Magnetic tunnel junction-based crossbars: improving neural network performance by reducing the impact of non-idealities

William A. Borders[1], Nitin Prasad[2,3], Brian D. Hoskins[1], Advait Madhavan[2,4], Matthew W. Daniels[1], Vasileia Georgiou[5], Tiffany S. Santos[5], Patrick M. Braganca[5], Mark D. Stiles[1], and Jabez J. McClelland[1]

[1]Physical Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA, william.borders@nist.gov
[2]Associate, Physical Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA
[3]Department of Chemistry and Biochemistry, University of Maryland, College Park, MD, 20742, USA
[4]Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, MD, 20742, USA
[5]Western Digital Research Center, Western Digital Corporation, San Jose, CA, 95119, USA

Increasingly higher demand in chip area and power consumption for more sophisticated artificial neural networks has catalyzed efforts to develop architectures, circuits, and devices that perform like the human brain. However, many novel device technologies suffer from non-idealities such as device variation, or circuit sneak paths that reduce network accuracy. Here, we report that an array of magnetic tunnel junction devices integrated with complementary metal oxide semiconductors (CMOS) greatly reduces the impact of non-idealities in the circuit and performs inference with accuracies nearly identical to software.

***Index Terms*—Spintronics, Magnetic tunnel junction, neuromorphic computing, binary neural network**

I. Introduction

As artificial neural networks scale to improve their potential for training computers to perform like the human brain, increasingly larger chip area and power budgets pose an issue for embedded applications. This issue arises from an architecture mismatch between neural networks and computers, leading to research in devices, circuits, and architectures that resemble the nature of the brain and demonstrate a potential for compact efficient artificial neural networks. One promising system is the binary neural network, reported to perform comparably to precise software-based models [1], mapped onto conventional computer memory to process and store in the same location [2-5]. A recent work demonstrated inference accuracy of 95.3 % on a binary neural network mapped onto a 15×15 crossbar array of magnetic tunnel junctions (MTJ) [5], the key component of magnetoresistive random access memory (MRAM).

One challenge that novel technologies face is the impact of non-idealities in the hardware on neural network performance. In the MTJ crossbar used in [5], line resistance and potential sneak paths resulted in high levels of variation in the effective device conductance and incorrect operation of the neural network. Operating the physical network requires a conversion factor, G_{norm}, between device conductance and network weights. Due to non-idealities, different values of the conversion factor are required to optimize accuracy of inference or reduce the error of the written weight matrix than the value approximated for the nominal properties of the array.

In this work, we report on a similar 15×15 MTJ crossbar this time integrated with CMOS to reduce line resistance and avoid sneak paths. We demonstrate that the reduced impact of these non-idealities results in the same G_{norm} for both optimizing accuracy and weights and report an improvement in accuracy of the network from 95.3 % to nearly identical accuracy with the software model (99.3 %).

II. Device Preparation and Setup

Experiments are carried out on a 15×15 array of MTJs integrated with a 180 nm CMOS process. MTJ stacks for patterning perpendicular easy-axis devices are deposited above exposed vias connected to the topmost CMOS metal layer and processed into 50 nm (nominal) diameter pillars using electron beam lithography and Ar^+ ion milling. A final Cr/Au metallization process is then performed to connect the top electrodes of the devices to the select transistors in the array. The 2 transistor 1 resistor (2T1R) array is fabricated without control circuitry; all measurements are through port-to-port measurements by a probe card connected to a 4-channel source measure unit and a switch matrix, contained in a single chassis. Single device measurements are performed by connecting the probes to the row, column, and gate (3.3 V) of the corresponding 2T1R cell. Due to the channel number limitation of the source measure unit, inference on the neural network is performed by accessing each device separately and sequentially.

III. Neural Network Operation

Similar to the work in [5], the Wine dataset [6] is used for classification, which includes 178 test samples with 13 input parameters each. A two-layer neural network with 13 input neurons, 6 hidden neurons, and 3 output neurons is used, resulting in a 13×6 and 6×3 weight matrix for layers 1 and 2, respectively. Ternary weights [-1, 0, 1] are implemented with two MTJs, where the conductance difference between the two proportional to the weight allows for implementation of negative weights. The network is first trained offline using 148 of the 178 samples to produce 300 unique weight matrix solutions using different weight initializations. Each solution is then written to the MTJ array to perform inference.

The crossbar is used to perform the vector-matrix multiplication (VMM) operation of the network by mapping the inputs of each layer to the corresponding read voltages at each row. We take advantage of Kirchhoff's laws to obtain the output of each layer as the resulting current on the column. The currents are then normalized into layer outputs by $V_{read} \times G_{norm}$, where V_{read} is the read voltage of the MTJ, and G_{norm} is approximated as G_{ON}-G_{OFF}, the MTJ ON-state average conductance and the MTJ OFF-state average conductance, respectively.

