Fully Analog ReRAM Neuromorphic Circuit Optimization using DTCO Simulation Framework

Anh Nguyen
Electrical Engineering
San Jose State University
San Jose, USA
anh.d.nguyen@sjsu.edu

Hoi Nguyen
Electrical Engineering
San Jose State University
San Jose, USA
hoi.nguyen@sjsu.edu

Sruthi Venimadhavan
Electrical Engineering
San Jose State University
San Jose, USA
sruthi.venimadhavan@sjsu.edu

Ayyaswamy Venkattraman
Mechanical Engineering
University of California
Merced, USA
vayyaswamy@ucmerced.edu

David Parent
Electrical Engineering
San Jose State University
San Jose, USA
david.parent@sjsu.edu

Hiu Yung Wong*
Electrical Engineering
San Jose State University
San Jose, USA
hiuyung.wong@sjsu.edu

***Abstract*— Neuromorphic inference circuits using emerging devices (e.g. ReRAM) are very promising for ultra-low power edge computing such as in Internet-of-Thing. While ReRAM synapse is used as an analog device for matrix-vector-multiplications, the neuron activation unit (e.g. ReLU) is generally digital. To further minimize its power and area consumption, fully analog neuromorphic circuits are needed. This requires Design-Technology Co-Optimization (DTCO). In this paper, we use our Software+DTCO framework for fully analog neuromorphic inference circuit optimization using ReRAM as an example. The interaction between software machine learning, ReRAM, current comparator, and ReLU are studied. It is found that the neuromorphic circuit is very robust to the variation of ReLU, which confirms the importance of DTCO simulation.**

Keywords—ReLU, ReRAM, DTCO, Neuromorphic, Machine Learning, Circuit Simulation, Verilog-A

I. INTRODUCTION

Machine learning (ML) using von Neumann architecture is facing the bottleneck issue of inefficient power consumption and large latency [1]. The issue is due to the physical separation of memory and computing units, causing most clock cycles to be used for intensive data exchange between the units rather than processing the information. Recently, various architectures have been proposed to overcome the bottleneck [2][3]. Among them, neuromorphic circuits which are analog circuits inspired by human brain architecture have been demonstrated to be a potential solution to the bottleneck. The major difference is to break the physical separation between memory and computing units by combining them at the same location to obviate data movement. The neuromorphic circuit can be regarded as an analog neural network in a memory crossbar array. The key component in most neuromorphic circuits is the analog memory device implemented using emerging memories. In recent years, emerging memories such as Resistive Random-Access Memory (ReRAM) [4][5], Phase Change Memory (PCM) [6], Ferro-Electric FET (FeFET) [7] have gained significant attention.

The neuromorphic circuit performance, however, depends strongly on various parameters such as input voltage range, loading impedance, temperature and sneak current in the cross-bar array interconnection [8] due to the non-linearity of both the memory device and the peripheral circuits. Since ML algorithms have built-in fault tolerance, the requirement of the circuit precision can be less stringent. So, it is important to have a Design-Technology-Co-Optimization (DTCO) framework to understand the interaction between the emerging memory, the circuit, and the ML algorithm to obtain the best trade-off. Moreover, the circuit performance is expected to be also dependent on the ML algorithm being used. Therefore, to achieve the ultimate optimization for various applications, it is necessary to co-optimize the ML algorithm (software) and circuits and devices (hardware).

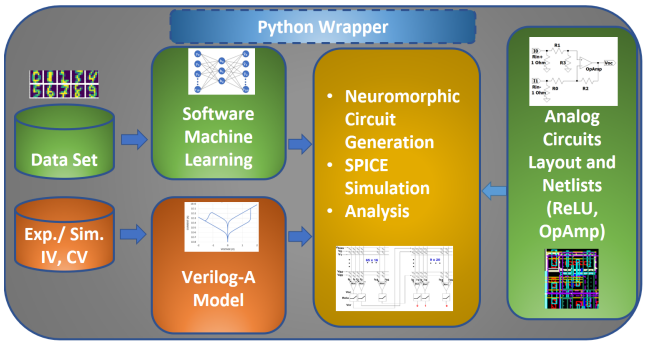


Fig. 1: Software+DTCO framework used. Verilog-A module is used when emerging devices do not have SPICE compact model.

