Abstract—In this paper, we present an efficient hardware mapping methodology for realizing vector matrix multiplication (VMM) on resistive memory (RRAM) arrays. Using the proposed VMM computation technique, we experimentally demonstrate a binarized-ADALINE (Adaptive Linear) classifier on an OxRAM crossbar. An 8×8 OxRAM crossbar with Ni3-nm HfO2/7 nm Al-doped-TiO2/TiN device stack is used. Weight training for the binarized-ADALINE classifier is performed ex-situ on UCI cancer dataset. Post weight generation the OxRAM array is carefully programmed to binary weight-states using the proposed weight mapping technique on a custom-built testbench. Our VMM powered binarized-ADALINE network achieves a classification accuracy of 78% in simulation and 67% in experiments. Experimental accuracy was found to drop mainly due to crossbar inherent sneak-path issues and RRAM device programming variability.

I. INTRODUCTION

In-Memory Computing (IMC) and analog hardware Vector Matrix Multiplication (VMM) approaches offer efficient alternatives to conventional computing for resource hungry modern-AI workloads [1]. Advance neural networks such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) involves extensive use of VMM operations. In particular, emerging resistive memory (RRAM) nanodevice based arrays offer an efficient and compact option for performing VMM operations in hardware. Several research groups [2]–[6] have demonstrated this analog computing method for a variety of applications such as linear equation solver [6], image processing [7], data compression [8], feature extraction [9], neural network inference [10], in-situ training [10], [11], etc. Multiple emerging memory nanodevice technologies have been utilized for this application : OxRAM [1], MRAM [1], PCM [12], Ferroelectric FET [13], ECRAM [14], Flash [15], etc. Owing to their higher device density, crossbar structures are preferred for VMM applications [16]. Typical implementations of crossbar based VMM operations utilize analog conductance states of RRAM nanodevices [1], [2], [4], [17]. However, difficulty to obtain multiple reliably programmable resistance states, non-linearity of analog RRAM device conductance and variability related issues pose a major challenge to this approach. Furthermore complex program-and-verify schemes may be required in certain cases [18]. In case of selector-free crossbars these issues are further enhanced due to the presence of sneak-paths [19]. Binary neural networks (BNNs) have been shown to have better performance (i.e. lower memory, time and energy requirements) [20], [21], compared to their full-precision (analog) counterparts, at the cost of a marginal accuracy trade-off. Hence using crossbar based VMM for hardware realization of BNN would face lesser challenges compared to fully analog alternatives. Furthermore, use of binary memory states leads to simplification of programming with relatively less impact on device endurance. Recently BNNs have been demonstrated on RRAM matrix using 2T-2R synaptic cells [22]. However, true density benefit of emerging RRAM technology can be exploited only by using selector-free RRAM crossbars. In the current study, we propose a hardware mapping methodology for realizing VMM using binary RRAM crossbars. Further we experimentally demonstrate the proposed technique with a case-study of binarized-ADALINE using OxRAM crossbar. To the best of our knowledge, this is the first experimental validation of BNNs using selector-free RRAM crossbars.

Compared to literature we present the following novel concepts in this paper:

1) Weight mapping methodology, algorithm and operation scheduling for implementing BNN on any RRAM array (ex- OxRAM, CBRAM, PCM, etc.).
2) Experimental demonstration of a binarized-ADALINE network on an 8×8 OxRAM crossbar for classification on UCI cancer dataset [23].

The paper is organized as follows: Section II describes basic
concepts relevant for this work. Section III describes the proposed methodology for mapping BNNs and training a binarized-ADALINE network. In Section IV, we describe the custom-test platform in detail along with experimental results on fabricated OxRAM crossbar.

II. BASICS AND BACKGROUND

A. Vector Matrix Multiplication (VMM) in Hardware

Fig. 1(a) describes the standard implementation of VMM operation in a generic two-terminal resistive nanodevice crossbar based architecture. Input is applied in the form of voltages ($V_{in,i}$) across all rows in parallel. Based on the conductance state ($G_{i,j}$) of the devices current integrates on each column. The resulting integrated current is converted to voltage using a Trans-Impedance Amplifier (TIA) with feedback resistor ($R_f$). Output voltage ($V_{out,j}$) or the output of VMM operation (across a given column j of the matrix) is defined by Eq. 1:

$$V_{out,j} = R_f \times V_{in,i} \times G_{i,j} \quad (1)$$

B. ADALINE

‘Adaptive Linear Element’ also known as ADALINE, was one of the first networks proposed based on “memistors” (not memristors) [24]. It is used for a variety of classification applications such as power-quality event detection [25], stereo-vision matching [26], etc. that require fast computation. It is used for a variety of classification applications such as power-quality event detection [25], stereo-vision matching [26], etc. that require fast computation. It is used for a variety of classification applications such as power-quality event detection [25], stereo-vision matching [26], etc. that require fast computation.

