3D-DPE: A 3D High-Bandwidth Dot-Product Engine for High-Performance Neuromorphic Computing

Miguel Angel Lastras-Montaño*, Bhaswar Chakrabarti*, Dmitri B. Strukov* and Kwang-Ting Cheng*†
*Department of Electrical and Computer Engineering, University of California, Santa Barbara, United States
{mlastras, bchakrabarti, strukov, timcheng}@ece.ucsb.edu
†School of Engineering, Hong Kong University of Science and Technology, Hong Kong

***Abstract*—We present and experimentally validate 3D-DPE, a general-purpose dot-product engine, which is ideal for accelerating artificial neural networks (ANNs). 3D-DPE is based on a monolithically integrated 3D CMOS-memristor hybrid circuit and performs a high-dimensional dot-product operation (a recurrent and computationally expensive operation in ANNs) within a single step, using analog current-based computing. 3D-DPE is made up of two subsystems, namely a CMOS subsystem serving as the memory controller and an analog memory subsystem consisting of multiple layers of high-density memristive crossbar arrays fabricated on top of the CMOS subsystem. Their integration is based on a high-density area-distributed interface, resulting in much higher connectivity between the two subsystems, compared to the traditional interface of a 2D system or a 3D system integrated using through silicon vias. As a result, 3D-DPE's single-step dot-product operation is not limited by the memory bandwidth, and the input dimension of the operations scales well with the capacity of the 3D memristive arrays.**

To demonstrate the feasibility of 3D-DPE, we designed and fabricated a CMOS memory controller and monolitically integrated 2 layers of titanium-oxide memristive crossbars. Then we performed the analog dot-product operation under different input conditions in two scenarios: (1) with devices within the same crossbar layer and (2) with devices from different layers. In both cases, the devices exhibited low voltage operation and analog switching behavior with high tuning accuracy.

I. Introduction

Artificial neural networks (ANNs) have shown to be excellent tools to solve a wide variety of problems that are otherwise too complex to solve with conventional methods. One of such problems is the *classification problem*, in which given an object, we are asked to determine whether the object is or is not a member of a set [1]. State-of-the-art ANN architectures for image classification such as VGGNet [2] and ResNet [3], require, among others, several layers of fully-connected networks, resulting in hundreds of millions of parameters, making the memory and the memory bandwidth the major bottlenecks of the system.

One of the most recurring and computationally expensive operations in ANN algorithms is the *dot-product* operation. As shown in Fig. 1(a), a hybrid CMOS/memristive circuit can be used to naturally implement the dot-product operation in a single step [4]. Programming the conductances c_i in one column of a memristive crossbar, and applying voltages v_i as inputs in the rows (top electrodes), result in a current I that is equal to the dot-product between the input vector $\mathbf{v} = \{v_1, \cdots, v_n\}$ and the conductance vector $\mathbf{c} = \{c_1, \cdots, c_n\}$.

The major challenge of computing the dot-product in the analog domain is that very large fan-in structures with tens of thousands of inputs are often needed to implement some fully-connected layers of an ANN. Furthermore, the large number of parameters needed to evaluate an ANN demands a high-bandwidth (analog) memory. These large fan-in and high-bandwidth requirements cannot be met with a $n \times n$ 2D crossbar, as it would need to be prohibitively large with n in the 10^3-10^4 range, whereas for yield and practical reasons, n should be closer to the 10^1-10^2 range.

To meet the bandwidth and fan-in challenges, we propose *3D-DPE*: a 3D High-Bandwidth Dot-Product Engine, in which the inputs to an ANN are not only distributed along one dimension, as in a 2D crossbar, but in two dimensions, forming an *area-distributed interface* between the memristive crossbar and CMOS subsystems. We employ a CMOL-like interface [5] for the CMOS subsystem and the address topology proposed in [6] to form a stack of multiple layers of memristive crossbar arrays. In contrast to using a single large 2D crossbar to implement the dot-product operation, 3D-DPE uses a hierarchical approach in which the large crossbar is divided into smaller crossbars, that we call *mini-crossbars*, and then multiple layers are stacked forming a 3D crossbar array. These two approaches are illustrated in Fig. 1(b). Such hierarchical approach allows 3D-DPE to take advantage of the high-density (and thus high memory capacity) of crossbar arrays of memristors while providing a high-bandwidth and large fan-in interface. The potential of 3D-DPE is huge: single-step dot-product operations with millions of inputs and the capacity of storing trillions of weights all within a single 1 cm^2 chip [6].