In [3], a MATLAB framework has been built to study the temperature, loading resistance, and input voltage range effect independently. It only studies the precision of circuit behavior instead of the accuracy of the final ML outputs. In [9], a system-level simulator is built using behavior models. Despite its lower speed, SPICE simulation provides more insight into circuit design and avoids the use of behavior models. In [10], SPICE is used but only for studying the behavior of loading resistance and has no interaction with ML algorithm design.

In this paper, we improve and use our Software+DTCO framework [11] to study the effect of various ReLU circuits on a ReRAM inference circuit performance (Fig. 1). The circuit is a neural network (NN) for hand-written digit recognition using ReRAM. Its accuracy in predicting hand-written digits is studied as a function of ReLU circuit designs.

*Corresponding Author: hiuyung.wong@sjsu.edu

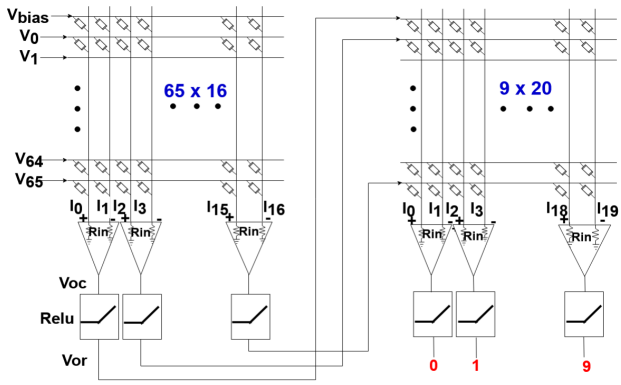


Fig. 2: Schematic showing the generated circuit for hand-written character recognition (only 1-layer case is shown for clarity.)

II. Software+DTCO Framework

Fig. 1 shows the Software+DTCO simulation framework for the neuromorphic inference circuit. The framework is written in python. Users can design a neuron network, NN, (by choosing the number of hidden layers and internal nodes) to train a machine in software for their purpose (e.g. hand-written character recognition). A neuromorphic SPICE circuit will be generated based on the NN designed in software (Fig. 2). Users can pick the emerging memory to be tested (e.g. ReRAM) and the weights in NN found in software training will be loaded to the SPICE compact model or Verilog-A model instances of the emerging memory. Users can also choose the type of sub-circuits (e.g. various ReLU circuit designs) to be included in the simulation. The sub-circuits may be post-layout netlists in which layout-dependent parasitic resistances and capacitances are included. Users can also choose various levels of accuracy for the components (e.g. whether an OpAmp is described by an ideal model or built from transistors of certain technologies). The framework will then invoke a commercial SPICE simulator to simulate the circuit to extract the desired metrics (e.g. inference accuracy) by performing a statistically meaningful number of simulations. The simulations can be performed in parallel. It is important to note that for certain NN design that gives the best accuracy in software (e.g. larger number of layers) might perform worse in neuromorphic circuits due to error propagation [11]. Therefore, Software+DTCO simulation is

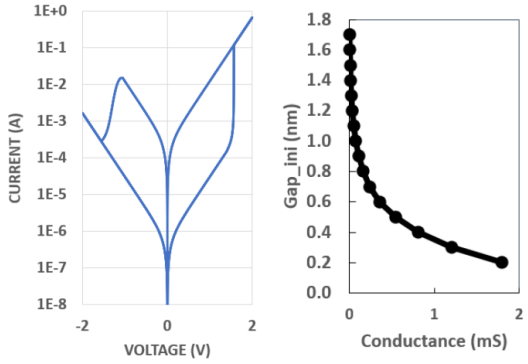


Fig. 3: Simulated ReRAM IV (left) and filament gap size as a function of ReRAM conductance (Right).

