Methods for High Resolution Programming in Lithuim Niobate Memristors for Neuromorphic Hardware

Chris Yakopcic, Shu Wang, Weisong Wang, Eunsung Shin, Guru Subramanyam and Tarek M. Taha
Department of Electrical and Computer Engineering
University of Dayton, Dayton, OH
cyakopcic1@udayton.edu

***Abstract*—Memristor crossbars are capable of implementing learning algorithms in a much more energy and area efficient manner compared to traditional systems. However, the programmable nature of memristor crossbars must first be explored on a smaller scale to see which memristor device structures are most suitable for applications in reconfigurable computing. In this paper, we demonstrate the programmability of memristor devices with filamentary switching based on $LiNbO_x$, a new resistive switching oxide. We have performed several characterization experiments that demonstrate the high resolution programmability of these devices. Once the optimal programming voltages and conductivity ranges for these memristor devices are determined, a method is presented to determine how many unique states can be achieved in these devices. Parameters obtained from this analysis can greatly strengthen memristor device models that implement realistic stochastic switching behavior. We show that 11 unique states can be stored in this device with 95% confidence, but many more states are achievable using feedback write techniques or in-situ learning. Therefore, we have determined that Lithium Niobate memristors are strong candidates for use in neuromorphic computing.**

Keywords—memristor; device; convolution; neural network; deep network

I. Introduction

The memristor [1] is a non-volatile two-terminal passive circuit element with a wide programmable resistance range. Memristors can be laid out in a high density grid known as a crossbar [2], which gives memristors the potential to be fabricated with an areal density greater than that of synapses in brain tissue [1]. These crossbars can be used to produce high density, extreme low-power, neuromorphic hardware [3,4] capable of performing many parallel multiply-add operations in the analogue domain. Existing papers [3-6] based on the simulation of neuromorphic memristor crossbars show promising results with these large high density crossbar structures. Furthermore, neuromorphic systems based on memristor crossbars have potential to perform at a power efficiency of 6 to 8 orders of magnitude greater than that of traditional RISC processors [4]. Recently, memristors architectures have become strong candidates for implementing hardware based on Convolutional Neural Networks (CNNs) [7-9], and Restricted Boltzmann Machines (RBMs) [10]. However, before these systems can be developed, reliable memristor device structures must be investigated.

As opposed to memristors for non-volatile memory systems [2], the ideal memristor in neuromorphic computing systems should have multiple accessible resistance levels that allow for a programmable resistance range [3, 5]. In this paper we show that filament switching within Lithium Niobate memristors [11,12] is capable of providing high resolution programmability. Several I-V characterization techniques were carried out in this work to determine how many conductivity levels can be programmed into a memristor device. We have determined it is possible to program 11 unique conductivity states with 95% confidence. However, it may be possible to program up to about 100 conductivity levels if a repetitive feedback write [7,10] mechanism or in-situ learning methods [13] are used. Lithium Niobate is a relatively new resistive switching material, and this is among the first work to show its potential for use in neuromorphic memristor circuits.

The rest of this paper is organized as follows: Section II describes the fabrication technique, layout, and physical operation of these Lithium Niobate memristor devices. Section III discusses several I-V characterization experiments that were performed to show how fine resolution programming can be achieved. Section IV presents a data analysis technique that can be used to quantify the programming resolution achieved in a memristor device, and Section V concludes the paper.

II. Lithium Niobate Memristors

The device structure (Ti / Pt / $LiNbO_x$ / $LiNbO_3$ / Ag / Au from bottom to top) proposed in this work is based on a 42 nm switching layer (thickness ratio of $LiNbO_x$ to $LiNbO_3$ = 4:1) prepared with pulsed laser deposition (PLD) using the method proposed in [14]. The oxygen deficient layer was developed by varying the oxygen level in the chamber. The bottom electrode consists of 10 nm of titanium and 50 nm of platinum and the top electrode consists of 10 nm silver and 50 nm of gold. The substrate is silicon topped with 1 μm of PECVD silicon dioxide. The resulting wafer contains arrays of individually isolated memristors that have different device overlap areas ranging from 5 to 25 μm^2.