To demonstrate the feasibility of 3D-DPE, we designed and taped out a CMOS memory controller that serves as a platform for the monolithic integration of memristive devices. We then fabricated different configurations of two layers of memristive crossbars on top of the CMOS chip. The memristive devices in both layers exhibited analog behavior and multiple levels of resistance values can be programed with high accuracy. We experimentally evaluated our dot-product engine under different inputs in two scenarios: The first, called *intralayer dot-product*, implements the dot-product *within* one layer of memristors (through multiple mini-crossbars). The second, called *interlayer dot-product*, implements the dot-product *between* two or more mini-crossbars in different layers of the crossbar arrays. In general, 3D-DPE can implement both intralayer and interlayer dot-product operations simultaneously.

II. Background

A. Artificial Neural Networks as Classifiers

The fundamental unit in an ANN is the *artificial neuron* shown in Fig. 2(a). In general, an artificial neuron contains n inputs, labeled x_1 to x_n, and a single output y. Each input x_i has associated a weight w_i (represented by a line). The output y is computed as a function of the *dot-product* between the input vector $\mathbf{x}$ and weight vector $\mathbf{w}$, $y = f(\mathbf{x} \cdot \mathbf{w})$, where f is a non-linear single-input scalar function called the *activation function*. Note that the computational complexity of f is relatively trivial compared to the complexity of the dot-product operation.

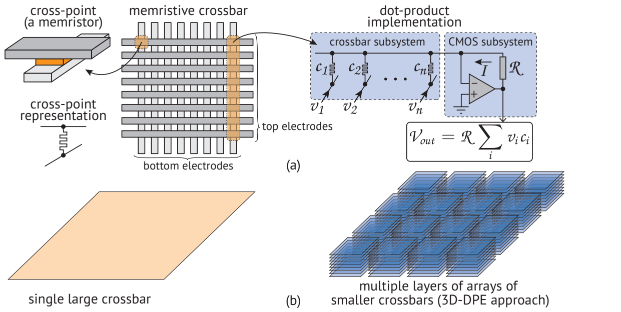

Fig. 1. (a) A memristive crossbar, a cross-point, and the analog dot-product implementation. (b) Two approaches to map the dot-product operation into a crossbar.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

A common architecture for ANN classifiers organizes the artificial neurons in three types of layers: an *input layer*, one or more *hidden layers*, and an *output layer*, as shown in Fig. 2(b). The number of inputs in the input layer is set by the size of the objects to classify, e.g., the number of pixels in an image. The number of outputs is determined by the number of classes. The size and topology of the hidden layer is architecture-dependent. Once its structure is defined, there are two phases for an ANN: a *training phase* and an *evaluation phase*. During the training phase, a set of inputs (called the *training set*) is used to iteratively train, i.e., modify, the weights of the network. During the evaluation phase, the class of an object is predicted by directly observing the *output scores* from the output layer.

An important characteristic of this type of ANNs is the rich interconnection that exists between the neurons in two adjacent layers, in which the output of every neuron is connected as input to all the neurons in the next layer. State-of-the-art ANN-based image classification architectures, such as the 19-layered VGGNet [2], require, among others, *fully-connected* layers with up to 4096 inputs. ResNet [3], the ILSVRC 2015 winner, is an "extremely deep" 152-layered ANN that heavily relies on simpler *convolutional layers*, but still requires a 1000-input fully-connected output layer. In both architectures, more than 10 billion dot-product operations and thousands of millions of parameters (weights) are needed during the training and evaluation phases. This is a nontrivial and computationally expensive task, even for high-end GPU-accelerated systems.

B. Resistive Random-Access Memories

A *memristor* or *resistive random-access memory cell* (RRAM cell) is a two-terminal device whose resistance can be reversibly changed by applying a voltage across its terminals. This change in resistance is non-volatile and can persist for years after the applied voltage is removed [4]. Although an RRAM cell can be treated as a binary memory cell, high-resolution multi-level switching behavior has been demonstrated with up to 7-bits of resolution [7]. This high tunability creates the opportunity of using RRAM in analog computing.