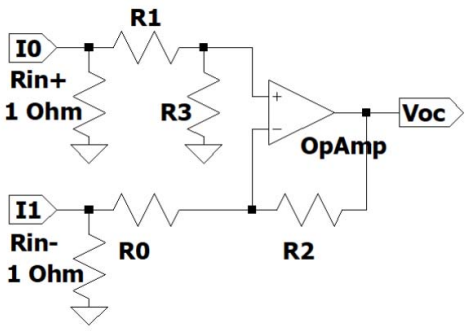


Fig. 4. Current subtractor and current-to-voltage convertor used in this study.

important for the ultimate optimization to not just select an NN that gives the best performance in software but also find a neuromorphic circuit that has the best performance for a given emerging memory and peripheral circuit technologies.

As an example, in this paper, hand-written digit recognition using NN is used [12]. The ReRAM model is obtained, modified, and calibrated based on [13] (Fig. 3). The ReRAM device and circuit are co-optimized so that the ReRAM filament gap has the required resistance range (2.5kΩ and 300kΩ) for the neuromorphic circuit (Fig. 3). Users can then choose different circuits and layouts to realize various functions in the neuromorphic circuit. Here, the effects of the current comparator and ReLU designs on the accuracy of the inference circuit are studied. 180 images are tested for inference accuracy. In the neuromorphic circuit simulation, it is found that 3-layer NN gives the highest accuracy [11]. So only 3-layer neuromorphic circuits are simulated unless specified.

III. Current Subtractor and Current-Voltage Convertor

Since ReRAM can only have positive conductance, to encode the negative weight from the NN, duplicated ReRAM strings are used followed by the current subtractor (Fig. 2). The current subtractor also converts the current to a voltage signal (Fig. 4). The input resistors to the subtractor, R_{in+} and

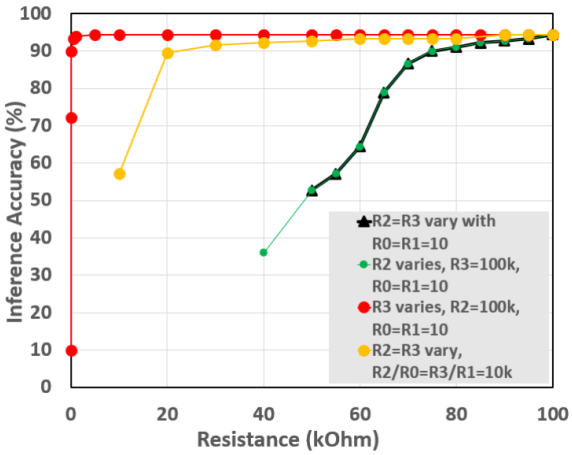


Fig. 5: Accuracy of the neuromorphic circuit (3-layer) as a function of resistor variation in the subtractor and converter.

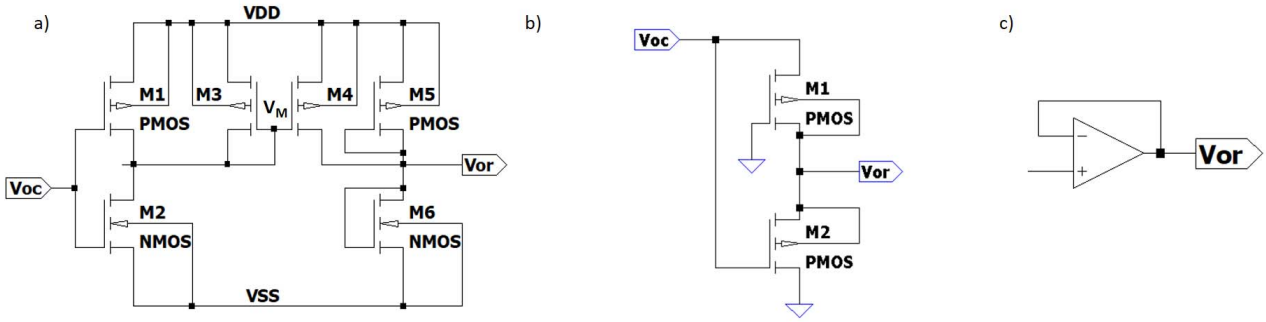


Fig. 6: Analog ReLU circuits used. a) 6T-ReLU based on [14]. b) 2T-ReLU. c) Unity-gain buffer is added to a) and b) to form 6T-ReLU-Buf and 2T-ReLU-Buf, respectively.

