Investigation of ReRAM Variability on Flow-Based Edge Detection Computing using HfO_2-Based ReRAM Arrays

Sarah Rafiq, *Student Member, IEEE*, Jubin Hazra, *Student Member, IEEE*, Maximilian Liehr, *Student Member, IEEE*, Karsten Beckmann, Minhaz Abedin, *Student Member, IEEE*, Jodh S. Pannu, *Student Member, IEEE*, Sumit K. Jha, *Member, IEEE* and Nathaniel C. Cady, *Member, IEEE*

***Abstract*—Resistive random-access memory (ReRAM) memristors are promising candidates for various compute in memory and flow-based computing approaches. As an alternative to traditional von Neumann computation, flow-based computing avoids serial movement of data between memory and processor. In this paper, we demonstrate arrays of 1 transistor 1 ReRAM (1T1R) to detect edges between 8 bit pixels using flow-based computing, and the effects of stochastic variation of ReRAM on edge detection outputs. Three different R_{off}/R_{on} resistance ratios (1.5:1, 2.5:1 or 28.6:1) were utilized to implement multiple flow-based edge detection computation matrices for 8 bit pixels. Edge detection was distinguishable for all R_{off}/R_{on} ratios used, for all flow-based computing matrices. However, the binary output resistance ratio of the matrices improved 3-fold when the patterned R_{off}/R_{on} ratio was increased to 28.6:1. A Gaussian simulation of ReRAM resistance variability validates the experimental data, with a correlation coefficient (r) of 0.9547. These results suggest a trade-off between the flow-based edge detection output ratio and the variability of the ReRAM resistance in R_{off}/R_{on} resistance ratio.**

***Index Terms*—HfO_2 ReRAM, 1T1R arrays, memristors, flow-based computing, edge detection, memory window, R_{off}/R_{on} ratio, multi-level resistance states.**

I. Introduction

DATASETS arising from sensors and devices on communications and sensing networks have increased exponentially and require extensive amounts of data storage and processing [1-6]. This is largely due to the advent of the internet of things (IoT) and edge-based computing. Previously, CMOS technology evolved rapidly with the development of digital systems on which conventional algorithms are executed. This approach makes use of the memory unit and the processing unit being in separate locations, in what is known as the von Neumann architecture. In this configuration, data must be sent back and forth between the memory and processing unit during computation. As the scaling of CMOS becomes limited, due to the approaching end of Moore's law [7] and finite bandwidth availability [1], these factors constitute an inherent bottleneck in the von Neumann architecture. This leads to the degradation of energy efficiency for performing computation [1-2, 4, 8].

Edge detection is a crucial and important operation that is leveraged in computer vision applications. Therefore, it is imperative to find alternative ways to carry out this elemental operation in low-power mode, especially for edge-based computing applications. Improvements in this field have been proposed utilizing new architectures in the digital domain and mixed-analog signals to accelerate computations [1]. Non-volatile memories (NVM), which operate in the analog domain, are attractive candidates to speed up elementary operations involved in these applications, due to their low energy consumption and high switching speed. Therefore, they are expected to boost the efficiency of conventional algorithms down the line [1, 4, 8-9].

Flow-based computing is one such approach that utilizes a two-dimensional non-volatile memory array to perform Boolean computations to perform in-memory computing, circumventing the von Neumann bottleneck. Although multiple flow-based computing designs have been proposed for edge detection applications, the effect of the stochasticity of Resistive Random Access Memory (ReRAM) arrays on the implementation of these computational designs are still lacking. In this work, we investigated the effect of different R_{off} and R_{on} resistance states used in holding the input variables on two-dimensional ReRAM arrays for flow-based computing, through experimental implementation of flow-based edge detection. Die-to-die electrical characterization of binary switching of ReRAM devices was carried out to obtain mean and standard deviation of the multi-level resistance states of fabricated HfO_2-based one transistor, one ReRAM (1T1R) cells. The impact of the variability of each R_{on} and R_{off} resistance states was investigated on the flow-based edge detection outputs.

Manuscript submitted October 19 2020; revised January 25 2021. This work was supported by the National Science Foundation under Award 1823015.

Sarah Rafiq, Jubin Hazra, Maximilian Liehr, Karsten Beckmann, Minhaz Abedin and Nathaniel C. Cady are with the College of Nanoscale Science and Engineering, SUNY Polytechnic Institute, Albany, NY 12203 USA (e-mail: srafiq@sunypoly.edu).

Jodh Singh Pannu and Sumit Kumar Jha are with the Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249 USA.

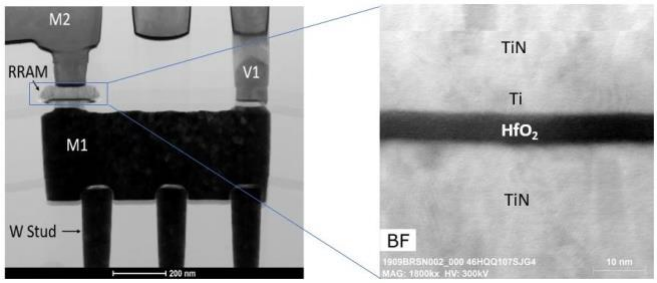


Fig. 1. Transmission electron micrograph (TEM) image of the fabricated 1-Transistor-1-ReRAM (1T1R) device. The right-hand side shows the cross-sectional TEM of the HfO_2 ReRAM device stack.

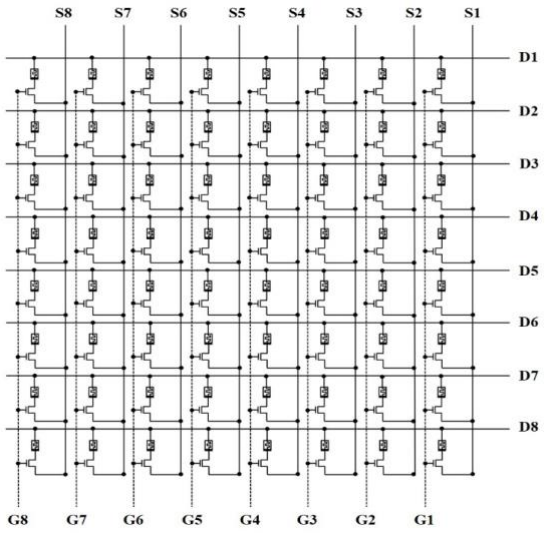


Fig. 2. Schematic of the fabricated 8x8 1-Transistor-1-ReRAM (1T1R) array. The eight source, eight drain and eight gate lines of the 1T1R devices are represented by the notations S1-S8, D1-D8 and G1-G8 respectively. Each device is accessed through its corresponding source, drain and gate lines.

II. Background and Related Work

A. Resistive Random-Access Memory

Memristive NVMs are typically two-terminal devices, including those that were proposed in 1972 by Leon O. Chua as the last fundamental passive circuit element, apart from resistors, capacitors and inductors [10]. Such a device retains the history of the applied electric field in terms of resistance and is therefore considered a form of memory. The device can be switched between high resistance state (HRS) and low resistance state (LRS).

One type of memristor, resistive random-access memory (aka: ReRAM) offers features such as non-volatility, scalability, and ease of fabrication. As such, these devices have been proposed for neural networks to hold weights as resistance states, as accelerators of mathematical computations such as the vector matrix multiplication, and to mimic biological synapses to help solve classification problems. Arrays of NVM, including ReRAM, have been used primarily in vector-matrix multiplication (VMM) in an analog fashion to carry out signal processing and solve classification problems [1-3, 11-12].

B. Edge Detection using Non-volatile Memory

Recently, non-volatile memories have also been used in edge detection [2, 4, 8-9]. The work of Mannion *et al.* demonstrated edge detection using a pair of ReRAM in a voltage divider configuration to detect the changes between frequencies, which was later correlated with variations in pixel intensities [9]. Although the method detected the frequency variations experimentally, the edge detection between neighboring pixels was done in a simulation model, made from extracted empirical data. Pajouhi *et al.* demonstrated edge detection using ant colony optimization algorithms through simulation as well [8].

