Analog Self-Timed Programming Circuits for Aging Memristors

Aidana Irmanova, *Associate Member, IEEE*, Akshay Maan, *Member, IEEE*, Alex James, *Senior Member, IEEE*, and Leon Chua, *Fellow, IEEE*

***Abstract*—Reliable programming crossbar memristors to the required resistive states is the challenge that hinders VLSI deployment of the memristive neural network circuits, as current memristive devices face the variability issues of resistive switching. There is also a need for on-chip control circuitry that detects malfunctioning memristive nodes in the crossbar due to the memristor aging. *Program and Verify* (*P&V*) schemes can be used for both controlling resistive switching as well as evaluating the functionality of memristors. In this brief, we propose a novel analog circuit design for the *P&V* approach of row-by-row programming bipolar memristors in a 1T1M crossbar configuration. The proposed control circuit (CC) is self-timed and performs both read and program operations, decreasing the overall programming complexity. CC design is verified with Spice simulations using low power 22nm high-k CMOS models and Modified S memristor model for large scale simulations. Parasitic of wire lines under thermal variation and CMOS variability were included for programming 1T1M crossbar partitions of the sizes 16 × 16, 32 × 32, 64 × 64, 128 × 128.**

***Index Terms*—Memristor, 1T1M crossbar, aging, programming, reliability.**

I. Introduction

MEMRISTIVE crossbar circuits are taking over the niche of the proposed on-chip neural network architectures as it enables analog in memory dot-product multiplication that saves computing time, power, and die area [1]. Memristive crossbar circuits become even more attractive with the possibility of on-chip training that paves the way for hardware implementations of continuous learning [2]. However, the mass production of memristive edge devices for wider use is hindered by the reliability issues of memristor functionality such as stochastic resistive switching, endurance, aging and the device to device variability across the process [3]. Fig. 1(b, c) shows the example of memristor aging, where with time, high and low levels of the resistive states deviate from the original resistance distribution (Measurement samples are in Supplementary Information (SI)). Although the lifetime of the devices depends on the inherent properties of the material [3], there are measures to prevent the damage in devices such as avoiding overwriting the device (writing the same value) and the controlling the amount of writing stimuli. Reusing aging memristors is also possible after spotting the device degradation with remapping techniques [5], [6]. With the given layout, for efficient implementation of online training memristive crossbars, there is the need for the control circuit (CC) that ensures the resistive switching and spots the aged or malfunctioning memristors.

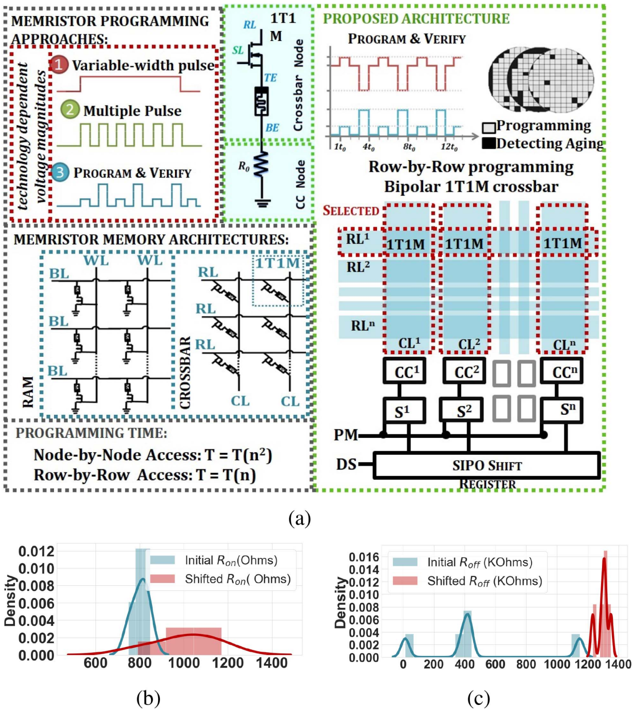


Fig. 1. System diagram of the analog trainable 1T1M crossbar system for programming and detecting the malfunctioning memristors. (a). Experimental measurement of Carbon (C) doped SDC memristor [8]. Shifted R_{on}(b) and R_{off}(c) values due to the aging after several iterations of programming (Supplementary material).