IV. Results

Figure 1 shows inference accuracy and root-mean-squared (RMS) deviation between the ideal and measured weight matrices versus G_{norm} for the passive array from [5] and the CMOS-integrated array. A large contribution to a reduction in accuracy in the previous work originated from the array's line resistance, approximated to 6 Ω per square. This line resistance increased variation between devices, resulting in a large discrepancy between the approximated $G_{norm,}$ the G_{norm} that optimizes accuracy, and the G_{norm} that minimizes the RMS deviation between the written weight matrix in hardware and the ideal one.

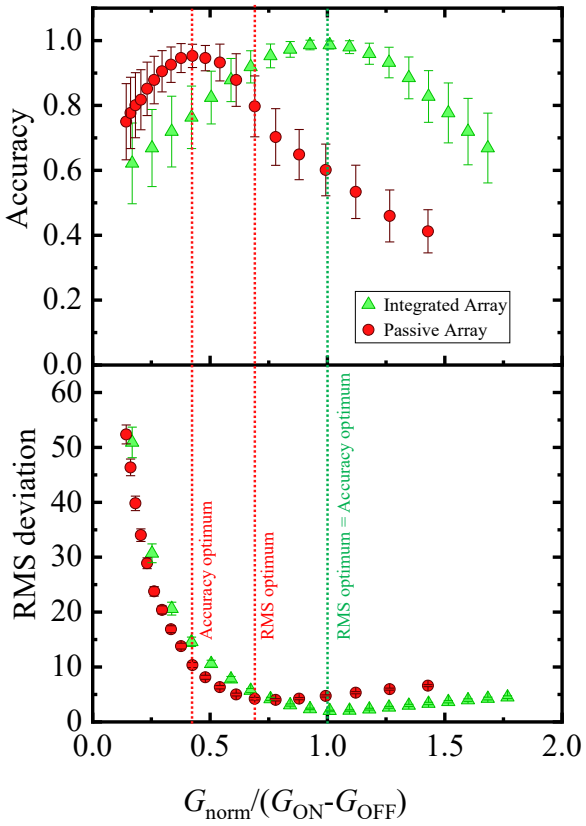

Fig. 1. Inference accuracy and RMS deviation between the ideal and written weight matrices as a function of the normalized G_{norm}, $G_{norm}/(G_{ON}$-$G_{OFF})$ for the passive MTJ array of [5], and the integrated array reported in this work. In the ideal case, $G_{norm} = G_{ON}$-G_{OFF}. Points and error bars represent the median and standard deviation, respectively, across 300 weight matrix solutions for each measurement.

In addition to a reduction in the impact of sneak paths, the integrated array shows a line resistance as low as 80 mΩ per square, two orders of magnitude less than the previous work. An average ON-state conductance of 106 μS and average OFF-state conductance of 47.5 μS results in an approximated G_{norm} = 58.5 μS. From the plots for both accuracy and RMS deviation, the approximated G_{norm} maximizes both parameters.

The overall accuracy of the network at an optimized G_{norm} is shown in Fig. 2. While the passive array shows a large variation in accuracies with a median of 95.3 %, the integrated array makes on average one more misclassification per solution and performs nearly as well as the software model at 98.7 % vs. 99.3 %.

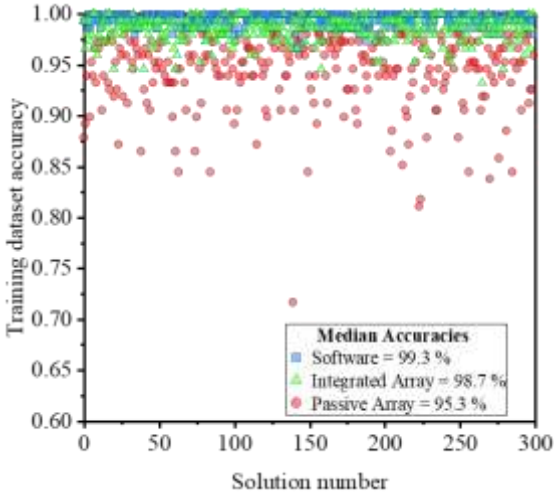

Fig. 2. Inference accuracy on the training dataset for all 300 weight matrix solutions when performed in software, previously reported passive array, and the integrated array reported in this work.

V. Conclusion

In this work we show that reducing line resistance and removing sneak paths by integrating with CMOS can drastically improve performance of hardware-based binary neural networks that can expedite the research and development of even larger networks based on MTJs.

References

[1] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics* vol. 8, pp. 661, Jun. 2019.

[2] S. Gao, *et al.*, "MRAM acceleration core for vector matrix multiplication and XNOR-binarized neural network inference," *2020 VLSI-TSA,* Hsinchu, Taiwan, Aug. 10-13, 2020.

[3] S. Jung, *et al.*, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature* vol. 601, pp. 211-216, Jan. 2022.

[4] P. Zhou, *et al.*, "Experimental demonstration of neuromorphic network with STT MTJ synapses," arXiv:2112.04749 [cs.NE], Dec. 2021.

[5] J. Goodwill, *et al.*, "Implementation of binary neural network on a passive array of magnetic tunnel junctions," *Phys. Rev. Appl.* vol. 18, 014039, Jul. 2022.

[6] UCI machine learning repository: wine data set, https://archive.ics.uci.edu/ml/datasets/wine.