Li *et al.* carried out edge detection using VMM on memristor crossbar array, where they supplied pixel intensities as voltages to an array mapped with conductance from a Discrete Cosine Transformation (DCT) matrix [2]. The array had a one transistor, one ReRAM (1T1R) architecture but was referred to as a memristor crossbar during the actual computation as all gates of the transistors were biased to give a passive array. However, an array of a much larger dimension, of 128 x 64 was used to implement edge detection, with trans-impedance amplifiers (TIA) used in their custom-made testing hardware.

In this work, we carried out an experimental demonstration of edge detection using flow-based computing [5]. Edge detection identifies the divider lines of different objects within an image. This work differs from [5] in that ReRAM variability was not considered in our previous work [5]. Our approach differs from conventional use of crossbar arrays such as VMM in that it uses the non-volatile memory array as a reconfigurable network to manipulate the flow of current through it, to give the output of edge detection. In this work, we present a novel experimental implementation of an edge detection between two pixels on an 8x8 1T1R HfO_2 array, where the mapping of the resistance states are determined by the input bits of the pixels in an unique flow-based computing design. This is carried out in 2 parts: (1) configure the array according to the input pixels by programming to target resistance states and (2) injecting the input read pulse for the flow-based computation and observing its output. Previously we have used this approach to implement a one-bit adder [6].

Work has been reported on the various designs for flow-based computing to detect an edge using memristor crossbars [4, 13], however, ReRAM is often regarded as a binary switch where it is assumed to completely block the current when the memristor is turned off, and all the off-state (high resistance state) memristors are expected to behave in a uniform manner. In contrast to ideal binary switches, physically implemented ReRAM devices exhibit rich dynamics in their resistance states, as well as device-to-device and cycle-to-cycle variability. Therefore it is critical to know the effect of ReRAM variability on flow-based computing designs. In this work, we have implemented flow-based computing edge detection algorithms using fabricated 1T1R memristors, and have demonstrated the effects of resistance variability and memory window (R_{off} vs. R_{on}) on edge detection accuracy.

III. Materials and Methods

A. 8x8 1T1R Arrays

Fully-integrated nanoscale CMOS/ReRAM structures were implemented on a 300 mm wafer platform using a custom ReRAM module within SUNY Polytechnic Institute's 65nm CMOS process technology. Figure 1 shows the transmission electron micrograph (TEM) image of the fabricated 1-transistor-1-ReRAM (1T1R) device, and the cross-sectional TEM of the ReRAM. As shown in Fig. 1, the ReRAM device stack is comprised of TiN top and bottom electrodes, a Ti oxygen exchange layer and a switching layer of HfO_2. The hafnium oxide-based ReRAM was integrated with a transistor at the Metal 1/via 1 (M1/V1) interface to form high density memory arrays. The size of the ReRAM is 100 x 100 nm^2. The switching layer of hafnium dioxide was deposited using atomic layer deposition (ALD) with a chlorine-based precursor at temperature of 300 °C. The excellent performance of these devices was presented earlier in [14]-[17]. The schematic of the fabricated 8x8 1T1R array is depicted in Fig. 2. In the array, top electrode lines are shared by the rows while the bottom electrode and transistor gate lines are shared by the columns. Each device is accessed through the gate of its integrated transistor, and its top and bottom electrodes. The arrays have demonstrated excellent yield of greater than 90 percent and therefore facilitated the testing of edge detection.

B. Electrical Characterization

Electrical characterization was carried out using an Agilent B1500 parametric analyzer and a B1530 module, which is a Waveform Generator / Fast Measurement Unit (WGFMU) for binary switching of the 1T1R devices. The edge detection computing was implemented using an Agilent E5270 parametric analyzer, a Keithley 707A switching matrix and a 2x12 probe card from Celadon Ultra High Performance Probe Cards. The probe card was used to access all the 24 pads of the schematic in Fig. 2 simultaneously for programming of the individual devices and testing of the flow-based edge detection.

C. Design of Flow-Based Computing for Edge Detection

Flow-based computing refers to creating specific flow (or path) of current in a two-dimensional array to carry out a logical computation. The logic 1 and logic 0 are represented by a ReRAM cell in the turned-on (Ron) state, and the turned-off (Roff) state, respectively. The concept is that when a ReRAM is in the LRS or Ron state, or holds a logic 1, it promotes the flow of current whereas a ReRAM is in a higher resistance state, logic 0, it prohibits the flow of current. As each 1T1R device is integrated between a row wire and a column wire, the applied current flows from the row to column or vice versa when the memristor is turned on. The ReRAM array is configured to hold the input variables in terms of logic 0 and logic 1. This manipulates the current flow when an input pulse is applied at input node. The Boolean output of a computation is measured as low or high resistance between the input and output nodes.

An edge detection computation kernel was developed using the flow-based computing approach. The edge detection design was generated by implementing an approximate ternary function on crossbar designs [4-5]. A pixel can have a gray-scale value between 0 and 255, and an edge between two pixels would exist if the difference between their gray-scale values exceeds a pre-defined threshold value. If the measured signal is above a given threshold, then is it considered as a logical high whereas if it is below the given threshold then it is a logical low. The term "pixel pair" used in this work denotes the pixel and its right and top neighbors in an input image. This helps detect horizontal and vertical edges in an image. The mapping of a given pixel pair onto the 8x8 RRAM crossbar is illustrated in [5]. Eight-bit pixels have been used in this work because a gray-scale pixel needs 8-bits to represent its complete range of values from 0 to 255. Decreasing this value might result in worse performance. On the other hand, increasing the bits beyond 8-bits will not result in a performance increase since in a grayscale image all possible values can be represented using 8-bits. Hence, all the extra bits will contain no new information.

An approximate ternary value function maps pixel pairs to true, false and a third value that has an uncertainty about the presence of an edge. The information about the presence or absence of an edge between a pixel pair, denoted as true or false values respectively, was obtained from human-annotated BSDS500 dataset [18]. Two conditions are set for the function to evaluate to true. Firstly, the frequency of an observed edge between a pixel pair, f1, needs to be above a threshold. Threshold is the minimum difference between the values of two gray-scale pixels needed to define them as part of an edge of an object in the image. Secondly, the ratio of f1 to the frequency of occurrence of same pixel pair in an image dataset, lies above a threshold. Otherwise, the pixel pair is mapped to a false value. In other words, the threshold value and the selective nature of the edge detection function has been calculated using the training image data-set from BSDS-500. The frequency of each unique pixel pair found in the image dataset was calculated and a threshold was chosen accordingly. Moreover, pixel-pairs that rarely occurred were ignored by the edge detection algorithm even if their difference exceeded the threshold value. This ensures that one-off pixel pairs occurring in detailed objects and images are not picked up by the algorithm. This method of creating a thresholding function that is selective helps mitigate errors caused by highly detailed images and results in smoother edges. All of the pixel-pairs have been tested through simulation, and the crossbar array designs have been used to detect edges on all images. The peak signal-to-noise ratio (PSNR) of the ground truth images versus the images produced by our designs on 8x8 arrays for edge detection is 13.7 dB [5].