R_{in-}, are chosen to be 1 Ω so that it is low enough compared to the ReRAM resistance.

For an ideal OpAmp, it can be derived that $V_0 = A(I_0 R_{in+} - I_1 R_{in-})$. With $R_0 = R_1 = 10\Omega$, $R_2 = R_3 = 100k\Omega$, it can be showed that $A = R_2/R_0 = 10^4$. The effect of resistor variations in the subtractor and convertor is then studied. Fig. 5 shows the results. When R_2 and R_3 are reduced together, the prediction accuracy of the circuit is higher than 90% until the resistance is dropped by 25% to 75kΩ. Further study shows that this is limited by R_2. If only R_3 is varied, the accuracy is still high even R_3 reaches 100Ω. This is because of the large value of A being used. If A is kept constant but the absolute values of R_0, R_1, R_2, and R_3 are changed proportionally, it is found that the accuracy decreases significantly when $R_{2,3}$=20kΩ and $R_{0,1}$=2Ω. This is because $R_{0,1}$ is now close to $R_{in+,in-}$, and the equation derived is no longer valid and the subtractor cannot function as it is supposed to be. This example shows that while the impact of resistance variation can be predicted qualitatively by circuit analysis, the inference prediction accuracy can only be obtained by using DTCO simulation.

IV. Analog ReLU Design

ReLU function is critical in NN and it retains the value if the input is positive and gives zero otherwise (Fig. 7). An analog ReLU circuit with 6 transistors (6T-ReLU) is adopted from [14] using 45nm design rules. The circuit is shown in Fig. 6a. The input inverter (formed by M1 and M2) inverts the input voltage V_{oc} into V_m, the gate voltage of M4, to turn on/off M4. When V_{oc} is negative, V_m is positive and M4 is in the cut-off region. The output voltage V_{or} is tuned to 0V by adjusting M5 and M6 of the voltage divider.

When V_{oc} is positive, V_m is low and M4 is in the saturation region. The output voltage V_{or} is derived from the small-signal model as in [14],

$$\Delta V_{or} = g_{m1} g_{m2} R_1 R_2 \Delta V_{oc} \quad (1)$$

with,

$$g_{m1} = (g_{m,M1} + g_{m,M2}) \quad (2)$$

$$R_1 = (r_{o,M1} || r_{o,M2} || 1/ g_{m,M3} || r_{o,M3}) \quad (3)$$

$$g_{m2} = g_{m,M4} \quad (4)$$

$$R_2 = (r_{o,M4} || 1/ g_{m,M5} || r_{o,M5} || 1/ g_{m,M6} || r_{o,M6}) \quad (5)$$

where g_m's and r_o's are the transconductances and output impedances of the corresponding transistor, respectively. By sizing the transistors so that $g_{m1} g_{m2} R_1 R_2 = 1$, the circuit in Fig. 6a will then function as a ReLU circuit.

This circuit was designed for CMOS synapses which have large input impedance (due to small gate current in CMOS). Therefore, its output impedance is too large for ReRAM synapses because of the low ReRAM input impedance (Note that the output of the ReLU is used as the input of the next NN layer, Fig. 2). Fig. 7 shows that when the loading to the ReLU circuit is in the order of the maximum ReRAM conductance of the technology used, it does not behave as a ReLU function. However, the accuracy still reaches 73% in the 1-layer case but degrades quickly in the 3-layer case due to error accumulation (Fig. 8). Therefore, a unity gain buffer (Fig. 6c) is added and the accuracy can reach 94% in the 3-layer case.