III. PROPOSED METHODOLOGY FOR VMM-BASED BNN

A. Training for Binarized-ADALINE

We use a modified version of the training algorithm proposed in [27]. The specific modifications include: (i) Training with only positive value inputs. This helps to simplify input encoding scheme compared to inputs with signed magnitudes as normally used for training BNNs [20]. (ii) Adding support for training a binarized-ADALINE network by using binary ‘tanh’ as the hard-limiting quantizer and binary weights. (iii) Support for training non-vision datasets by use of min-max normalization on input features. The training algorithm is summarized in Algorithm 1. Post-training, binary weights (+1,-1) are generated.

B. Weight Mapping Strategy

Proposed weight mapping strategy for using two-terminal RRAM device crossbar is shown in Fig. 2. In this discussion, we primarily focus on the binarized-ADALINE as an example network since it represents a basic unit for realization of a multi-layer BNN. Every weight vector is mapped using two separate components i.e. $W^+, W^-$ in consecutive rows of the crossbar. If length of input vector is greater than number of columns, we partition weights and allocate them on separate set of rows. Since two devices are utilized for representing each logical weight (shown in Fig. 2), the utilization of crossbar is reduced by 50%. Summation performed at each row of the crossbar can be represented by Eq. 2 (where $k$ denotes output class, $i$ denotes polarity i.e. +ve, -ve and $p$ denotes partition of feature vector [0,n]). Post summation stage (i.e. TIA output), final sum voltages from rows $W^+, W^-$ are subtracted at the 2nd stage (Eq. 3) leading to a class-wise score $V_{o,k}$, which is used for generating a class decision at the final stage. In case of partitions, all output voltages of a respective polarity (i), for class $k$ ($V_{o,k,i}^p$) are first stored and summed (Eq. 3) before proceeding to the subtraction and class-decision stages. In proposed binarized-ADALINE, we use analog inputs of 8-bit precision. 8-bit Inputs are realized by utilizing pulse-width modulation (PWM) based encoding.

Algorithm 1 Algorithm for training binarized-ADALINE

Input: Training data X, Expected target Y, Epochs N, Batch-size B
Output: Target $Y_t$.

**Initialisation**:
1. $X_o = X - \text{min}(X) / (\text{max}(X) - \text{min}(X))$
2. $X_i = \text{round}(X_o \times 255)$
3. $Y_o = \text{One-HotEncode}(Y)$
4. Initialise weight vector W
5. Binarize W

**LOOP Process**
6. for $t = 0$ to $N$
7. $Y_t[B] = W \cdot X_i$
8. error = $Y_t[B] - Y_o[B]$
9. Calculate $\delta W$ using squared-hinge loss with ADAM optimizer
10. Binarize W
11. end for

Fig. 2. Proposed scheme for computation and weight mapping using two-terminal resistive crossbar. Every logical weight value (i.e. +1 or -1) is mapped on the crossbar using 2 paired devices from consecutive rows (i.e. rows $W^+$ and $W^-$) of the same column. The paired devices are always programmed to complementary states (LRS-HRS or vice-versa). In particular, to realize logical weight +1, device from the first row is programmed to LRS and the paired device in the consecutive row is programmed to HRS. For realizing logical weight -1, programming is inverted (i.e. first row device is in HRS while consecutive row device is in LRS). Eight logical weight values (-1,1,1, -1,1,1,-1) are programmed using 16 (4x4) devices. The first four weights corresponding to class $k=0$ (-1,1,1,-1) are partitioned in rows 1 and 2, while the next four weights corresponding to class $k=1$ (1,1,-1,-1) are partitioned in rows 3 and 4. Note, input voltages are applied on columns and current integration occurs across rows. This is due to the fact that DAC units used in our experimental setup generate only positive voltages. Negative voltages are effectively realized by grounding device top-electrode and applying +ve DAC signal at device bottom electrode. Programming and read paths are isolated using CMOS switches for each channel.
which requires integration across 255 cycles to generate the final class score. Fig. 3 summarizes sequence of operations.

\[ V_{o,p}^{k,i} = W_{i,p}^{k} \cdot X \]  
\[ V_{o}^{k,i} = \sum_{p=1}^{n} V_{o,p}^{k,i} \]  
\[ V_{o,k} = V_{o,k}^{+} - V_{o,k}^{-} \] 

IV. EXPERIMENTAL RESULTS

A. Testbench & Dataset

Fig. 4 shows the custom-testbench built for this application. The main components are:

1) Microcontroller: Drives the overall VMM operation scheduling. It is also used for communication with host PC, managing control signals to all other ICs, OxRAM crossbar programming/sensing/weight-mapping partitioning, computation of outputs post-TIA stage, final class score computation and inference decision.

2) Sensing Circuitry: Consists of TIA for converting current to voltage signal and ADC unit (from microcontroller).