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

It is often stated that metal-oxide resistance switching devices produce a change in conductivity due to the formation of metallic filaments made up of oxygen deficient ions [15,16]. Fig. 1 (a) shows the physical layer structure as well as the directions of filament formation within the $LiNbO_3$ / $LiNbO_x$ switching layer. As a positive voltage is applied to the top electrode, the positively charged oxygen deficient Lithium Niobate ions are forced toward the $LiNbO_x$ layer, reducing the chance of a filament formation, and thus lowering device conductivity. When a positive voltage is applied to the bottom electrode, the opposite occurs and oxygen deficient ions are forced into the $LiNbO_3$ layer, rupturing the oxide film and producing metallic paths that increase device conductivity.

In some cases, when devices are developed for fast binary switching [2], a filament formation can be thought of as a process where a device is either conducting (with a completed filament) or not conducting (with a broken filament). However, the formation of multiple filaments, or the change in thickness of a filament can produce a wider range of attainable conductivity values, as seen in this work.

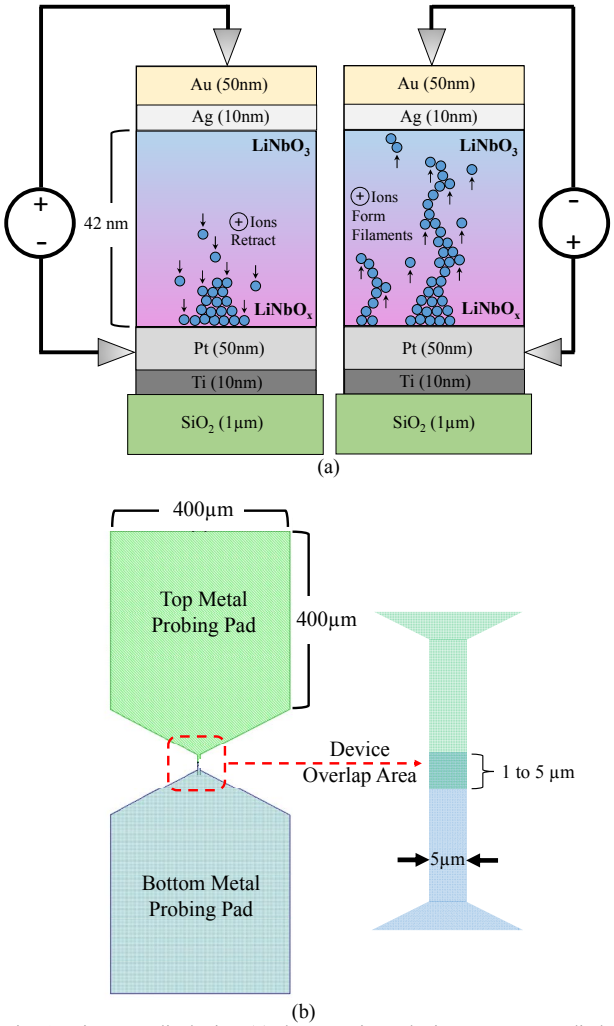

Fig. 1. Diagrams displaying (a) the memristor device structure studied with the impact of applied voltage polarity, and (b) the overhead view of how the top and bottom metals form devices in their overlap area.

Fig. 1 (b) provides the overhead view of the device structure when a memristor is defined based on the overlap area of the two metal layers. The top and bottom metal layers each provide a large probing pad so every individual memristor can be accessed. An etching step during the device fabrication process removes all oxide material that was patterned on top of the bottom metal probing pad so the bottom electrode can be easily accessed.

III. Memristor Characterization

A number of different current-voltage characterizations were completed with the objective of showing that these devices are suitable for high resolution neuromorphic programmability.