High-density RRAM arrays can be obtained using the *crossbar architecture*, in which at every cross-point, an RRAM cell is formed as shown in Fig. 1(a). This architecture does not require an *access element* per RRAM cell, instead, the access element (usually a transistor) is shared by n RRAM cells, i.e., a 1TnR architecture. This contrasts with the 1T1R architecture (1 transistor per 1 RRAM cell) in which the overall size of the memory cell is dominated by the size of the transistor, thus annulling the high-density benefits of RRAM.

As depicted in Fig. 1(a), an RRAM crossbar can implement the costly dot-product operation in a single step. Moreover, the vector-matrix multiplication can also be implemented in constant time. For this, the input vector is mapped as voltages applied to the rows of an RRAM crossbar, and the values of the matrix are directly mapped to the conductances of each crosspoint. The resulting vector is proportional to the current in each column of the RRAM crossbar.

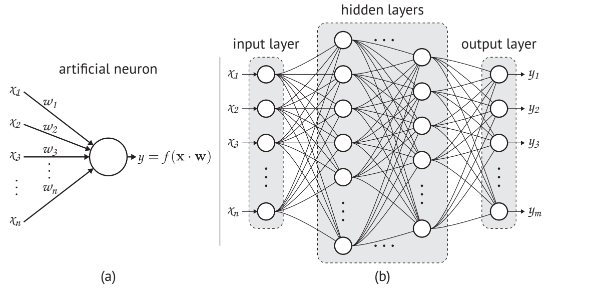

Fig. 2. (a) An artificial neuron with n inputs and a single output y. The output is a function of the weighted sum of all the inputs. (b) A multi-layer artificial neural network with n inputs, one or more hidden layers, and m outputs.

III. 3D High-Bandwidth Dot-Product Engine

To achieve the large number of inputs required by state-of-the-art and future ANN classifiers, 3D-DPE employs a hierarchical approach, in which instead of using large 2D RRAM crossbars of tens of thousands of inputs, each crossbar is divided into equally-sized $n \times n$ *mini-crossbars* with n in the range of 10^1-10^2. The rationale is the following: First, the segmentation of the nanowires forming an RRAM crossbar allows the use of an area-distributed interface between the RRAM subsystem and the underlying CMOS drivers [5]. Second, multiple layers of mini-crossbars can be stacked to effectively reduce the footprint of an RRAM cell [6].

Fig. 3(a) shows the proposed hierarchical organization of 3D-DPE to implement the dot-product between the input vector $\mathbf{v}$ and the conductance vector $\mathbf{c}$ (implicitly depicted as RRAM cells). For the sake of visual simplicity, only the RRAM cells involved in the dot-product operation are shown. 3D-DPE consists of B *banks*, each with T *tiles*, which in turn are formed by an array of up to L^2 mini-crossbars, where L is the number RRAM crossbar layers. Each mini-crossbar $\mathcal{M}_{i,j,k}$ can be uniquely identified with the three integers i, j, k with $1 \leq i \leq T$, $1 \leq j \leq L$ and $1 \leq k \leq B$. The input vector $\mathbf{v}$ is similarly divided into smaller *mini-vectors* $\mathbf{v}_{i,j,k}$. Each mini-vector $\mathbf{v}_{i,j,k}$ corresponds to the input vector of each mini-crossbar $\mathcal{M}_{i,j,k}$. Note, however, that whereas the size of a mini-vector is n, the size of a mini-crossbar is $n \times n$.