The accuracy of the edge detecting algorithm is measured by comparing the signal-to-noise ratio and pixel difference of produced images with the ground truth provided in BSDS-500. The accuracy of this edge detection method on 8x8 arrays have been reported to be 72.8% [4]. Although our method used gray-scale images, the original images were converted from colored images to gray-scale before detecting the edges. Fig. 3(a) shows an image from the BSDS-500 dataset (which is colored) and Fig. 3(b) shows the edges produced from our edge detection method. The figures in Fig. 3(b) are produced using crossbar designs as mathematical functions. The ideal crossbar designs

produce either a true (ON) value or false (OFF) value without ambiguity. The performance of our algorithm was as good or better than the current state of the art using memristors [5]. Therefore, colored images can always be pipelined into our edge detecting algorithm by converting to gray-scale. It is worth noting, however, that one of the limitations of converting RGB color to gray-scale it may be possible for two RGB colors to have the same Y-value in YUV notation. Due to their same Y-value, this would make it difficult to detect the edge between the two RGB colors in the gray-scale value if YUV notation is used and is a limitation of gray-scale itself. The optimal crossbar design that implements this function was found using a massively-parallel simulated annealing search [4-5]. It makes sense to test designs that yield either a true or false value, rather than an uncertain value, to assess the functionality of the design. To verify the correctness of the simulation, multiple selected pixel-pairs were tested experimentally on 8x8 arrays. The patterned R_{off}/R_{on} ratio of 28.6 used in this work showed that there is sufficient margin for the multiple pixel pairs tested.

Fig. 3. (a) An image from the BSDS-500 dataset and (b) the edges produced

from edge detection using 8x8 ReRAM array.

In this work, the design was implemented as a proof of concept of the detection of an edge between two 8-bit pixels on an 8x8 1T1R array, and the functionality of the design was analyzed with respect to the different combinations of binary resistance states used in pattering the array. The inherent variability associated with each of these resistance states are due to the characteristics of HfO_2 ReRAM arrays.

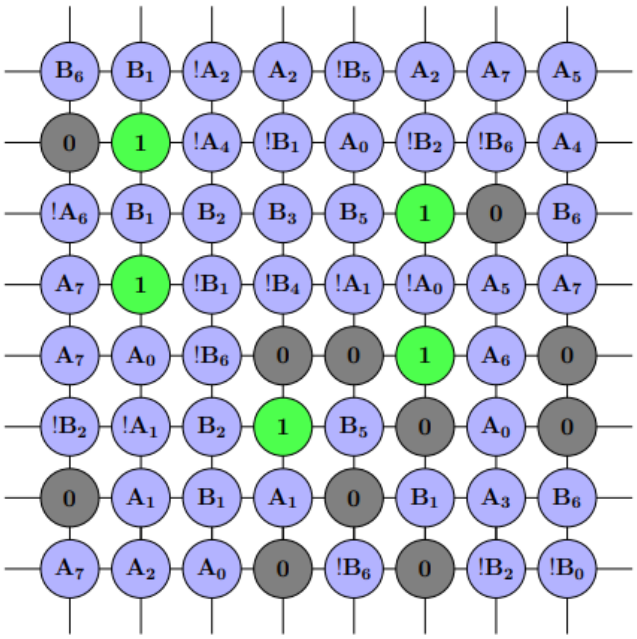


Fig. 4. The 8x8 array design for edge detection between two 8-bit pixels. The bits of the two pixels, are represented by variables A0-A7 and B0-B7. "!" is used to represent the negation of the variables A0-A7 and B0-B7. The cells at 1 and 0 remain fixed at Ron and Roff resistance states respectively.

D. Methodology of Flow-Based Edge Detection Experiment

The implemented edge detection design is shown in Fig. 4. The bits of the two pixels are represented by variables A0-A7, and B0-B7 respectively. A0 and B0 denote the most significant bits of the two pixels, while A7 and B7 denote the least significant bit of the two pixels. The design consists of literals of these variables and their negations. The negations are represented by "!" preceding the input variables A0-A7 and B0-B7. The rest of the cells in Fig. 4 do not depend on the input variables. These cells are assigned a permanent logic 1 or logic 0, irrespective of the input configuration and are intended to lower the number of programmed cells per iteration. Each element in the design is represented by the binary resistance state of a 1T1R cell, in an 8x8 1T1R array as that shown in Fig. 2.

In order to carry out flow-based computing successfully, the steps involved in the implementation of the flow-based edge detection design is outlined in the flow chart in Fig. 5. The first step involves generating a Boolean design consisting of the input variables, input bits A0-A7 and B0-B7, that can be configured in the array to hold the current inputs. The generated logic 1 and logic 0 are mapped to Ron resistive states and Roff resistive states respectively, on the 8x8 array. Then, a read and

verify scheme was used to verify the correctness of the patterned resistance. Finally, an input signal (read voltage or current) was applied to a specific nanowire and the output is

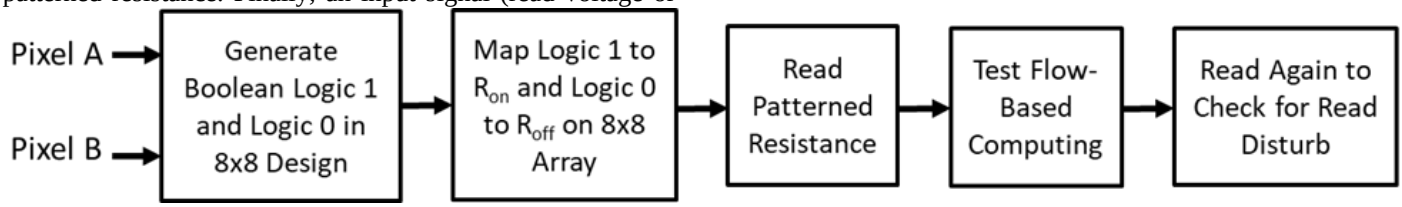


Fig. 5. The flow-chart of the sequence of steps required in the implementation of flow-based edge detection design.

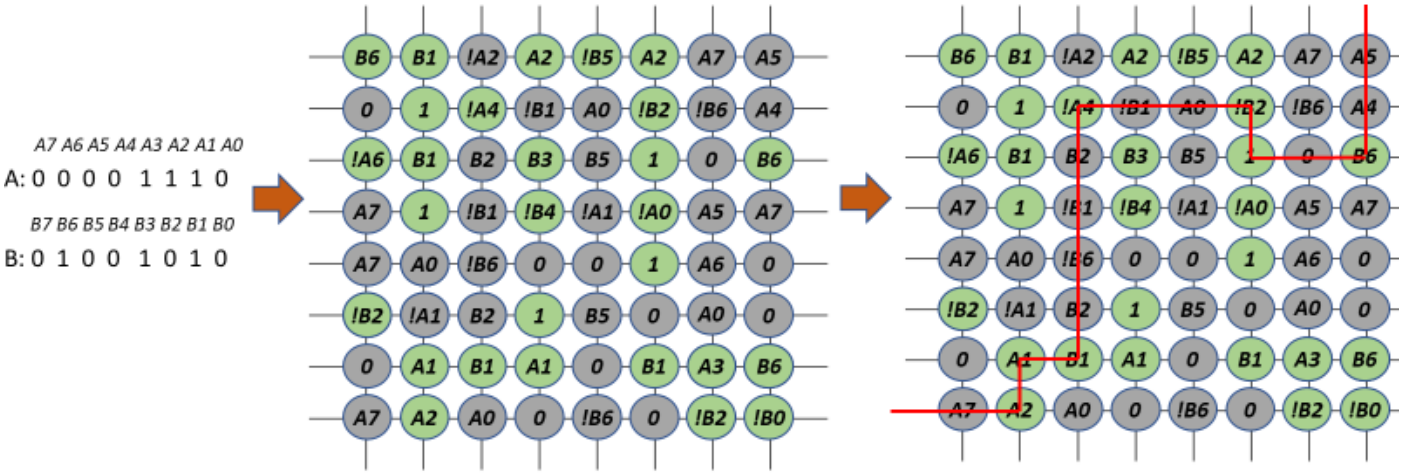


Fig. 6. The mapping scheme of the input bits of two pixels, pixel A and pixel B, onto the 8x8 1T1R array and the subsequent flow-based computation. The green and gray cells denote turned-on and turned-off memristors respectively. The programmable cells change with the 16 input bits, while the non-programmable cells remain fixed at logic 1 and logic 0. After the mapping, an input voltage pulse is applied to bottom row and output is observed along last column. The resistance between the input and output nanowires was extracted by measuring current flow at the column nanowire.

observed along a different nanowire during the computation. In all of these steps, the current is sensed by source measurement unit (SMU) of the parametric analyzer E5270. A current compliance of 1 mA is also specified for the SMU measuring the current. The output signal holds the result of the computation being performed in terms of a Boolean high or low output, according to the edge detection algorithm described in section III. C, which is *a priori*. The logic 1 output corresponds to the presence of an edge, while a logic 0 output corresponds to the absence of an edge. To validate the Boolean result in section III. C experimentally, edge detection was investigated on actual RRAM devices with different RRAM off/on resistance states to assess the impact of RRAM device characteristics on the output signal. The computation is then followed by another read and verify to check for read disturb. As the resistance that is patterned dictates the flow of current during the computation, the second read/verify step ensures the patterned resistance have stayed the same during the computation to give the correct readout. A read voltage of -100 mV was used for the computation. However, this second read and verify step is optional and may be skipped to reduce power consumption.