A. Memristor I-V Characteristic

A cyclic voltage sweep was performed on each suitable memristor on the wafer (about 100 devices total) to determine which devices will operate most reliably. This is a common first step in our characterization process to ensure devices have formed properly, and that they are able to switch as intended in both the positive and negative regimes. Fig. 2 displays two example I-V characteristics obtained for different devices on the wafer that have overlap areas of 7.5 and 10 μm^2. These I-V characterizations show many data points present in the switching regions between the points of minimum and maximum conductivity. This implies that fine resolution programming is possible using these devices. Different memristors that are more suited for binary memory applications generally have fewer data points in this region [2], demonstrating the possibility of high speed switching. However, that is not desired when using these devices for analog weight storage in a neuromorphic circuit.

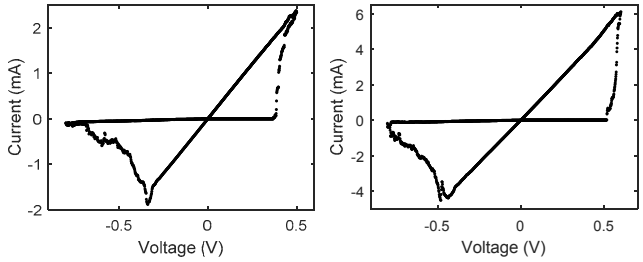

Fig. 2. Cyclic I-V characterizations of the $LiNbO_x$ devices with total overlap area of (a) 7.5 μm^2 and (b) 10 μm^2.

B. Repetitive Sweep Characterizations

This test is often performed [17,18] to show that several stable conductivity states are realizable within a memristor device. This test is similar to the cyclic voltage sweep in the previous subsection, but no negative voltages are applied between successive positive sweeps. Also, the maximum positive voltage applied should be smaller than the maximum voltage in a fully cyclic voltage sweep (see Fig. 2). This way, the conductivity of the device will only incrementally increase with each sweep to induce intermediate stable resistance states.

Fig. 3 shows the resulting I-V characteristic when performing several repetitive positive voltage sweeps on a

memristor device with an area of 10 μm^2. The result shows that repetitively sweeping the device induces multiple resistance states. Also, the amount change in conductivity is dependent on both the magnitude of the voltage applied and the existing conductivity state relative to the maximum conductivity boundary. This is why the amount of resistance change is decreasing as larger amplitude voltage sweeps are applied. This relationship between voltage, conductivity, and conductivity boundary is something that we have observed is several other devices when developing memristor modeling techniques [19-23].

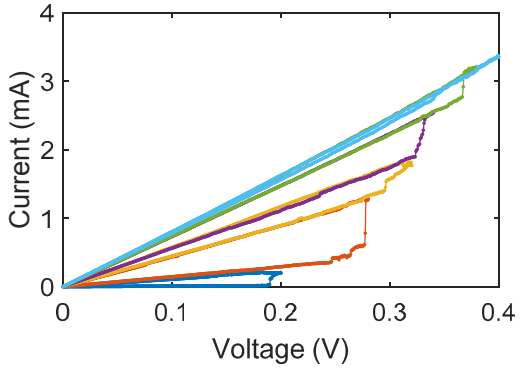

Fig. 3. Resulting I-V characteristic when multiple positive voltage sweeps are applied to a 10 μm^2 memristor device. The characteristic of each positive sweep is plotted in a different color so each can be differentiated more clearly.

Fig. 4 displays the experiment based on repetitive positive sweeping differently to show the higher resolution programmability. The conductivity of the device increases by varying amounts after each voltage sweep. The first few sweeps do not produce a significant conductivity change, but a steady change begins to occur at about 0.3 V.

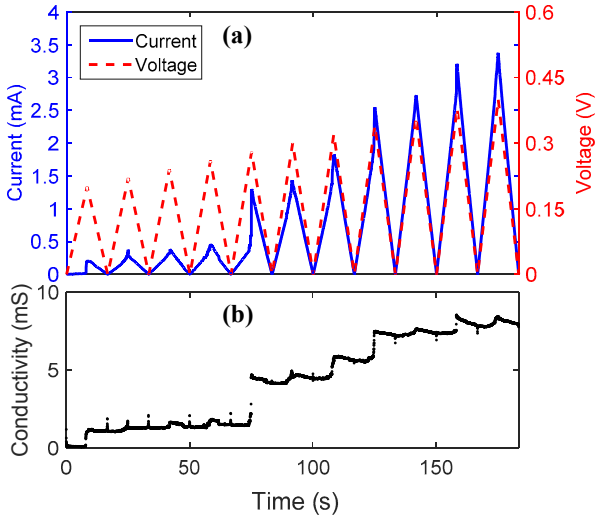

Fig. 4. Memristor characterization using repetitive positive voltage sweeps to induce intermediate conductivity states within a 10 μm^2 memristor device. Plots display (a) the voltage and current waveforms and (b) the dynamic device conductivity as a function of time.