In this brief, we propose the novel design of the bipolar memristor programming control circuit (CC) for 1T1M crossbar architecture that uses *Program & verify* scheme for ensuring the

Manuscript received September 6, 2020; accepted October 11, 2020. Date of publication October 20, 2020; date of current version March 26, 2021. This brief was recommended by Associate Editor L. A. Camunas-Mesa. *(Corresponding author: Aidana Irmanova.)*

Aidana Irmanova is with the School of Engineering and Digital Sciences, Nazarbayev University, Nur-Sultan 010000, Kazakhstan (e-mail: aidana.irmanova@nu.edu.kz).

Akshay Maan is with the Bio Medical Technology Wing, Indriyam Biologics Pvt. Ltd., Trivandrum 695012, India (e-mail: akshay.maan@gmail.com).

Alex James is with the School of Electronics, Indian Institute of Information Technology and Management, Trivandrum 695581, India (e-mail: apj@ieee.org).

Leon Chua is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: chua@eecs.berkeley.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSII.2020.3032282.

Digital Object Identifier 10.1109/TCSII.2020.3032282

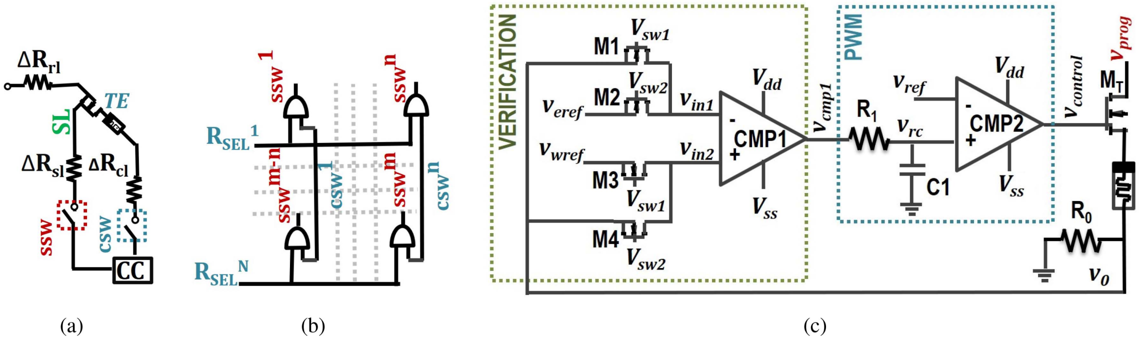


Fig. 2. (a) 1T1M node connection to the CC. (b) Address decoder circuit: Activates SL Latch (*SSW*)) of the target memristor ANDing the control signals of Row selection (R_{sel}) and CC activating V_{dd} signal (*CSW*). Connects the ($V_{control}$) node of the CC to the gate of the target node (c) Proposed CC Design: CC is connected to the target memristor. V_0 represents the node connected to the bottom electrode of the memristor via *CSW*.