A 3D-DPE bank (Fig. 3(b)) is a self-contained unit that is formed by peripheral row and column decoders, peripheral read and write (R/W) circuitry, and a central area-distributed interface in which several layers of RRAM mini-crossbars are monolithically integrated by means of an array of high-density vias. The mini-crossbars contained in a 3D-DPE tile are highlighted in red at the left of Fig. 3(b). The same tile is isolated and detailed at the far right of Fig. 3(b). All tiles in a bank share the same row decoder, but have their individual column decoder and R/W circuitry. Also note that a tile is only accessed by the fraction of the area-distributed interface that is directly below it (highlighted in yellow). Finally, at the right of Fig. 3(b) we highlight in red a possible set of mini-crossbars that are involved during a dot-product operation. The 3D stair-like distribution of the mini-crossbars comes from the mapping between layers as proposed on [6], which results in only one active mini-crossbar per layer. Note that while the same topology of the L-layered 3D stair-like crossbar can be obtained with a single 1-layer 2D crossbar that is L times longer (a crossbar of size $n \times nL$), 3D-DPE provides L times more capacity to store weights, within the same chip area.

The simplified diagram of a mini-crossbar and its interface with the CMOS chip is depicted in Fig. 3(c). It is made up of a high-density array of n by n nanowires being accessed with an array of blue and red vias, which in turn are controlled with the peripheral decoders. A pair of blue and red vias together with their CMOS *access elements* form a *CMOS cell* [5]. The design and complexity of the CMOS cell depend on its final application. For high-density memory applications, simpler designs are preferred, e.g., by having a single transistor as an access element. For reconfigurable logic, a CMOS inverter can be used in addition to the access transistors [8]. For neuromorphic applications, a summing amplifier can be included in the CMOS cell, although depending of its size, it can be shared by all the mini-crossbars in a tile, or even by all the tiles in a bank.

Each 3D-DPE bank can perform the dot-product operation with up to $n \times T \times L$ inputs. The number of inputs for a single dot-product operation can be further increased by using up to B banks

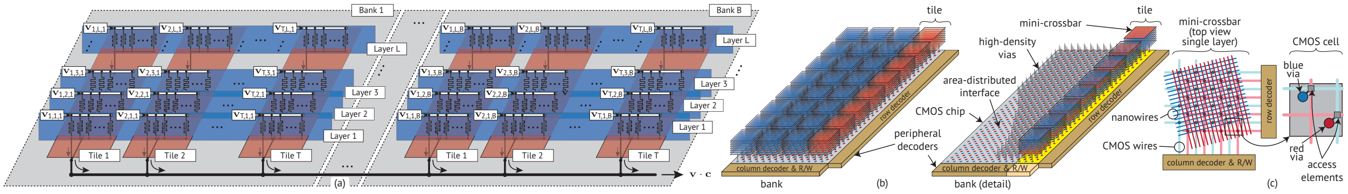

Fig. 3. (a) Hierarchical implementation of the dot-product operation between the vector **v** and the conductance vector **c** (shown as RRAM cells). Only the RRAM cells involved in the dot-product operation are shown. (b) A 3D-DPE bank, formed by lateral decoders, read/write circuitry, and a central area-distributed interface, from which a high-density vias interconnect multiple layers of mini-crossbars. (c) Schematic of a mini-crossbar and a CMOL cell.

in parallel, resulting in a total of $n \times T \times L \times B$ inputs, which is the upper limit of the dimension of the input vectors for dot-product. Even for a small system with $n = 64$, $T = 8$, $L = 8$ and $B = 8$, 3D-DPE can provide up to 32K inputs and a capacity to store 16M weights. For a more aggressive but still realistic system with $n = 256$, $T = 256$, $L = 32$, $B = 64$, and 20 nm RRAM cells [9], 3D-DPE can provide up to 2M inputs per bank and store 1T (10^{12}) weights in a single chip that is within 1 cm^2.

IV. Experimental Evaluation

To demonstrate the feasibility of 3D-DPE we monolithically integrated two layers of titanium-oxide 3D RRAM crossbars on a CMOS memory controller. The CMOS chip was fabricated on a ON-Semi C5 0.5 μm process and contains read and write circuitry as well as row and column decoders [10], [11]. This CMOS chip has been previously used to successfully integrate one layer of single RRAM cells, as in a 1T1R architecture [12]. We employed the programming algorithm proposed in [7] to program the weights of the RRAM cells.