Note that the plot in Fig. 4 (b) shows a slight reduction in conductivity as the voltage magnitude is reduced during each voltage sweep. Memristors not only change conductivity based on physical internal state, they also have a voltage dependent resistance component similar to a diode [18]. This will not disrupt the read process within a crossbar as long as all reads are performed at approximately the same voltage.

C. Pulse Characterizations

The Keithley 2400 Source Meter used to generate these results is capable of being programmed to deliver a pulse measurement at the rate of one Power Line Cycle (PLC). Therefore, we are able to study the impact of repetitively pulsing these memristor devices with a varying voltage magnitude and a pulse width of about 17 ms. We chose to use this test equipment because we want to show that these devices have a relatively high resolution when large pulse widths are applied. Therefore, we should be able to get even finer programmability in the microsecond scale. Many other memristor devices can switch over their full resistance range in less than 10 ns, leading to the thought that high-cost sub-nanosecond pulses would be required to achieve high resolution programmability [5].

The result in Fig. 5 shows the conductivity programming resolution we were able to achieve. As opposed to using repeating identical pulses [12], using a variable programming voltage in this characterization allows us to quickly determine which voltage magnitude is best for programmability. Also, we are able to easily observe the impact that voltage magnitude has on programming resolution. Fig. 5 (b) shows that applying a voltage of 0.33 V induces a very significant increase in conductivity, and this is likely a voltage too large for the high resolution programming we wish to achieve in this device. The range of programming voltage magnitudes that appear to be most effective are between 0.26 to 0.32 V.

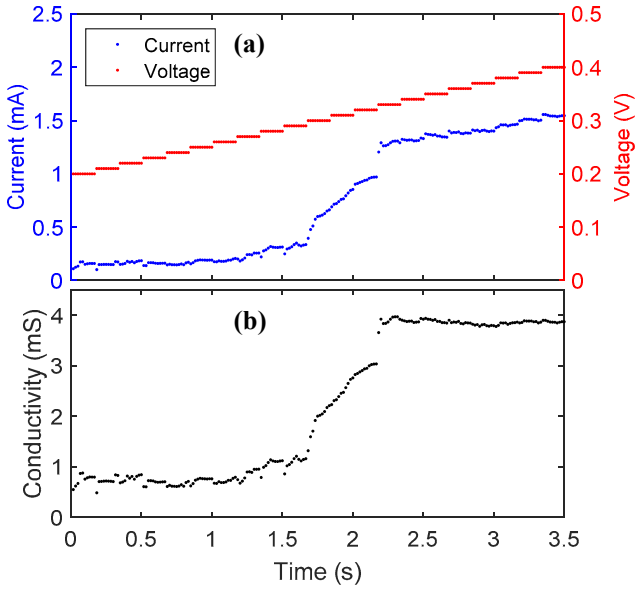

Fig. 5. Memristor characterization using repetitive pulses to induce intermediate conductivity states in a 10 μm^2 device. Plots display (a) the voltage and current waveforms and (b) the dynamics device conductivity as a function of time.

IV. Programming Resolution

Fig. 5 shows that the most desirable linear range for high resolution memristor programming occurred between 0.26 and 0.32 V, so that data was extracted for the study in this section. Within this region, there is a very smooth transition in conductivity from about 1 to 3 mS. The change in conductivity ΔG was recorded for each of these points to determine how much conductivity change a voltage pulse is likely to produce.