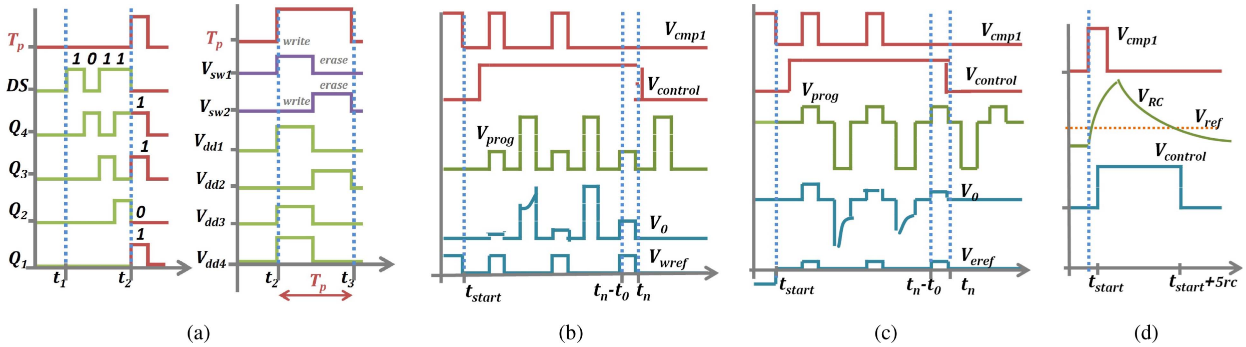


Fig. 3. Crossbar Row selection for parallel programming memristors:(a) 1T1M memristor node connection to the CC depending on the programming mode (PM). DS - data series, Q - is the SIPO SR outputs. (b) Time diagram of the CMP_1 outputs and voltage levels at the nodes $V_{control}$, V_{prog}, V_0, V_{eref} and V_{ref} during WRITE (b) and ERASE (c) operations. (d) PWM process.

memristor programming and detecting the aged devices. With the *P&V* approach: the resistance of the memristor is measured after application of the certain programming stimuli (Fig. 3 (b,c)) until the desired state is reached. The circuit spots the aged memristor if the device is not switching to the reference state in the given time limit. CC is self-timed, meaning it stops the programming stimuli at the target 1T1M node, yet allowing the parallel update of memristors row-by-row.

The key contributions of this brief are:

1) Bipolar Memristor programming in the 1T1M crossbar with the proposed CC using 22nm-node high-k PTM models under CMOS variability.
2) System design of the analog trainable 1T1M crossbar block including analog CC for programming and detecting malfunctioning memristors, as well as the digital components for routing the programming output data (aging detection decisions).
3) Comparative analysis for programming precision loss across the row line, power consumption and leakage current for different crossbar partitions sizes: 16 × 16, 32 × 32, 64 × 64, 128 × 128, including parasitic of wire lines, CMOS variability under thermal variation.

In the following section proposed architecture is described and the details of the CC design and operation are provided. Next, the impact of the CMOS variability, thermal variations, and parasitics on the performance of the CC is analyzed. Section III discusses the related works and the aspects of competing approaches.

II. Proposed Architecture

In this brief, we propose a novel analog circuit design for the *P&V* approach of row-by-row programming aging bipolar memristors in 1T1M crossbar configuration (Fig. 1(a)). The key aspects of the CC are:

1) For row-by-row programming bipolar memristors V_{prog}(Fig. 3(b, c)) is applied to the entire row, while column lines are used for sensing output signals.
2) For reading the resistive state memristors require stabilization time after stimuli application. Therefore programming and reading are follow-up operations. Figure 1(a) and 3(b) shows the *P&V* signal V_{prog}, where reading operation (resistance verification) follows the programming operations (WRITE and ERASE).
3) Resistance verification is implemented comparing the voltage drop across the target memristor V_0 to the V_{eref} or V_{wref} references in the CC. V_{eref} or V_{wref} are reference voltages for R_{off} and R_{on} respectively, and precalculated using resistive divider equation. Values of V_{eref} and V_{wref} depend on the V_{read} of the V_{prog} programming stimuli applied to the row line and the resistance R_0 in the CC which is connected to the target memristor during programming.
4) CCs access the crossbar memristors row-by-row, therefore programming time is ($T(n)$). The select transistor of memristors controlled by the $V_{control}$ generated by CC disconnects the memristor from the row line in the given time limit T_0.

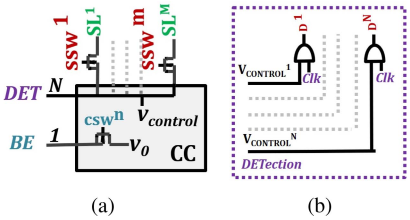
5) If in the given time limit T_0 there are select nodes remain connected to the CC, active $V_{control}$ nodes are memorized as malfunctioning or the programming T_0 period repeats with stronger stimuli V_{write} or V_{erase} depending on the algorithm.

