenko, E. M. Kussul, and T. N. Baidyk, “Sparse binary distributed encoding of scalars,” Journal of Automation and Information Sciences, vol. 37, no. 6, pp. 12–23, 2005.

[30] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Computation, vol. 1, no. 4, pp. 541–551, 1989.