A hypothetical mapping scheme for detection of an edge between the input pixels, A=00001110 and B=01001010 is illustrated in Fig. 6. The gray circles represent turned-off memristor while the green circles refer to turned-on memristor. As edge detection was investigated with different patterned Roff/Ron ratios, the gray circles were programmed with different resistance states for each ratio. For all patterned R_{off}/R_{on} resistance ratios, LRS 1T1R elements, represented by green circles, were programmed using a set voltage of 2 V and a current compliance of 245 µA. For the highest R_{off}/R_{on} resistance ratio, HRS 1T1R elements, represented by gray circles, were programmed using a reset voltage of -1.5 V and a current compliance of 140 µA. This set of conditions was chosen for reset because lower current compliance yields higher HRS states [14]. Additionally, for the lower R_{off}/R_{on} resistance ratios, the gray circles were programmed with a set voltage of 2 V and current compliances of 60 µA and 110 µA, to yield lower resistance states for the gray circles. As RRAM cells hold their state, cells with permanent logic 1 and logic 0 in Fig. 6 were programmed once and did not need to be re-programmed when the bits of pixels changed with each new set of input pixels. The cells that hold the input variables are re-programmed if their bit changes with the input pixels. After the mapping scheme, the computation was carried out by applying a read pulse to the bottom row of the design and the output is observed along the last column.

The red line shown in Fig. 6 is one of many possible sneak

paths in the array. As there are multiple sneak paths in the array, the paths will not be unique. The number of sneak paths existing do not affect the outcome if they do not connect the input to the output. Hence, only the sneak paths that connect the input to the output are significant for our computation. During the computation, all the gates of the transistors were fully biased to yield a crossbar array. The current flow observed between the input and output nodes when a read voltage is applied, denotes whether an edge between the two pixels was present or not. A low current measurement, which corresponded to the high resultant resistance readout, implied the absence of an edge, while a high current measurement implied the presence of an edge between the two pixels.

E. Simulations of Flow-Based Edge Detection Computing

To validate the experimental results from flow-based computing, simulations of the 8x8 array elements were carried out in LTSpice software for two scenarios, the ideal case and the non-ideal case. Firstly, we showed the effect of an ideal case for the different 8-bit inputs if no resistance variation were present in the resistance states. This was done by simulating an 8x8 array of resistors with fixed binary resistance states, which were the extracted average LRS and HRS resistance states used in experiment. Secondly, the characterization of HfO_2 devices revealed that different resistance states exhibit different variations, with HRS exhibiting the greatest variation in resistance. The variability in each resistance state was extracted

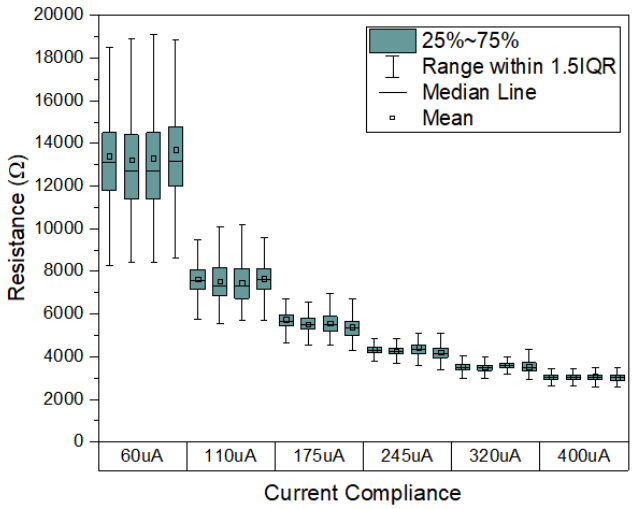


Fig. 7. Box-and-whisker plot for the die-to-die resistance variation in 8x8 1T1R array devices across a 300 mm wafer, with respect to current compliance (by varying Vg of the control transistor). The devices were switched for over 10,000 cycles at each current compliance. The standard deviation of the LRS resistance decreased with increasing current compliance.

from experimental characterization of HfO_2 1T1R arrays, and this was incorporated into Gaussian distributions within 3 standard deviations of respective resistance variability for each resistance state. The Gaussian simulation was run for 200 cycles in LTSpice, and the resultant resistance readout for each edge detection output was compared for three different resistance states used for the R_{off} patterned resistance.

TABLE I

THE VALUES OF PIXEL A AND PIXEL B IN EACH OF THE FIVE PATTERNS ARE SHOWN BELOW. THE EXPECTED OUTPUT LOGIC OF 1 CORRESPONDS TO THE PRESENCE OF AN EDGE BETWEEN THE 2 PIXELS AND IS EXPECTED TO CONTRIBUTE TO LOW RESISTANCE IN THE EDGE DETECTION TEST. SIMILARLY, THE EXPECTED OUTPUT LOGIC OF 0 IMPLIES THE ABSENCE OF AN EDGE AND THEREFORE WOULD RESULT IN HIGH RESISTANCE IN THE OUTPUT OF EDGE DETECTION TEST. THE PATTERNS CONSIST OF 3 LOW RESISTANCE PATTERNS AND 2 HIGH RESISTANCE PATTERNS.

Edge Detection Pattern #	Pixel A	Pixel B	Expected Output Logic	Expected Resistance
1	00001110	01001010	1	Low
2	00110101	01001010	1	Low
3	10110111	01110000	1	Low
4	01101110	01111111	0	High
5	01010010	11101011	0	High

IV. RESULTS AND DISCUSSION

A. Measurement of ReRAM Resistance Variability

In order to efficiently perform computations on resistive memory arrays, the programming of multiple resistance states should be performed with reasonable accuracy. The role of integrated transistor on-chip is significant in decreasing the variability of resistance states programmed during the SET operation of the device, compared to arrays of nonlinear memristors [2] and arrays without selectors [21]. In the 1T1R configuration, the transistor can be used to set current compliance, helping to control the desired resistance state of the ReRAM before it reaches the SET switching dynamics [21]. Device-to-device repeatability was assessed through the accurate and repeatable programming of six distinct resistance states on multiple 8x8 1T1R arrays, using an endurance over 10,000 cycles, across multiple die on a 300 mm wafer. The results for the die-to-die resistance is shown in Fig. 7.