A. Control Circuit Operation

The 1T1M node in the crossbar Fig. 2(a, c), consists of one selecting transistor in series with a memristor (Fig. 1 (a)), where the selector has the role of enabling or disabling a memristor node from the crossbar. Each CC is designed to switch ON or OFF only one selecting transistor in a row, thus it has an access to one memristor in a row line (RL). When the crossbar at the programming mode, each column of the crossbar connects its CL node to the corresponding Control Circuit (CC), Fig. 1(a), via *CSW* latch (Fig. 2(a)). While the selected row *RL* is connected to the driving source V_{prog} generator, row selector R_{sel} signal row activates the address decoder as in Fig. 2(b) that connects the gate lines of the selecting transistors of 1T1M to the $V_{control}$ node of the CC. At the same time the Control Circuit (CC) is powered, and drives the selecting transistor to V_{dd}. Activated selecting transistor connects the row line and *TE* nodes of the target memristor.

B. Control Circuit Design

The control circuit (CC) consists of three blocks: verification, pulse width modulation (PWM), and a potential divider as shown in Fig. 2(b). At the programming mode, the verification block compares the voltage drop V_0 to a reference voltage V_{wref}, (or V_{eref}) to decide whether writing to (erasing) the memristor has to be continued. For calculation reference levels (V_{wref} or V_{eref}) equation of the potential divider of the R_0 and target memristor are used. To avoid the error series latch and selector transistor resistance values should be taken into account in the equation. The verification block consist of four transistor switches *M1, M2, M3, M4* (NMOS devices) and a comparator CMP_1 (designed as in [9]) shown in Fig. 2(c). The switches are required for selecting and setting the specific reference signal, V_{eref} or V_{wref}, based on the mode of operation. *M2* and *M4* are controlled with the WRITE mode signal $Vsw1$ while *M1* and *M3* are controlled with the ERASE mode signal $Vsw2$ (inverted $Vsw1$). In this brief, we use pulsed V_{wref} and V_{eref} in order to compare the memristance only during READ period of the V_{prog}: when V_{prog} is at the read period V_{wref}, (or V_{eref}) pulse is ON, when V_{prog} at the write or OFF state V_{wref}, (or V_{eref}) pulse is OFF. Thus the output of the V_{cmp1}, outputs V_{dd} pulse with the width of t_0 when $V_{eref} < V_0$ or $V_{wref} > V_0$. If CMP_1 outputs a positive pulse, CC will continue the programming operation until the next read. Next block PWM is used to convert the comparator output pulse V_{cmp1} with the duration $1t_0$ (ON period) to V_{select} with the duration (ON) $5t_0$. V_{select} drives the gate line of the selecting transistor 1T1M. The ON duration of the $5t_0$ is determined by the time that required to keep driving the gate line of *MT* at V_{dd} for the one more following programming and reading periods of the V_{prog}: $1t_0$ for continuing reading, $2t_0$ for V_{write1} and for V_{read1}. PWM block consists of the RC low pass filter and the amplifying comparator CMP_2 [9]. The capacitance of the capacitor *C1* defines the operation frequency of the CC. R1 and C1 values were set, such as the 5RC duration that discharges C1 would be as long as $5t_0$: The RC filter transforms the V_{cmp1} square pulse to triangular signal and the comparator CMP_2 restores it to the $V_{control}$ ($5t_0$) square pulse, which is shown in Fig. 3(d). Thus, $V_{control}$ keeps the selecting transistor switched on till the following READ pulse. The reference voltage at the negative pin of the CMP_2 regulates the duration of the $V_{control}$ pulse. When programming time for the each row begins, first t_{start} period is required to start the operation of the CC. During t_{start} the CMP_1 inputs are always at the condition $V_{in1} > V_{in2}$. It is handled with V_{eref} and V_{wref} input (Fig. 3(b, c)). $V_{in1} > V_{in2}$ forces CMP_1 to output positive V_{cmp1} pulse that drives the $V_{control}$ till the first READ period of the V_{prog}, which is depicted in the Fig. 3(a, b). It has to be noted that the offset of the V_{eref} and V_{wref} are not necessarily at 0V, and may be varied to observe $V_{in1} > V_{in2}$ at OFF period (Supplementary information).