Figs. 4(a-b) show the area-distributed interface of the CMOS chip as well as a close-up of the pads used to access the RRAM devices. The lateral decoders (not shown in the figure) allow the selection of any pair of blue and red pads. Fig. 4(c) shows a diagram of the steps to fabricate two layers of our 3D RRAM crossbar structure. This is a simplified CMOL-like structure that allows an arbitrary number of layers to be fabricated reusing the same photolithography masks. In addition to the 3D RRAM crossbar, we also fabricated small 2×2 and 3×3 RRAM crossbars, as well as single devices for test purposes. Fig. 4(d) is an AFM image showing the partial structure (three cross-points) of an integrated RRAM crossbar.

The first experiment we performed was the dot-product operation between multiple RRAM cells located *within* the same layer (layer 1) in different 3D-DPE tiles. We call this operation, the *intralayer dot-product*. Fig. 5(a) summarizes the results of the intralayer dot-product with 3 inputs. Each input (also called a channel) had a 40 mV peak-to-peak sinusoidal waveform of different frequencies: 400 Hz, 500 Hz and 600 Hz, for channels 1, 2 and 3, respectively. The dot-product operation is performed on-chip, i.e., the multiplication of the input voltages with the conductances, as well as the addition of currents are performed inside the chip, however, the resulting current is converted to voltage with an off-chip transimpedance amplifier (TIA). The resulting voltage is plotted at the far left of Fig. 5(a). Then we proceeded to program each channel individually while maintaining the other two constant. At the right of Fig. 5(a) we show the evolution from top to bottom of each channel. These values were obtained by extracting the values at 400 Hz, 500 Hz and 600 Hz of the Fourier transform of the output of the TIA. In the first stage, the weight (in arbitrary units) of channel 1 was changed in relatively “coarse” steps from 1.44 to 1.26. In the second stage, the weight of channel 2 was changed in “finer” steps from 0.54 to 0.58. Finally, the weight of channel 3 was changed in “intermediate” steps from 0.34 to 0.40. While the weights are reported in arbitrary units, they are all in the same scale. The evolution of the output is subtle, but it allows us to demonstrate the precision with which we can program the devices.

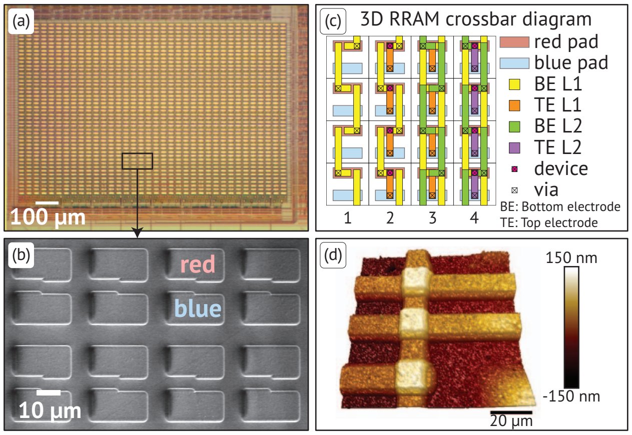

Fig. 4. (a) Optical micrograph of the area-distributed interface of the CMOS memory controller. (b) SEM image detailing the interconnection pads at the top metal layer. (c) Steps to fabricate two layers of the 3D RRAM crossbar. (d) AFM image of three integrated RRAM cells.

In the second experiment, we performed the dot-product between RRAM cells located on different layers (between layers 1 and 2). We call this an *interlayer dot-product*. The number of inputs in this experiment was limited by the number of layers (two in this case). The methodology is similar to the intralayer dot-product. In this case we had a low-frequency sinusoidal input (input 1 in layer 1) and high-frequency sinusoidal input (input 2 in layer 2) with a frequency 10× higher than that of input 1. The results are summarized in Fig. 5(b). We first increased the weight of the low-frequency input (input 1) from 0.7 to 0.92 while keeping the other constant, resulting in an output with a higher low-frequency component. As a visual aid, we included a gray background with the peak-to-peak envelope of the initial waveform. Then we increased the weight of the high-frequency component (input 2) from 0.1 to 0.5 and kept the other constant, resulting in a clearly more pronounced ripple at the output. In contrast to the experiment of Fig. 5(a), there is a small crosstalk between the inputs, as evidenced by the apparent change in the weight of the device that we did not program (see for instance the small drop in the weight of the device in layer 1 when programming the device in layer 2). We found this change to be only apparent and due to the comparable resistance of the RRAM cells and the CMOS access transistor (a few kΩ) which resulted in a non-negligible voltage drop across the access transistor. A bigger access transistor minimizes this effect.