The multiple resistance states were achieved by a write and verify scheme, where the current compliance during the SET operation is modulated in the range of 60 µA to 400 µA, through the gate voltage of the control transistor of the device. Different current compliance values gave rise to distinct states across multiple die, with the higher resistance states having greater variability. The applied set and reset voltages were 2 V and -1.5 V respectively, with a rise/fall time of 10 µs. The hold time of each programming pulse is 0 seconds. The standard deviation of the resistance states was suppressed from 2 kΩ to less than 200 Ω, with a five-fold change in mean resistance from 15 kΩ to 2.8 kΩ. This is explained as follows. As the gate voltage is increased, the NFET transistor lets more current through the memristor and increases its current compliance. This leads to an increase in the number of oxygen vacancies generated in the conductive filament and therefore widens the filament's cross-sectional area. As the conductive filament becomes Ohmic in nature during the SET operation, its resistance, which is inversely proportional to its cross-sectional area, decreases with increasing current compliance. However, at the higher current compliance, the resistance saturates due to the finite size of the conductive filament [20]. Based on the results from Fig. 7, a current compliance of 245 µA was used to program the lower resistance state (logic 1) in the pattern as

this resistance state was still close to that of 400 µA due to the plateau observed in resistance at higher current compliance.

The transistor's resistance variation may contribute to the low resistance states (LRS), but the transistor's resistance variation is irrelevant in the high resistance states (HRS). This is because HRS variability is dominated by RRAM. For LRS transistor resistance variation, we explain the following. During SET operation, the current compliance provided by the transistor controls the size of conductive filament of the LRS state. With lower current compliance, there are fewer oxygen vacancies in the filament, which results in more resistance variation of the LRS state. Therefore, as the measurements were performed at room temperature, the LRS resistance variation due to the size of the conductive filament may prevail over the transistor's resistance variation. For a comparison of the 1T and 1T1R resistance variation, we have characterized the transistor resistance and its standard deviation at the same gate voltage of the NFET that provided current compliance of 60 µA in Fig. 7. For this gate voltage of the NFET, the average 1T1R resistance was 15 kΩ with a standard deviation of 2000 Ω. The transistor resistance is on average 2.4 kΩ with a standard deviation of 275 Ω. Hence, the transistor contributes very little to the resistance variation observed in the 1T1R LRS resistance state for current compliance of 60 µA in Fig. 7.

Although having transistors integrated with each ReRAM device increases the footprint of the array, they have an important role in making the flow-based computing possible, which is outlined as follows. Apart from serving as a current limiter, transistors block so-called current sneak paths in the array during the programming and reading of a 1T1R cell. Sneak current is an issue in memristor crossbar arrays without access devices, where the input signal flows through undesired path(s) of memristors with lower resistance, in parallel to the signal flowing through the device being programmed [19]. If a current path through three shorted memristors exists, then the input signal would bypass the device being programmed. This yields an incorrect reading of the programmed resistance state [19], and may also cause disturbance to the resistance states of the neighboring memristors during programming [12]. Therefore, using transistors as the access devices in an array is essential for the accurate programming and reading of a memristor cell. Furthermore, connecting an external transistor with crossbar arrays to provide current compliance gives rise to significant parasitic capacitance. This issue can be mitigated by having a transistor integrated on-chip. The on-chip transistor also helps to limit the overshoot current during reset operation and keeps it below the compliance current, hence giving a better control of compliance [20].

Although the low resistance states can be accurately tuned with access transistors, the high resistance states exhibit much greater variability and cannot be properly tuned to narrow resistance ranges like the low resistance states. The standard deviation in the HRS states can be as high as 25.3 kΩ for our device stack [14]. These variations are an intrinsic property of metal-oxide ReRAM due to the stochastic atomic motion under valence change mechanism (VCM) switching [22].

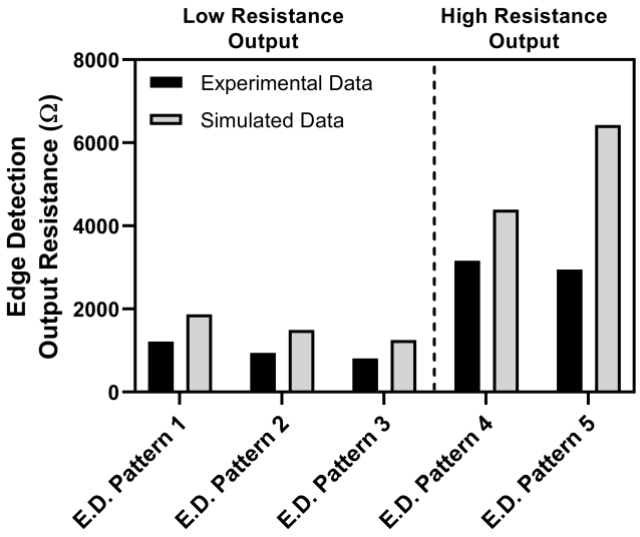


Fig. 8. Comparison of experimental edge detection outputs of 5 pixel-pairs (each pattern denotes a pixel-pair) with their ideal case LTSpice simulation using HRS and LRS patterned arrays. The result confirms that an edge is detected in the first three pixel pairs, and no edge is present between the last 2 pixel-pairs. The results were validated by LTSpice simulation assuming HRS of 100 kΩ and LRS of 3.5 kΩ, with no resistance variation.

B. Implementation of Edge Detection Algorithm on 8x8 1T1R Arrays

The edge detection design for flow-based computing needs the input to be mapped as binary resistance states. The performance of the design was validated by determining the existence of an edge between five pairs of 8-bit pixels. The values of each pixel-pair and their expected resistance and binary output is shown in Table I. For convenience, each pixel-pair will be referred to by their respective pattern number in the following sections. The implementation of the first pixel pair is illustrated in Fig. 6. The remaining pairs were also implemented in a similar fashion. An edge is expected between the first three pairs of pixels, indicated by low resistance readout corresponding to logic 1. Meanwhile, the absence of an edge is expected for the last two pixel-pairs, which would be indicated by the high resistance readout corresponding to logic 0.

The five patterns were implemented on the 8x8 1T1R arrays to validate the functionality of the flow-based computing edge detection design. The arrays were patterned with HRS as logic 0 and an average LRS of 3.5 kΩ as logic 1 in the experiments. This gives a R_{off}/R_{on} patterned resistance ratio of ~28.6:1. The binary outputs were further validated by LTSpice simulation. The empirical outputs from the edge detection computation of the five patterns are reported in Fig. 8, along with their expected resistance readouts in an ideal case (without variation) from LTSpice simulation. The LTSpice simulation assumed the HRS and LRS values to be 100kΩ and 3.5 kΩ, respectively, in each of the five patterns and gave their expected resistance readout. The expected results from simulation of the 8x8 array conform to the empirical results for all five pairs of pixels. The first three pixel pairs yielded lower resistance readout than the final two pixel pairs. The binary outputs in the experiment are also distinguishable, with a

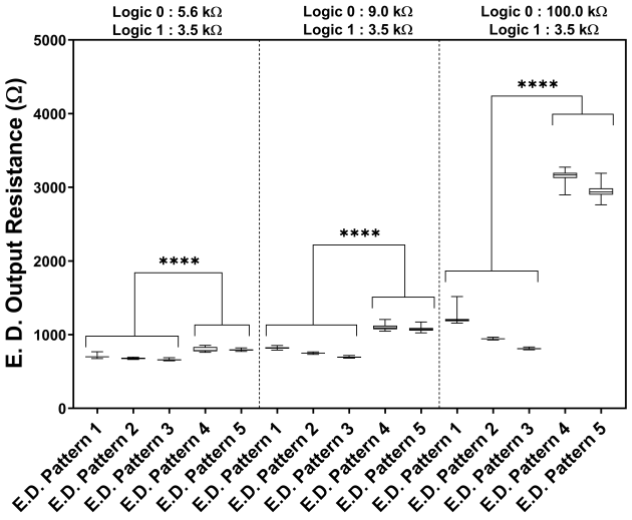


Fig. 9. Effect of three different patterned Roff/Ron (Logic 0 / Logic 1) ratios on the binary outputs of edge detection of 5 patterns (3 low and 2 high patterns) is shown for over 50 cycles. The patterned resistance R_{off}/R_{on} ratios of 1.5:1, 2.5:1 and 28.6:1 correspond to logic 0 being programmed with 110uA, 60uA and HRS respectively, shown from left to right. One-way ANOVA on the three datasets show that the high resistance and low resistance patterns can be distinguished for all three conditions. However, HRS used as logic 0 gives the best ratio of 3.08:1 between the binary outputs, as opposed to 1.16:1 and 1.44:1 for the first two conditions. The variability of the high resistance patterns also increased with increasing logic 0 patterned resistance.