3) DAC units: For generating required OxRAM programming signals. We used \( V_{set} = 3.3 \) V, \( V_{reset} = -5.5 \) V and \( V_{read} = -0.8 \) V for programming OxRAM devices in our crossbar. (High values of \( V_{set}, V_{reset} \) are used in our prototype devices as they have large active area (100 x 100 \( \mu m^2 \)). Negative \( V_{reset}, V_{read} \) were realized by applying +ve voltage signals at the bottom electrode of OxRAM device and grounding the top electrode, thereby generating effective -ve \( V_{TB} \) (where \( V_{TB} = V_{Top} - V_{Bottom} \)).

4) CMOS Switches: For path selection (during SET, RESET and READ operation) and compliance current control.

For validation of the proposed methodology, we trained the binarized-ADALINE network using a binary classification dataset (Breast Cancer dataset) from the UCI machine learning repository [23]. The dataset consists of 357 benign and 212 malignant cells.

B. Fabricated OxRAM Crossbar Chip

8x8 OxRAM crossbar with resistive switching stack of Ni/3-nm HfO\(_2\)/7 nm Al-doped-TiO\(_2\)/TiN (shown in Fig. 5(a)) was fabricated for this study. First, 500 nm thick TiN film was deposited on thermal-SiO\(_2\) (500 nm)/Si wafer by reactive DC sputtering. The wordlines were then patterned by optical photolithography (first mask) and dry etching using inductively-coupled plasma (ICP). The bottom, 7 nm thick ATO dielectric, was then deposited by interchanging varying amount of TiO\(_2\) and Al\(_2\)O\(_3\) PE-ALD cycles. Upper, 3 nm thick dielectric HfO\(_2\) film, was deposited using TDMAHf (Tetrakis(dimethylamido)hafnium) and O\(_2\) plasma. All depositions were carried out at 250 °C using remote plasma hot-wall reactor PE-ALD system. The TE pattern (similar to the BE pattern but rotated 90°) was defined using second mask and 100 nm thick Ni top electrode film was deposited by DC sputtering and patterned using lift-off technique. This way, an 8x8 crossbar was formed with 100 \( \mu m \) wide perpendicular TiN and Ni wordlines and bitlines sandwiching the dielectric bilayer, forming 64 OxRAM devices with 100x100 \( \mu m^2 \) active area at each crosspoint. The third mask and ICP dry etching step was performed to open the contact windows (etch the dielectrics) to the wordline contact pads. Wire bonding and packaging were the final steps for the OxRAM crossbar encapsulation. DC IV characteristics of 64 OxRAM devices in the 8x8 crossbar are overlaid in Fig. 5(b). In our study, we have used two distinguishable well separated OxRAM resistance states (HRS/LRS) as shown in Fig. 5(c).

C. BNN Results on OxRAM Crossbar

To perform BNN computation with OxRAM crossbar, row-level READ operations were used. Since we used \( V_{read} = -0.8 \) V, due to negative read voltage current integration happened over the row elements. For binarized-ADALINE network (see Fig. 6(a)), we partitioned and mapped the weight vector over multiple rows (shown in Fig. 6(c)) and performed VMM operations. The output class was decided by Eq. 4.
as described in Section III-B. The 8-bit wide positive input values were encoded as PWM duty cycles applied to crossbar columns (Fig. 6(b)) leading to current integration over time. In PWM encoding, the input value (ranging from 0 to 255) was translated to pulses with fixed voltage (0.8 V). Specific pulse-widths were n × 17 ms (where n = number of clock cycles [0-255]). Prior to executing VMM operations the entire crossbar was initialized to HRS (Fig. 6(d,e)) for reliable weight programming. Crossbar resistance distribution post-mapping of trained weights is shown in Fig. 6(f,g). Training was performed on 80% of the UCI cancer dataset with random shuffling. Classification accuracy results are shown in Table I. We can observe a reduction in classification accuracy between simulated and experimental network. The drop in accuracy can be attributed to sneak-path issue inherent to crossbars. Due to sneak-path leakage, the effective integrated current value across a specific row is diminished. Another factor contributing to accuracy loss is the variability in OxRAM device programming (i.e. LRS and HRS distributions) as evident from Fig. 5(c), 6(g). An effective strategy to mitigate these effects would be to implement two separate ADALINE networks in place of a single neuron. Deep binarized multi-layer networks can also be explored, using larger RRAM arrays to further improve the learning performance as shown in simulation studies [28]–[30]. It should be noted that by using binary OxRAM states (HRS/LRS), we have limited the effect of variability that would otherwise reflect in an analog resistance VMM implementation. Furthermore, using OxRAM crossbar solely for inference relaxes the endurance requirements for the devices.

V. CONCLUSION

In this paper, we presented a novel methodology for mapping BNN operations on two-terminal binary NVM device based crossbars of arbitrary sizes. We experimentally demonstrate the realization of a binarized-ADALINE classifier, for UCI Cancer dataset, on fabricated 8×8 OxRAM crossbar. The demonstrated methodology can be extended to deeper BNNs through proposed energy-efficient in-memory VMM computation strategy.

ACKNOWLEDGEMENT

This work was supported in part by SERB-CRG/2018/001901 and CYRAN AI Solutions.