Fig. 6 shows the resulting distribution of change in conductivity (ΔG). On average, each pulse caused the memristor conductivity to change by 35.8 µS. Based on the data in Fig. 5, the device conductivity changes from about 0.5 to 4 mS for a total change of 3.5 mS. Dividing this range by the mean of ΔG determined in Fig. 6 results in about 97 programmable states. However, there is a significant variance in this data where the standard deviation of conductivity change is calculated to be 78.1 µS. Therefore, it may not be fair to assume that this device has a resolution of 97 unique states. To fully account for the variance in the data, the actual achievable resolution should be calculated as 35.8 µS ± 2×78.1 µS if a 95% confidence is required. After considering this large variance, the actual number of unique states is determined to be about 11.

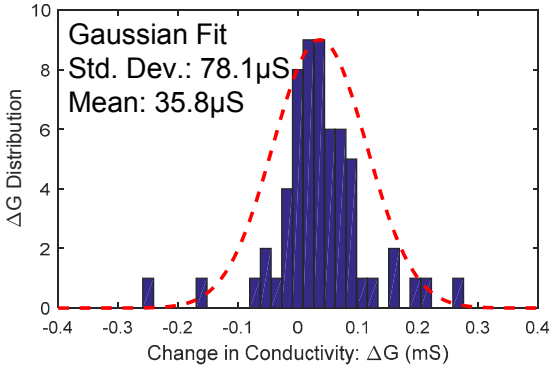

Fig. 6. Data distribution (along with Gaussian fit) for changes in conductivity due to voltages pulses with magnitude between 0.26 and 0.32 V.

It seems most reasonable that memristor programming resolution should be defined in part by application. Some memristor programming systems may not need a 95% confidence during programming to operate correctly. Some systems are designed to iteratively program a memristor within a crossbar until the correct conductivity is determined [10]. These systems will know if there is an overshoot during programing and will correct for it, eventually landing closer to the correct conductivity. Furthermore, neuromorphic hardware that implements in-situ learning [13] self corrects for overshoots during the training phase. In-situ learning is desirable in memristor hardware because of its ability to correct for errors within memristor grids.

However, it may be more logical for an ex-situ learning system with a less flexible programming mechanism to be designed based assumption that fewer unique states are achievable. The presented memristor should work well in these types of systems, because prior work shows that memristors capable of storing 8 to 16 unique values are strong enough to implement several different learning algorithms including multilayer perceptrons [10], RBMs [10], and CNNs [7].

As a final thought, it is important to note that the information obtained in this analysis can be used to improve the accuracy of modeling switching noise in memristor devices. In device models that implement stochastic switching behavior [22], the information obtained in Fig. 6 is useful in determining variance parameters for realistic noise variables.

V. Conclusion

This paper presents a memristor device capable of fine resolution programming, and a method for determining programmable range and noise magnitude during switching within memristors. The ability to achieve greater than 3-bit resolution at 95% confidence when being programmed with millisecond-scale pulses is of significant benefit to neural system designers. Therefore, the Lithium Niobate devices presented in this paper are strong candidates for use in neuromorphic architectures.

As future work, there are several different ways we plan on testing the limits of device resolution. The characteristics observed when programming these devices were all completed at a rate of 1 PLC. In the future we plan to use microscale voltage pulses to affirm our assumption that smaller pulses will lead to higher resolution programmability. Also, the cyclic I-V characteristic in Fig. 2 shows a resistance ratio of about 100. This implies that the conductivity range of this device is something much greater than the smaller programmable range that was found when pulsing the device. We will continue these experiments to determine if a greater conductivity range can be programmed at a high resolution, leading to more programmable states.

References

[1] G. S. Snider, "Cortical Computing with Memristive Nanodevices," SciDAC Review, (2008).

[2] S. H. Jo, K.-H. Kim, W. Lu, "High-Density Crossbar Arrays Based on a Si Memristive System" Nano Lett., 9(2), 2009, pp. 870-874.

[3] C. Yakopcic, R. Hasan, and T. M. Taha, "Memristor Based Neuromorphic Circuit for Ex-Situ Training of Multi-Layer Neural Network Algorithms," IEEE IJCNN, 2015.