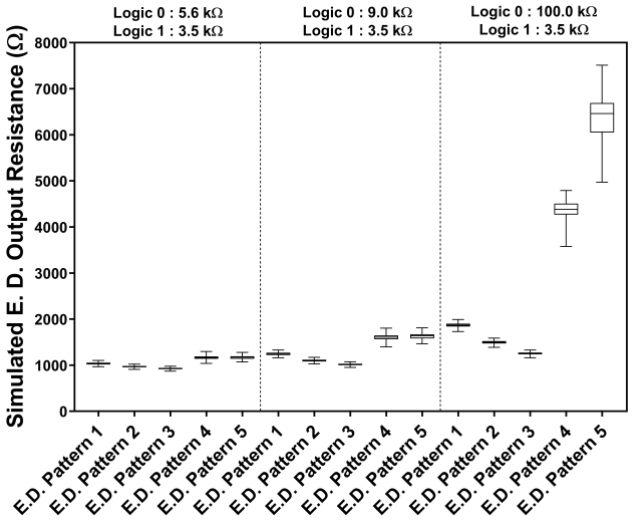


Fig. 10. LTSpice Gaussian simulation of the variability of the high resistance and low resistance patterns for the different patterned Roff/Ron ratios of 1.5:1, 2.5:1 and 28.6:1 using the respective standard deviations of the patterned Ron and Roff resistance states. The greatest high/low output resistance ratio also has the highest variability in overall high resistance and low resistance patterns. Hence, there is a tradeoff between the variability of high resistance and low resistance patterns, and the high/low binary output resistance ratios.

high/low output ratio of ~3:1. The resistance difference between experimental and simulated results of the fifth pixel-pair may be attributed to the inherent variability of ReRAM devices.

C. Effect of ReRAM Roff/Ron Ratio and their Respective Resistance State Variability on Flow-Based Computing

As ReRAM is known to exhibit significant variability in its resistance states (which we demonstrated in Fig. 7), three different ratios of patterned binary resistance states were evaluated on the 8x8 array, while keeping the programming conditions of logic 1 the same. In the three cases, the logic 0 were programmed to be 6 kΩ, 9 kΩ and 100 kΩ while their corresponding logic 1 were about 3.5 kΩ respectively. This yielded three different R_{off}/R_{on} patterned ratios of 1.5:1, 2.5:1 and 28.6:1 respectively. The logic 1 was programmed with LRS state at current compliance of 245 µA. For the low R_{off}/R_{on} patterned resistance ratios, the logic 0's were programmed with LRS states at current compliance of 60 µA and 110 µA . The highest R_{off}/R_{on} ratio has HRS state as logic 0. The low resistance states have a much lower variability than high resistance states due to the Ohmic nature of the current-voltage characteristic of the device. The five patterns were repeated for the three patterned ratios for over 50 cycles to investigate the repeatability of the edge detection outputs. The respective edge detection binary outputs from the three different patterned R_{off}/R_{on} ratios of 1.5:1, 2.5:1 and 28.6:1, on the detection of an edge between 5 pixel pairs are shown for over 50 cycles in Fig. 9. The upper and lower caps of each box plot represent the minimum and maximum output resistance. There are two important observations from Fig. 9.

First, the 5 patterns (each containing a pixel-pair) consist of three low binary outputs and two high binary outputs. An ordinary one-way ANOVA on the three patterned R_{off}/R_{on} ratios reveal that there is significant difference between the high and low binary outputs. Hence, the binary outputs can be distinguished for all three patterned R_{off}/R_{on} ratios. However, the high/low output ratio of the flow-based computing improves by three-fold from 1.16:1 to 3.08:1 when the highest programmed resistance is used as logic 0 in the pattern. This is because the lower R_{off}/R_{on} patterned resistance ratios significantly degrade the binary outcome of pixel pairs that do not have an edge (a high binary output), where a high resistance readout is expected. This is possibly caused by the overall resistance drop due to the decreased resistance in each of the multiple sneak paths that are created in the computation. This result shows that a high R_{off}/R_{on} patterned resistance ratio is vital for the successful implementation of the edge detection design.

Secondly, the variability of individual low and high binary outputs increases with increasing resistance state used for logic 0. This is possibly due to the increasing variability associated with R_{off} used for logic 0, since the resistance state of logic 1 was maintained for all the three R_{off}/R_{on} ratios. In order to assess this hypothesis, a Gaussian LTSpice simulation was carried out for the five patterns in the next section. The pattern number 4 used for the lowest R_{off}/R_{on} ratio seemed to have high variability due to the result consisting of multiple arrays for the 50 cycles, where some arrays have higher resistance readouts due to processing conditions.

Regarding power consumption of the flow-based computation, a read voltage is applied to obtain the edge detection output. The time needed for output to stabilize would depend on the testing equipment or read-out circuit that is employed. For the work presented herein, it would take a

minimum of 1us for the edge detection output to stabilize. Hence, the power of computation for highest R_{off}/R_{on} ratio would be about 10 pJ for the low resistance edge detection outputs, and 3.3 pJ for the high resistance edge detection outputs. As for programming the array to hold the data, we have used pulses of 10 us rise/fall time, which consumes 480 pJ-850 pJ. However, we have demonstrated the ability to program our 1T1R ReRAM cells with 100 ns rise/fall time, which would reduce energy for programming at 100 uA SET to 20 pJ. Thus, in the worst case scenario, this approach would consume ~960 pJ in programming 48 of the devices in the array for each new pair of 8-bit pixels.

D. Benchmarking

The impact of variability of the Roff and Ron resistance states on the five patterns were assessed using Gaussian LTSpice simulation of the patterns with the respective R_{on} and R_{off} standard deviations extracted from experimental data. For the case where HRS is used as logic 0, the standard deviation for HRS was taken as 0.5 of the mean HRS value of 100 kΩ. The Gaussian simulation results for the three patterned R_{off}/R_{on} ratios on the outputs of edge detection computation is shown in Fig. 10. The ratio of the edge detection outputs between the high resistance and low resistance patterns improved significantly with increasing patterned R_{off}/R_{on} ratio due to greater absolute resistance for high resistance patterns, as observed from experiment. Also, the variability of the resistance states used for R_{off} clearly increased the variability of the edge detection outputs for all five patterns, where the outputs of patterns 4 and 5 give the most variability when HRS is used for logic 0. This is also consistent with the experimental data. However, the absolute resistance readouts of patterns 4 and 5 appear to be much higher. Although there is no change in HRS resistance after a random HRS value from the Gaussian distribution is used in simulation, the HRS resistance in the experiments could be reduced partially by a few kΩ (still in the HRS regime) due to sneak paths in the array rubric. This may have resulted in lower experimental resistance outputs for these patterns. The read noise is normally higher for HRS regime which could have also contributed to the experimental data. Nevertheless, there is a trade-off between the edge detection output ratio and the variability associated with the five patterns in both experiment and simulation.

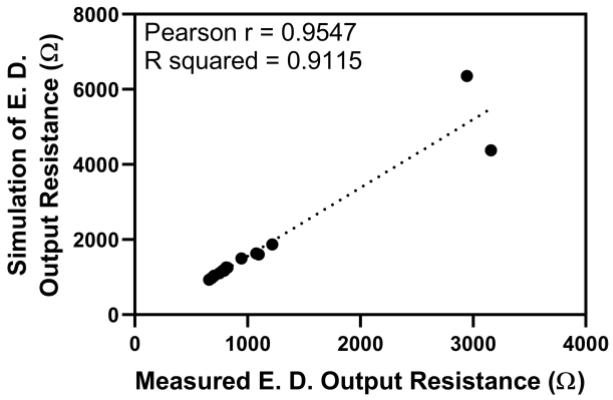


Fig. 11. The Pearson correlation coefficient of the experimental edge detection outputs for 5 patterns and their LTSpice Gaussian simulation for the three patterned Roff/Ron ratios of 1.5:1, 2.5:1 and 28.6:1. The coefficient r of 0.9547 shows a strong positive correlation between the experimental data and the simulation.