(a) (b)

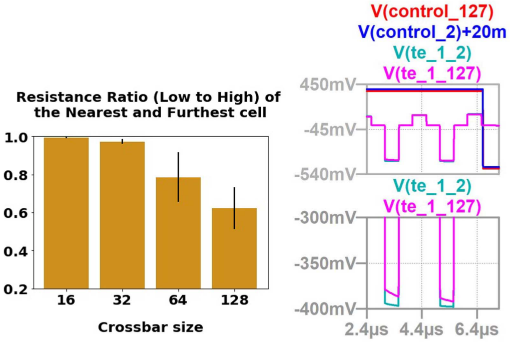
Fig. 4. Interface of the CC at the $V_{control}$ node (a) Aging Detection circuit (b).

C. Aging Detection

There are three cases of programming scenario that can take place during proposed circuit operation: (1) The initial state of the memristor is already at the desired R_{on} or R_{off} state. Hence the programming pulses stop at target *TE* after the first read pulse. (2) The initial state of the memristor is not at the desired state. After one or more programming pulses, the memristor is successfully programmed to the required state; and (3) The initial state of the memristor is not at the desired state and even after T_0 of several programming cycles, the memristor is stuck at the initial or intermediate state. The first two cases are part of the normal operation of the programming circuit where the CC is able to achieve desired memristor states in finite time, while the third case leads to the programming failure and keeps the $V^n_{control}$ of the CC^n at V_{dd}. Aging detection circuit on the other hand spots the high V_{dd} of the $V^n_{control}$ and routes the signal to the decision making block which could store the address for further remapping schemes or repeat the programming process of the RL_n. Aging detection circuit is proposed as the *n* number of AND gates connected to the $V^n_{control}$ of the each CC^n as shown in Fig. 4(a) and (b). *Clk* is a periodic signal synchronized with the T_0 programming time.

III. Circuit Performance

Variability Simulation: For the simulation of presented CC 22nm high-k PTM model [10] and the Modified S Memristor model were used. Memristor model was set with following

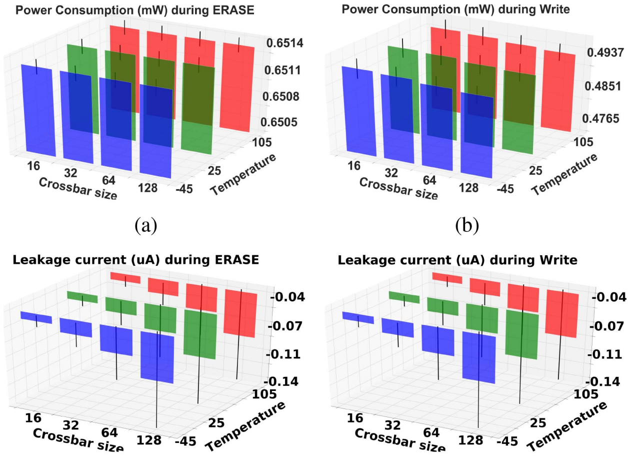


(b)

Fig. 5. Impact of the crossbar scaling on programming variability across the row line.

parameters: $V_{th} = 0.27$, a=10, $R_{ON} = 3\text{K}\Omega$, $R_{OFF} = 600\text{K}\Omega$, k= $-5e10$ [11]. Simulations include resistive and capacitive parasitics of the crossbar lines: ΔR = 1mΩ, was set for each node as shown in Fig. 2(a). The *SSW* and *CSW* NMOS-PMOS bilateral switches included to the simulation adding additional resistance. Line capacitance was set as in [12], for each column $\Delta C = 1pF$. CMOS transistors sizes was randomly varied up to 2%, and temperature condition was swept from -45 up to 105. Average power consumption for each CC and leakage current of each column was calculated at -45, 25, 105 C degrees for the crossbar partition sizes 16 × 16, 32 × 32, 64 × 64 and 128 × 128 (Figure 6).