To simplify the circuit, based on [15] and [16], a new 2T-RELU is proposed (Fig. 6b). Assume the threshold voltages ($V_{th,M1}$, $V_{th,M2}$) of M1 and M2 are zero, when V_{oc} is negative, M2 is turned on because $V_{GS,M2} = V_{oc} - V_{or}$ and V_{or} can be discharged to 0V, while M1 is turned off as $V_{GS,M1} = 0V - V_{oc} > 0V$. However, the threshold voltages of M1 and M2 are not set to zero but with finite negative values. Therefore, for small negative (or positive) values of V_{oc}, V_{or} is determined by the voltage divider formed by the two off-state transistors, M1 and M2 (Fig. 7). When V_{oc} is positive, M2 is turned off because $V_{GS,M2} > V_{th,M2}$. But M1 is turned on as $V_{GS,M1} = -V_{oc} <$

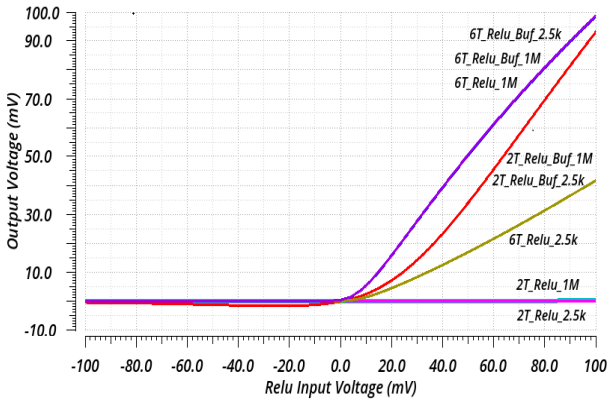
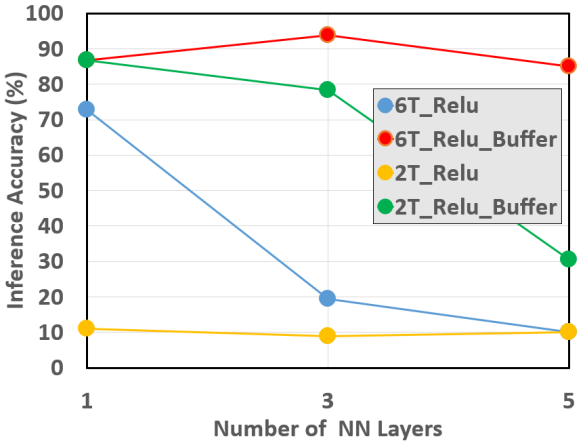


Fig. 7. Performance of the ReLU circuits in Fig. 6 with 1MΩ and 2.5kΩ loadings.

$V_{th,M2}$. Therefore, the behavior of 2T-ReLU can be summarized as



`Fig. 8. Inference accuracy as a function of ReLU designs and number of NN layers.

$$V_{or} \approx 0V \text{ if } V_{oc} \leq 0V \quad (2)$$
$$V_{or} \approx V_{oc} \text{ if } V_{oc} > 0V \quad (3)$$

which is essentially a ReLU circuit with non-idealities.

However, the analysis above is only true if there is no loading. When there is loading with small input resistance (such as that of the next layer ReRAM), V_{or} will be pulled down substantially. As can be seen in Fig. 7, the 2T-ReLU does not behave as a ReLU even with loading impedance is as large as 1MΩ and gives very bad inference accuracy (Fig. 8). Therefore, an output buffer needs to be added (Fig 6c). Although with a buffer, it still has non-idealities, the neuromorphic circuit performs well with an accuracy of 87% in the 1-layer case. Therefore, by using the framework, we found a simpler analog ReLU that can achieve 87% accuracy for this task which is impossible without this framework.