V. Related Work

L. Gao *et al.* [13], demonstrated the analog dot-product operation using two discrete RRAM cells and an operational amplifier. M. Prezioso *et al.* [14] demonstrated the use of the dot-product operation to classify three classes of 3×3 patterns using a 12×12 RRAM crossbar. Recently, M. Hu *et al.* [15] proposed a dot-product engine using 2D 1T1R crossbars. In their work, they present an algorithm to map arbitrary values to the conductances

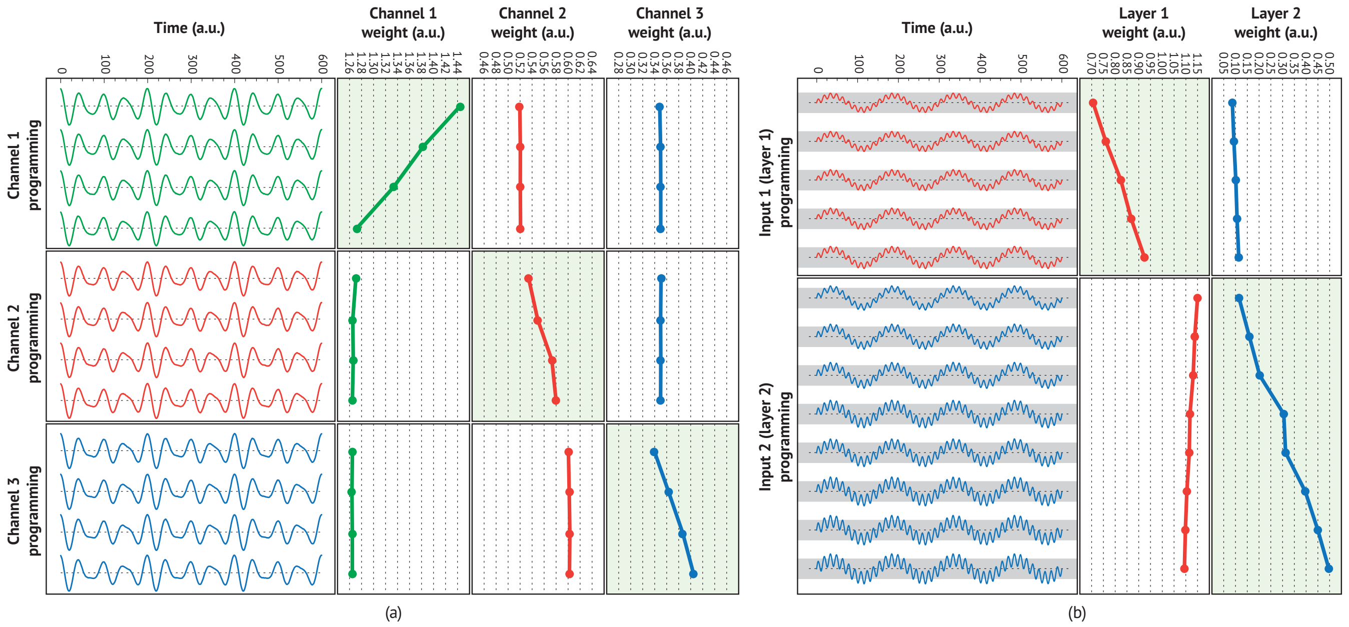

Fig. 5. Evaluation of the dot-product operation using 3D-DPE. (a) Intralayer dot-product operation with three inputs (channels). (b) Interlayer dot-product operation with two inputs, each coming from a different crossbar layer. The input/channel being programmed is highlighted with a light-green background.

of the RRAM cells, however, they validated their approach only via simulations. Regarding RRAM-based architectures for neuromorphic computing, X. Liu *et al.* [16] recently proposed the use of a mixed-signal interconnection network to assist the communication between memristive crossbars. On other CMOS/RRAM integration efforts, Q. Xia *et al.* [17] experimentally demonstrated a hybrid reconfigurable logic using planar 2D RRAM cells fabricated on top of a CMOS chip using a CMOL interface. J. Sandrini *et al.* [18] integrated a 2D 8×8 RRAM crossbar on a CMOS chip. Recently, H. Li *et al.* [19] demonstrated a stack of four RRAM cells accessed with a single transistor. To the best of our knowledge, 3D-DPE is the first experimental demonstration of a monolithically integrated functional 3D RRAM crossbar on a standard CMOS chip.