Speed, Temperature, Power and Accuracy: The switching dynamic of memristor programming depends on the intrinsic properties of the materials such as switching threshold, resistance levels, switching speed. These parameters predetermine the operating frequency and reference voltage magnitudes in the circuit. Therefore following circuit parameters as well as power consumption and programming precision may differ for each memristor technology. In this brief, pulses with the duration of $t_0 = 0.5\mu s$ and voltage amplitudes $V_{erase} = [-0.4V]$, $V_{write} = [0.8V]$ and $V_{read} = 0.2V$ were used as programming patterns shown in Fig. 3(b, c). The parameters and control signals such as $R_0 = 500\Omega$, $R_1 = 5\text{k}\Omega$; $C_1 = [0.1\text{nF}]$ and $V_{ref},= [26mV]$ for $t_0 = 0.5\mu s$ and, $V_{dd} = 0.8V$, $V_{ss} = -0.8V$. The initial resistance states R_{ini} were randomly set in the range of $[R_{on}, R_{off}]$. V_{eref} and V_{wref} were set as 5mV and 15mV respectively, with resistive divider equation ($V_{in} = V_{te} \approx V_{read}, R_1 = \text{M}, R_2 = R_0$). The offsets of V_{eref} and V_{wref} should satisfy the initial condition $V_{in2} > V_{in1}$. To increase the comparison precision of CMP_1 and CMP_2, we incorporated opamp at the node V_0 to amplify the V_0 signal. The lower the conductance to be programmed the higher amplification will be required. In this brief we use the same design for CMP_1 and CMP_2 as well as for amplifier at node V_0 [9], $V_{dd'} = 0.5V$, $V_{ss'} = -0.5V$. Different levels of power supply is provided to decrease the power consumption. The Fig. 6 shows linear rise of the leakage current during ERASE with an increasing temperature and the size of the crossbar partition. Power consumption repeats the trend increasing up to $0.65mW$. For WRITE operation the mean power consumption varies in the range of $0.5mW$, with the higher leakage current.



(a) (b)

(c) (d)

Fig. 6. Mean and standard deviation of the power dissipation during write and erase operations (a). Average leakage current of the column lines for different sizes of the crossbar partition (b).

As the scaling the crossbar size increases the loading effect and the length of the line introduce the varied parasitic, the error of the nearest and furthest cell was calculated presetting the same initial state and parameters to the memristor (Fig. 5). Overall, it can be noted that the scaling the crossbar size with given parameters irrespective of the temperature variation introduce the variation to the stimuli that each memristor sees at the row. Figure 5(a) shows the ratio of the nearest and furthest cell being programmed to the same states in the row at the room temperature with +2% variation to the CMOS devices. Scaling the partition starting from 64×64 drops the ratio to on average to 0.8 while at the size 128×128 it drops down to 0.4. It is related to the voltage drop of V_{prog} across the row line which reduces the programming stimuli magnitude, due to the voltage drop across the line length and the loading effect of the CC. Fig. 5(b) shows the V_{prog} drop across the row line affecting the programming magnitude at the nearest ($V\_te\_1\_2$) and furthest cell ($V\_te\_1\_127$) nodes. For the CCs that are located further from the source generator, this issue can be addressed by altering V_{prog}, V_{wref} and V_{eref} proportionally to the voltage drop.