V. Conclusions

A software+DTCO simulation framework is constructed in which users can perform co-optimization of synapses (traditional CMOS or emerging devices) and neuron and software ML. A fully analog ReRAM neuromorphic inference circuit is optimized using the framework. It is found that the inference accuracy is very robust. Moreover, the circuit can tolerate up to 25% variation of current comparator resistance value and is robust against impedance mismatching between ReLU neuron and ReRAM synapse. A simplified 2T-ReLU design is proposed and verified by using this framework and can achieve 87% accuracy despite the non-idealities.

Acknowledgment

This project was supported by San Jose State University College of Engineering Small Group Project Team Fund (2019).

References

[1] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahna, "Estimation of energy consumption in machine learning," Volume 134, December 2019, Pages 75-88.doi: 10.1016/j.jpdc.2019.07.007

[2] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning", 2018 J. Phys. D: Appl. Phys. 51 283001.

[3] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," 2016 53rd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2016, pp. 1-6. doi: 10.1145/2897937.2898010

[4] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S. P. Wong, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," Adv. Mater. 2013, 25, 1774–1779. https://doi.org/10.1002/adma.201203680.

[5] H. -S P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–Oxide RRAM," Proceedings of the IEEE, vol.100, no.6, pp.1951,1970, June 2012, doi: 10.1109/JPROC.2012.2190369

[6] S. Kim, M. Ishii, S. Lewis, T. Perri, M. BrightSky, W. Kim, R. Jordan, G. W. Burr, N. Sosa, A. Ray, J.-P. Han, C. Miller, K. Hosokawa, and C. Lam, "NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning," 2015 IEEE International Electron Devices Meeting (IEDM), Washington, DC, 2015, pp. 17.1.1-17.1.4. doi: 10.1109/IEDM.2015.7409716

[7] X. Sun, P. Wang, K. Ni, S. Datta, and S. Yu, "Exploiting Hybrid Precision for Training and Inference: A 2T-1FeFET Based Analog Synaptic Weight Cell," 2018 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2018, pp. 3.1.1-3.1.4. doi: 10.1109/IEDM.2018.8614611

[8] M. Hu, H. Li, Q. Wu and G. S. Rose, "Hardware realization of BSB recall function using memristor crossbar arrays," DAC Design Automation Conference 2012, San Francisco, CA, 2012, pp. 498-503.

[9] P. Chen, X. Peng and S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," 2017 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2017, pp. 6.1.1-6.1.4, doi: 10.1109/IEDM.2017.8268337.

[10] P. Gu, B. Li, T. Tang, S. Yu, Y. Cao, Y. Wang, H. Yang, "Technological exploration of RRAM crossbar array for matrix-vector multiplication," The 20th Asia and South Pacific Design Automation Conference, Chiba, 2015, pp. 106-111. doi: 10.1109/ASPDAC.2015.7058989

[11] H. Cao, T. Lam, H. Nguyen, A. Venkattraman, D. Parent, and H. Y. Wong, " Study of ReRAM Neuromorphic Circuit Inference Accuracy Robustness using DTCO Simulation Framework," Accepted by IEEE Workshop on Microelectronics and Electron Devices, 2020.

[12] Dua, D., and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California

[13] Jiang, Z., et al., (2014). Stanford University Resistive-Switching Random Access Memory (RRAM) Verilog-A Model. nanoHUB.

[14] J. Zhu, Y. Huang, Z. Yang, X. Tang and T. T. Ye, "Analog Implementation of Reconfigurable Convolutional Neural Network Kernels," 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Bangkok, Thailand, 2019, pp. 265-268, doi: 10.1109/APCCAS47518.2019.8953177.

[15] M. Yilmaz, B. A. Tunkar, S. Park, K. Elrayes, M. A. E. Mahmoud, E. Abdel-Rahman, and M. Yavuz, "High-efficiency passive full wave rectification for electromagnetic harvesters ," Journal of Applied Physics 116, 134902 (2014); https://doi.org/10.1063/1.4896668.

[16] Priyanka P., et al., CMOS Implementations of Rectified Linear Activation Function in VLSI Design and Test. VDAT 2018. Communications in Computer and Information Science, vol 892.