VI. Concluding Remarks

In this paper we propose 3D-DPE, a highly parallel dot-product engine, to be used as an accelerator for artificial neural networks (ANNs). 3D-DPE is made up of a CMOS subsystem and several layers of high-density resistive random-access memory (RRAM) crossbars, monolithically integrated on top of the CMOS subsystem. 3D-DPE leverages the simplicity of implementing the dot-product operation in the analog domain in constant time, and provides very high-bandwidth between the CMOS subsystem and the RRAM crossbar by means of a high-density area-distributed interface. We experimentally validated 3D-DPE by monolithically integrating 2 layers of RRAM crossbars and implementing the analog dot-product operation on 2D and 3D RRAM crossbars. Although these were simple experiments, the potential of 3D-DPE is huge: fan-in structures of millions of inputs and terabyte-scale memory capacities, all in a $1\,\text{cm}^2$ chip.

While we mainly discussed the acceleration of ANN-based algorithms, 3D-DPE is a general-purpose engine that can accelerate any application that requires linear transformations. Most of these applications are already designed around the idea of a limited bandwidth, however, we believe that freeing the developers of this restriction will result on a new class of high-performance neuromorphic applications.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research under the MURI grant FA9550-12-1-0038, and the Industrial Technology Research Institute (ITRI).

References

[1] V. N. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.
[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.
[4] J. J. Yang, D. B. Strukov, and D. R. Stewart, “Memristive devices for computing,” *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.
[5] K. K. Likharev and D. B. Strukov, “CMOL: Devices, circuits, and architectures,” in *Introducing Molecular Electronics*. Springer, 2005, pp. 447–477.
[6] D. B. Strukov and R. S. Williams, “Four-dimensional address topology for circuits with stacked multilayer crossbar arrays,” *PNAS*, vol. 106, no. 48, 2009.
[7] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, “High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm,” *Nanotechnology*, vol. 23, no. 7, p. 075201, 2012.
[8] D. B. Strukov and K. K. Likharev, “CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices,” *Nanotechnology*, vol. 16, no. 6, p. 888, 2005.
[9] “The International Technology Roadmap for Semiconductors (ITRS), System Drivers, 2013, http://www.itrs.net/.”
[10] M. A. Lastras-Montaño, A. Ghofrani, and K.-T. Cheng, “Architecting Energy Efficient Crossbar-based Memristive Random Access Memories,” in *NANOARCH*, 2015.
[11] M. Payvand *et al.*, “A Configurable CMOS Memory Platform for 3D Integrated Memristors,” in *ISCAS*, 2015.
[12] J. Rofeh *et al.*, “Vertical Integration of Memristors onto Foundry CMOS Dies using Wafer-Scale Integration,” in *ECTC*, 2015.
[13] L. Gao, F. Alibart, and D. B. Strukov, “Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices,” in *VLSI-SoC*, 2012.
[14] M. Prezioso *et al.*, “Training and operation of an integrated neuromorphic network based on metal-oxide memristors,” *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.
[15] M. Hu *et al.*, “Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication,” in *DAC*, 2016.
[16] X. Liu *et al.*, “RENO: A high-efficient reconfigurable neuromorphic computing accelerator design,” in *DAC*, 2015.
[17] Q. Xia *et al.*, “Memristor-CMOS hybrid integrated circuits for reconfigurable logic,” *Nano letters*, vol. 9, no. 10, pp. 3640–3645, 2009.
[18] J. Sandrini *et al.*, “Co-Design of ReRAM Passive Crossbar Arrays Integrated in 180 nm CMOS Technology,” *JETCAS*, 2016.
[19] H. Li *et al.*, “Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing,” in *Symposium on VLSI Technology*, 2016.