IV. Discussion and the Related Works

We put forward two aspects of the designed circuit that addresses open problems of the memristor programming field. The first is related to the incorporating the programming into memristive crossbar architecture in a single chip. Analog programming circuit that uses voltage domain to communicate analog values as in proposed CC (V_0, V_{wref}, V_{eref}) enables the tuning the state reference (V_{wref}, V_{eref}) depending on the distribution of R_{on}, R_{off} values of the memristors, which is essential during remapping the aged memristors or on-chip analog learning. The second aspect is related to the up to date problems of memristor technology as stochastic switching and aging. Provided that in 1T1M crossbars there will be memristors that can switch within the duration of the t_0 pulse or N number of t_0, the proposed CC addresses the device to device variability issue by repeatedly applying programming pulses with predetermined amplitude. As regards the aging, the application only required number of programming stimuli to the devices prevents over-stressing memristors, while already malfunctioning devices are detected with CC. Although the $P\&V$ is a common method for programming traditional memory architectures in [13]–[15] including memristive ReRAM that are not designated for dot product calculation shown in Fig. 1(a), there are programming circuits implementation for analog crossbars that performs the dot-product operation. The methods of programming and sensing differ and can be simultaneous as well as follow up $P\&V$. Earlier implementation of simultaneous programming and sensing circuits include [16]–[19] where the reference current is measured during application of the single programming pulse with different width or during multiple short pulse application shown in Fig 1(a): approach (1) and approach (2). However, the application of prolonged DC signal (1) approach may be damaging for the devices, and the lower signal magnitudes as in $P\&V$ (3) are required for ensuring the resistive switching in aging devices or with stochastic switching. In crossbar configuration using $P\&V$ to verify or read the memristance at each node, memristor should be accessed node-by-node or row-by-row. With the node-by-node access, programming time increases up to $T(n^2)$, that is why row-by-row programming is preferred in the proposed architecture Fig. 1(a). In order to decrease the node-by-node T of $P\&V$ schemes, programming algorithms were proposed for efficient tuning memristance in [20], [21], generating the V_{read}, V_{write} and V_{erase} signals with SCS instruments as [22]. Although the source generator circuits is not considered in this brief, proposed CC can be used in combination with the algorithms in [20], [21]. In [21] memristive cells are fine programmed for only R_{off} or R_{on} states, thus decreasing the programming cycles. While [20] proposes the algorithms that precalculate the required initial magnitude and duration of the applied programming pulses.

V. Conclusion and Future Work

The programming circuit enables the on-chip incorporation with analog dot product computing memristive crossbar arrays that paves the way for on-line tuning and continuous learning neural network architectures. The presented work proposes reliable row by row programming and aging detection circuit for memristor arrays. The circuit is designed to be tuned for the different reference voltages applied to the control circuit that remaps the different R_{on} and R_{off} levels, thus enabling re-configurable programming which is required for reusing the early aged memristor devices. Each memristor is programmed with the required amount of stimuli that might vary due to the device aging or inherent stochasticity.

References

[1] P. Lin *et al.*, “Three-dimensional memristor circuits as complex neural networks,” *Nat. Electron.*, vol. 3, no. 4, pp. 225–232, 2020.

[2] A. Samajdar, P. Mannan, K. Garg, and T. Krishna, “GeneSys: Enabling continuous learning through neural network evolution in hardware,” in *Proc. 51st Annu. IEEE/ACM Int. Symp. Microarchit. (MICRO)*, 2018, pp. 855–866.

[3] M. Zhao, B. Gao, J. Tang, H. Qian, and H. Wu, “Reliability of analog resistive switching memory for neuromorphic computing,” *Appl. Phys. Rev.*, vol. 7, no. 1, 2020, Art. no. 011301.

[4] T. Dongale *et al.*, “Investigation of process parameter variation in the memristor based resistive random access memory (RRAM): Effect of device size variations,” *Mater. Sci. Semicond. Process.*, vol. 35, pp. 174–180, Jul. 2015.

[5] Y. Xiang *et al.*, “Impacts of state instability and retention failure of filamentary analog RRAM on the performance of deep neural network,” *IEEE Trans. Electron Devices*, vol. 66, no. 11, pp. 4517–4522, Nov. 2019.