To get more insight of how the Gaussian simulation results compare to the experimental data for the five edge detection patterns, a Pearson correlation coefficient was extracted from the results for the three patterned R_{off}/R_{on} ratios of 1.5:1, 2.5:1 and 28.6:1. The x-axis of Fig. 11 shows the average of 50 cycles endurance experiment for edge detection patterns (from Fig. 9). The y-axis shows the average of 200 Gaussian simulation cycles (from Fig. 10). The correlation between the experimental data and the simulated data is shown in Fig. 11. The correlation coefficient r of 0.9547 shows a strong positive correlation between the experimental data and the simulation. Due to higher resistance variation of HRS used as logic 0, the data for patterns 4 and 5 for the highest R_{off}/R_{on} ratio of 28.6:1 have drifted from the simple linear regression line. This is because the experimental results average is lower than the simulated average of 200 cycles. This could be attributed to the fact that patterns 4-5 have higher variability than patterns 1-3 for 28.6:1, and patterns 1-5 from other R_{off}/R_{on} ratios (1.5:1, 2.5:1). As mentioned previously, this may be caused by the reasons outlined in previous paragraph. The rest of the data in Fig.11 follows the regression line due to the lower variability present in their edge detection output resistances.

In terms of application of flow-based edge detection using HfO_2 ReRAM arrays, one needs to take into account the binary high/low edge detection output ratio and their associated variability. For instance, low edge detection output ratio may be tolerated by systems with high sensitivity, as it would benefit from the more consistent binary outputs. On the other hand, applications that are more robust would require better distinction between the edge detection outputs (i.e. higher output ratio), where a loss in accuracy due to greater variability of binary outputs is acceptable. Improvements in the processing conditions of HfO_2 ReRAM can help reduce the variability of the high resistance state to be used for logic 0, which would lower the variability of the edge detection outputs to give the best results for flow-based edge detection computing. Alternatively, changing the switching material of the metal-oxide ReRAM to tantalum oxide for instance, that has a higher memory window (R_{off}/R_{on} ratio) may also improve the performance of flow-based edge detection computing.

For future work on flow-based edge detection, optimization of circuit architecture is needed for real-time processing. These circuit architectures could include a combination of 8x8 array modules, with shared DACs and ADCs with multiplexers, and circuitry to measure flow-based current at the output node, which could be shared between the array modules as well. The system could be controlled using FPGA. For the output node circuitry, a multiplexer (mux) may be used to interface the output node of array to either trans-impedance amplifier (TIA) and ADC to measure current when programming devices, or to TIA and a threshold comparator (could be an op-amp comparator circuit) with a reference voltage to give high/low

Boolean output of edge detection for flow-based computing. As such, flow-based computing could be performed using the same biasing scheme used in this work for 8x8 RRAM array modules.

An alternative approach that could be used for real-time processing would be to leverage a large 1T1R array/ crossbar array, where 8x8 sub-arrays can be used for flow-based edge detection. This would require testing systems that can provide multiple driving voltages, along with probe cards, switching matrix, a microcontroller to control the signals applied to arrays, ADCs and DACs which could be shared, and threshold comparators.

V. Conclusions

Metal oxide ReRAM has shown great promise in flow-based computing to overcome the bottleneck of von Neumann architecture. Edge detection between pairs of 8-bit pixels has been implemented using flow-based computing on 8x8 1T1R HfO_2 arrays, and the effect of different HfO_2 resistance states and their variability on edge detection computing is investigated. Flow-based computing requires the reconfiguration of an array according to a design consisting of Ron (the lower resistance state) and Roff (the higher resistance state). The functionality of the flow-based edge detection design was investigated using different patterned R_{off}/R_{on} ratios, and the respective variability of these Ron and Roff resistance states used in the pattern. Arrays of 1T1R were first characterized for binary switching, and optimal programming conditions were suggested for the Ron resistance state, and two of the Roff resistance states in the pattern, apart from HRS. The arrays were patterned with R_{off}/R_{on} ratios of 1.5:1, 2.5:1 and 28.6:1 for all the pixel-pairs and repeated for over 50 cycles to assess the variability of these binary edge detection outputs. A one-way ANOVA on the experimental data shows that the binary low and high edge detection outputs are distinguishable for all the patterned R_{off}/R_{on} ratios. However, the results show that the ratio of binary high/low edge detection outputs improve by three-folds from 1.16:1 to 3.08:1 when the patterned resistance ratio is increased from 1.5:1 to 28.6:1. The variability of all the binary outputs also increases with the variability of the patterned R_{off} resistance state. A Gaussian simulation of the respective variability of each resistance state used in pattern validates the experimental data with a correlation coefficient r of 0.9547. Therefore, there is a trade-off between the flow-based edge detection output ratio and the variability of the edge detection outputs. The trade-off may be improved by changes in processing conditions of HfO_2 ReRAM stack to reduce HRS variability, or using a different metal oxide ReRAM that has greater R_{off}/R_{on} ratio and less HRS variability, to lower the variability of edge detection outputs and increase their binary output ratio.

References

[1] M. Hu et al., "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication", in *Proc. 53rd Annu. Design Autom. Conf. (DAC)*, Austin, TX, USA, 2016, pp. 1-6, DOI: 10.1145/2897937.2898010.

[2] C. Li et al., "Analogue signal and image processing with large memristor crossbars", *Nature Electron.*, vol. 1, no. 1, pp. 52-59, Jan. 2018. DOI: 10.1038/s41928-017-0002-z.

[3] D. Lelmini, H. -S. P. Wong, "In-memory computing with resistive switching devices", *Nat. Electron.*, vol. 1, pp. 333-343, Jun. 2018, DOI: 10.1038/s41928-018-0092-2.

[4] J.S. Pannu et al., "Data-driven Approximate Edge Detection using Flow-based Computing on Memristor Crossbars," in *IEEE Albany Nanotechnology Symp. (ANS)*, Albany, NY, USA, 2019.

[5] J. S. Pannu et al., "Design and Fabrication of Flow-Based Edge Detection Memristor Crossbar Circuits1," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 67, no. 5, pp. 961-965, May 2020, doi: 10.1109/TCSII.2020.2984155.

[6] Z. Alamgir, K. Beckmann, N. Cady, A. Velasquez, and S. K. Jha, "Flowbased computing on nanoscale crossbars: Design and implementation of full adders," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2016, pp. 1870–1873.

[7] M. M. Waldrop. "The chips are down for Moore's law", *Nat.*, vol. 530, pp. 144-147, 2016.

[8] Z. Pajouhi and K. Roy, "Image edge detection based on swarm intelligence using memristive networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 9, pp. 1774–1787, Sep. 2018, DOI: 10.1109/TCAD.2017.2775227.

[9] D. J. Mannion, A. Mehonic, W. H. Ng, and A. J. Kenyon, "Memristor-Based Edge Detection for Spike Encoded Pixels," *Front. Neurosci.*, vol. 13, pp. 1-11, Jan. 2020. DOI: 10.3389/fnins.2019.01386.

[10] L. Chua, "Memristor-The missing circuit element", *IEEE Trans. Circ. Theory*, vol. 18, pp. 507-519, 1971.

[11] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors", *Nat.* , vol. 521, pp. 61-64, May 2015.

[12] F. Merrikh Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva and D. Strukov. "Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits", *Nat. Comm.*, vol. 9, pp. 1-7, 2018, doi: 10.1038/s41467-018-04482-4.