[4] T. M. Taha, R. Hasan, and C. Yakopcic, "Memristor Crossbar Based Multicore Neuromorphic Processors," IEEE International SOCC, 2014.

[5] C. Yakopcic and T. M. Taha, "Determining optimal switching speed for memristors in a neuromorphic system," Electronics Letters, 51(21), 2015, pp. 1637-1639.

[6] F. Alibart, E. Zamanidoost, and D.B. Strukov, "Pattern classification by memristive crossbar circuits with ex-situ and in-situ training," Nature Comm., vol. 4, Jun. 2013.

[7] C. Yakopcic, M. Z. Alom, T. M. Taha, "Memristor Crossbar Deep Network Implementation Based on a Convolutional Neural Network" IEEE/INNS International Joint Conference on Neural Networks, pp. 963 – 970, July, 2016.

[8] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, V. Srikumar, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," ACM/IEEE 43rd Annual International Symposium on Computer Architecture, pp. 14-26, June, 2016.

[9] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y, Xie, "PRIME: A Novel Processing-in-memory Architecture for Neural Network Computation in ReRAM-based Main Memory," ACM/IEEE

43rd Annual International Symposium on Computer Architecture, pp. 27-39, June, 2016.

[10] C. Yakopcic, R. Hasan, and T. M. Taha, "Memristor Based Neuromorphic Circuit for Ex-Situ Training of Multi-Layer Neural Network Algorithms," IEEE IJCNN, 2015.

[11] S. Wang, W. Wang, C. Yakopcic, E. Shin, G. Subramanyam, and T. M. Taha "Lithium Based Memristive Device" IEEE National Aerospace and Electronics Conference, 2015.

[12] S. Wang, W. Wang, C. Yakopcic, E. Shin, G. Subramanyam and T. M. Taha, "Experimental study of LiNbO3 memristors for use in neuromorphic computing," Microelectronic Engineering 168 (2017) 37–40.

[13] R. Hasan and T. M. Taha, "Enabling Back Propagation Training of Memristor Crossbar Neuromorphic Processors," IEEE/INNS International Joint Conference on Neural Networks, 2014.

[14] H. Li, Y. Xia, B. Xu, H. Guo, J. Yin, and Z. Liu, "Memristive behaviors of LiNbO3 ferroelectric diodes," Appl. Phys. Lett., 97, 012902, (2010).

[15] X. Pan, Y. Shuai, C. Wu, W. Luo, X. Sun, H. Zeng, S. Zhou, R. Bottger, X. Ou, T. Mikolajick, W. Zhang, and H. Schmidt, "Rectifying filamentary resistive switching in ion-exfoliated LiNbO3 thin films," Applied Physics Letters, 108, 032904 (2016).

[16] X. Liu, K. P. Biju, J. Lee, J. Park, S. Kim, S. Park, J. Shin, S. Md. Sadaf, and H. Hwang, "Parallel memristive filaments model applicable to bipolar and filamentary resistive switching," Appl. Phys. Lett., 99, 113518 (2011).

[17] A. S. Oblea, A. Timilsina, D. Moore, and K. A. Campbell, "Silver chalcogenide based memristor devices," in Proc. Int. J. Comp. Netw., Oct. 2010, pp. 1–3.

[18] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," Nano Letters, vol. 10, pp. 1297–1301, Mar. 2010.

[19] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized Memristive Device SPICE Model and its Application in Circuit Design," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32(8) August, 2013 pp. 1201-1214.

[20] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Memristor SPICE Modeling," in Advances in Neuromorphic Memristor Science and Applications, R. Kozma, R. Pino, and G. Pazienza (eds), Springer. July 2012.

[21] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, S. Rogers, "A Memristor Device Model," IEEE Electron Device Letters, 30(10), 1436-1438, October 2011.

[22] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Impact of Memristor Switching Noise in a Neuromorphic Crossbar" IEEE National Aerospace and Electronics Conference, June, 2015.

[23] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Memristor SPICE Model and Crossbar Simulation with Nanosecond Switching Time," IEEE International Joint Conference on Neural Networks (IJCNN), August 2013.