[6] S. Zhang, G. L. Zhang, B. Li, H. H. Li, and U. Schlichtmann, “Lifetime enhancement for RRAM-based computing-in-memory engine considering aging and thermal effects,” in *Proc. 2nd IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, 2020, pp. 11–15.

[7] S. N. Mozaffari, K. P. Gnawali, and S. Tragoudas, “An aging resilient neural network architecture,” in *Proc. 14th IEEE/ACM Int. Symp. Nanoscale Archit.*, 2018, pp. 25–30.

[8] K. A. Campbell, “Carbon-chalcogenide variable resistance memory device,” U.S. Patent 9 118 006, Aug. 25, 2015.

[9] V. Saxena and R. J. Baker, “Indirect compensation techniques for three-stage fully-differential op-amps,” in *Proc. 53rd IEEE Int. Midwest Symp. Circuits Syst.*, 2010, pp. 588–591.

[10] (Oct. 2007). *22nm PTM Model for Metal Gate High-k CMOS: V2.0.* [Online]. Available: ptm.asu.edu

[11] D. Biolek, Z. Kolka, V. Biolková, Z. Biolek, M. Potrebić, and D. Tošić, “Modeling and simulation of large memristive networks,” *Int. J. Circuit Theory Appl.*, vol. 46, no. 1, pp. 50–65, 2018.

[12] N. Wald and S. Kvatinsky, “Understanding the influence of device, circuit and environmental variations on real processing in memristive memory using memristor aided logic,” *Microelectron. J.*, vol. 86, pp. 22–33, Apr. 2019.

[13] L. Wei *et al.*, “13.3 A 7Mb STT-MRAM in 22FFL FinFet technology with 4ns read sensing time at 0.9V using write-verify-write scheme and offset-cancellation sensing technique,” in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2019, pp. 214–216.

[14] P. Jain *et al.*, “13.2 A 3.6Mb 10.1Mb/mm^2 embedded non-volatile ReRAM macro in 22nm finfet technology with adaptive forming/set/reset schemes yielding down to 0.5 V with sensing time of 5ns at 0.7V,” in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2019, pp. 212–214.

[15] N. Papandreou *et al.*, “Programming algorithms for multilevel phase-change memory,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2011, pp. 329–332.

[16] R. Berdan, T. Prodromakis, and C. Toumazou, “High precision analogue memristor state tuning,” *Electron. Lett.*, vol. 48, no. 18, pp. 1105–1107, 2012.

[17] W. Yi *et al.*, “Feedback write scheme for memristive switching devices,” *Appl. Phys. A, Solids Surf.*, vol. 102, no. 4, pp. 973–982, 2011.

[18] Y. Zhao, B. Li, and G. Shi, “A current-feedback method for programming memristor array in bidirectional associative memory,” in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, 2017, pp. 747–751.

[19] M. Ramadan, N. Wainstein, R. Ginosar, and S. Kvatinsky, “Adaptive programming in multi-level cell ReRAM,” *Microelectron. J.*, vol. 90, pp. 169–180, Aug. 2019.

[20] E. J. Merced-Grafals, N. Dávila, N. Ge, R. S. Williams, and J. P. Strachan, “Repeatable, accurate, and high speed multi-level programming of memristor 1T1R arrays for power efficient analog computing applications,” *Nanotechnology*, vol. 27, no. 36, 2016, Art. no. 365202.

[21] K. V. Pham, S. B. Tran, T. V. Nguyen, and K.-S. Min, “Asymmetrical training scheme of binary-memristor-crossbar-based neural networks for energy-efficient edge-computing nanoscale systems,” *Micromachines*, vol. 10, no. 2, p. 141, 2019.

[22] *Keithley 4200-SCS Semiconductor Characterization System.* Accessed: Aug. 29, 2020. [Online]. Available: https://www.testequity.com/products/4306/