[13] S. Khokhar, and A. Khalid, "Nanoscale memristive crossbar circuits for approximate edge detection in smart cameras," in *2018 IEEE 9th Annu. Info. Tech., Electron. and Mob. Comm. Conf. (IEMCON)*, Vancouver, BC, Canada, 2018, pp. 749-754.

[14] J. Hazra, M. Liehr, K. Beckmann, S. Rafiq and N. Cady, "Improving the Memory Window/Resistance Variability Trade-Off for 65nm CMOS Integrated HfO2 Based Nanoscale RRAM Devices," in *2019 IEEE Intl. Integ. Reliability Workshop (IIRW)*, South Lake Tahoe, CA, USA, 2019, pp. 1-4, DOI: 10.1109/IIRW47491.2019.8989872.

[15] K. Beckmann et al., "Towards synaptic behavior of nanoscale ReRAM devices for neuromorphic computing applications", *ACM J. Emerg. Technol. Comput. Syst. (JETC) Special Issue on New Trends in Nanoelectronic Device, Circuit and Archit. Design*, vol. 16, no. 2, Apr. 2020, DOI: 10.1145/3381859.

[16] M. Liehr et al., "Fabrication and Performance of Hybrid ReRAM-CMOS Circuit Elements for Dynamic Neural Networks", in *Proc. Intl. Conf. on Neuromorphic Syst. (ICONS '19). Assoc. Comput. Machinery*, New York, NY, USA, 2019, pp. 1-4.

[17] K. Beckmann, J. Holt, H. Manem, J. Van Nostrand and N.C. Cady, "Nanoscale hafnium oxide RRAM devices exhibit pulse dependent behavior and multi-level resistance capability", *MRS Advances*, vol. 1, no. 49, pp. 3355-3360, 2016.

[18] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. Eighth IEEE Intl. Conf. Comput. Vision. ICCV 2001*, Vancouver, BC, Canada, 2001, pp. 416-423, vol. 2, DOI: 10.1109/ICCV.2001.937655.

[19] J. J. Yang and R. S. Williams, "Memristive devices in computing system: Promises and challenges," *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 9, no. 2, May 2013.

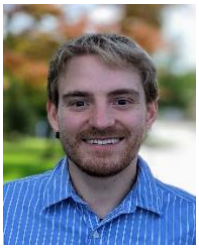
[20] G. Bersuker, D.C. Gilmer, D. Veksler, "Metal-oxide resistive random access memory (RRAM) technology: Material and operation details and ramifications," in *Advances in Non-Volatile Memory and Storage Technology*, 2nd ed., 2019 ch. 2, pp. 35-102. DOI: 10.1016/B978-0-08-102584-0.00002-4.

[21] Merced-Grafals et al., " Repeatable, accurate, and high speed multi-level programming of memristor 1T1R arrays for power efficient analog computing applications," *Nanotechnology*, vol. 27, no. 36, Aug. 2016. DOI: 10.1088/0957-4484/27/36/365202.

[22] H. -. P. Wong et al., "Metal-Oxide RRAM", *Proc. IEEE*, vol. 100, no. 6, pp. 1951-1970, Jun. 2012.

Sarah Rafiq graduated from International Islamic University Malaysia (IIUM) with a Bachelor of Engineering in Electronics - Communications (Honours) in 2017. During her degree, she had an internship with Intel Programmable Solutions Group in 2016, where she worked on the 20nm technology product Arria10. She is currently pursuing her Ph.D. degree in Nanoscale Engineering at the College of Nanoscale Science and Engineering, SUNY Polytechnic Institute. Her research focuses on electrical characterization of Resistive Random Access Memory (ReRAM) devices, and their potential application for in-memory computations on integrated CMOS-ReRAM arrays.

Jubin Hazra graduated from University of Southern California, Los Angeles with a master's degree and Engineer's degree in Electrical Engineering in 2014 and 2016 respectively where his research was mostly based on design, fabrication and characterization of III-V compound semiconductor electronic and optoelectronic devices on Si substrates. He also has one year working experience in ASML US Inc in AZ, US as a senior optics engineer prior to joining SUNY Poly as a full time PhD student in Jan 2018. He joined Cady lab in Aug 2018 and currently his research is focused on hardware implementation of neuromorphic chips using memristor-CMOS hybrid technology and improving the ferroelectric performance in HZO films. He has been an active IEEE student member since 2018 and authored several papers in reputed journals and conference proceedings.



Maximilian Liehr obtained his BA and MEng from Rensselaer Polytechnic Institute in Troy, NY. He is currently a Ph.D. graduate student of Nanoscale Engineering in the Colleges of Nanoscale Science & Engineering at SUNY Polytechnic Institute. Maximilian Liehr has active research interests in development of Non-volatile nanoelectronics including Resistive Random Access Memory (ReRAM) devices and neuromorphic computing.

Karsten Beckmann received his B.Sc. and M.Sc. degree in Electrical Engineering and Information Technology form the Technical University of Darmstadt, Germany, in 2011 and 2013, respectively, and his Ph.D. in Nanoscale Engineering from the Colleges of Nanoscale Science and Engineering (CNSE), University at Albany (now SUNY Polytechnic Institute). He is currently a Senior Process Integration Engineer at NY CREATES / SUNY Poly working on emerging memory solution with a focus on resistive devices and selector technology. His strong background spans interdisciplinary fields combining Material Science, Nanofabrication and Electrical Engineering with a focus in electrical testing (DC, RF and automated testing), module integration into a 65 nm technology and materials development (ReRAM, Selector, Electrodes).

Minhaz Abedin graduated from Chittagong University of Engineering and Technology, Bangladesh, with Bachelor's degree in Electrical and Electronics Engineering. His undergraduate research focused on designing and characterizing CMOS Image Sensors. He received his Master's degree in Nanoelectronics, research focusing on fabrication and integration of microfluidics and silicon photonic sensors for Lab On a Chip, the project recently extended for COVID-19 antibody test. Currently he is pursuing his PhD degree with research focus on nanoscale devices for neuromorphic applications. Apart from research, he enjoys volunteering to outreach and humanitarian activities and photography as a pastime activity.

Jodh S. Pannu obtained his B.Tech degree in Electronics and Communication Engineering from Punjab University in the year 2015. He joined the Masters and later the computer science PhD program of University of Central Florida in the year 2016 and 2018 respectively. Later transferred to University of Texas at San Antonio in 2020. His research interests include post moore's law technologies such as in-memory computation and quantum computing.

Sumit K. Jha is Professor of Computer Science at the University of Texas San Antonio (UTSA). Dr. Jha received his Ph.D. in Computer Science from Carnegie Mellon University. Before joining Carnegie Mellon, he graduated with B.Tech (Honors) in Computer Science and Engineering from the Indian Institute of Technology Kharagpur. Dr. Jha has worked on R&D problems at Microsoft Research India, General Motors, INRIA France and the Air Force Research Lab Information Directorate. His research has been supported by the National Science Foundation (NSF), DARPA, the Office of Naval Research (ONR), the Air Force Office of Scientific Research (AFOSR), the Oak Ridge National Laboratory (ORNL), the Royal Bank of Canada, the Florida Center for Cybersecurity, and the Air Force Research Laboratory (AFRL). He is a full member of the Sigma Xi and is a recipient of the IEEE Orlando Engineering Educator Excellence Award. Dr. Jha was awarded the prestigious Air Force Young Investigator Award and his research has led to four Best Paper awards.

Prof. Nathaniel Cady obtained his BA and Ph.D. from Cornell University in Ithaca, NY. He is currently an Empire Innovation Professor of Nanobioscience in the Colleges of Nanoscale Science & Engineering at SUNY Polytechnic Institute. Prof. Cady has active research interests in the development of novel biosensor technologies and biology-inspired nanoelectronics, including novel hardware for neuromorphic computing.