

A Mixed-Mode Neuron with On-Chip Tunability for Generic Use in Memristive Neuromorphic Systems

Sagarvarma Sayyaparaju, Ryan Weiss and Garrett S. Rose

2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, China

©2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Citation Information (BibTex):

```
@INPROCEEDINGS{sayyaparaju2018mixed,  
    author="Sagarvarma Sayyaparaju and Ryan Weiss and  
          Garrett S. Rose",  
    title="A Mixed-Mode Neuron with On-Chip Tunability for  
          Generic Use in Memristive Neuromorphic Systems",  
    booktitle="IEEE Computer Society Annual Symposium on VLSI,  
          2018"  
    month="July",  
    year="2018",  
    address="Hong Kong, China",  
    pages="441-446"  
}
```

A Mixed-Mode Neuron with On-Chip Tunability for Generic Use in Memristive Neuromorphic Systems

Sagarvarma Sayyaparaju, Ryan Weiss and Garrett S. Rose

Department of Electrical Engineering & Computer Science

University of Tennessee, Knoxville

Knoxville, Tennessee 37996 USA

Email: {ssayyapa, rweiss2, garose}@utk.edu

Abstract—Memristors are two-terminal nanoscale devices that provide an efficient way of implementing non-volatile synaptic weights. Realization of large scale memristive neuromorphic systems is reliant on rigorous custom design of neurons that are optimized to accumulate and spike according to the conductance range of the specific memristor employed. However, each new custom neuron design entails meticulous design effort and exorbitant re-fabrication costs. Circumventing this issue, we propose a generic mixed-mode neuron suitable for use across multiple memristor device implementations. The proposed neuron’s accumulation rate is tunable with a set of reference voltages that are provided externally, through the pins on the chip. Hence, the proposed neuron exhibits on-chip accumulation rate tunability and is independent of the specific memristor chosen. This is exemplified in this paper by considering a small scale shape recognition network that employs these neurons. Moreover, the proposed neuron is shown to consume less energy per spike in comparison to a traditional integrate and fire neuron built in the same technology while occupying comparable silicon (layout) area. Lastly, the proposed neuron is shown to reduce the overall pin count for large scale memristive neuromorphic systems.

1. Introduction

Neuromorphic systems are an ensemble of circuits designed to resemble the processing scheme in biological neural systems, at the hardware level. These neural systems consist of processing units called neurons interconnected with plastic synapses, whose connection strength (termed the “weight”) is modifiable [1]. Such systems are deviant from the predominantly sequential data flow based traditional von Neumann machines.

Synapses in neuromorphic circuits provide the “weighted” electrical interconnection between the neurons. The weight of the synapse determines the strength of the signal passed on from the neuron preceding the synapse (“pre-neuron”) to the one succeeding it (“post-neuron”). Although synapses have been shown using devices such as resistors, capacitors and floating-gate devices, as mentioned in [2], a synapse with a programmable non-volatile weight that lends itself for efficient learning remained elusive.

A nanoscale two-terminal device termed as the “memristor” was demonstrated in 2008 in the seminal paper [3], which was proposed theoretically in 1971 [4]. Soon, this device was slated as a good candidate for use as a synapse in CMOS-memristor hybrid neuromorphic systems [5]. Memristors are two-terminal devices whose instantaneous conductance represents the synaptic weight. This conductance state of the device is non-volatile, and can be modified by supplying an appropriate amount of charge/flux to the device. Thus, the memristor allows for the synaptic weights to be carefully programmed and controlled. Also, the flux flowing into the device can be designed to produce online learning behavior such as the spike-timing-dependent plasticity (STDP) [6]. Silicon neurons when used in conjunction with memristor based synapses give rise to memristive neuromorphic systems. The following section dwells further into the implementation details of such systems.

2. Related Work

Authors in [5], [7] have demonstrated STDP based learning in the memristor device, using an integrate and fire model for charge accumulation in the neuron. Using a similar spiking neuron, asynchronous memristor-based architectures were built in [8] to emulate a visual cortex. Authors in [9] have built and simulated a memristor crossbar based character recognition system. In [10], an experimental demonstration of a memristor crossbar performing image classification has been shown. In [11], a scalable memristor-based neural chip has been presented. All of the above memristive synapse based neural systems were demonstrated using an analog integrate and fire neuron [12] that employs a capacitor for charge integration. For the success of such systems, these neurons must accumulate at the desired rate, for which their capacitance value needs to be designed to suit the specific memristor type being used. This implies that these neurons must be custom designed for the specific application being considered.

An extensive review of other silicon neuron implementations can be found in [13]. However, it may be noted that the neuron designs summarized therein either implement conductance based dynamics within the neuron or employ

capacitive charge-discharge based mechanism for their operation. This implies that these neurons are also required to be designed to function for predefined synaptic weights, which makes them rigid and necessitates a redesign for a new set of synaptic weights that may be desired.

The design of memristive neuromorphic systems with these existing neurons thus relies on the type of memristor chosen, since the neurons in the system need to be tailored for the currents provided by the memristor. However, redesigning neurons for every specific application and/or for every memristor device type proves costly not only in terms of the design effort, but also in terms of the fabrication expenses for the new lithographic masks used each time a new neuron is fabricated.

Addressing the aforementioned issues, in this paper, we propose the design of a mixed-mode neuron that is generic in terms of the synapses that can be used with it. The main contributions of this paper can be summarized as follows:

- 1) The proposed neuron is suitable for use across multiple memristor device implementations and thereby eliminates redesign needs and recurring mask costs.
- 2) The proposed neuron is capacitor-free, thus eliminating the need for rigorous custom analog design.
- 3) On-chip tunability using reference voltages for adaptation across a wide range of memristive synaptic implementations is illustrated.
- 4) Reduction in the overall pin count for large scale neuro-chip implementations is shown.

3. Case Study

In order to illustrate the on-chip tunability shortcoming of traditional analog neurons, we present a case study by considering a small-scale shape recognition task using analog integrate and fire neurons and a bi-memristor synapse.

3.1. Bi-Memristor Synapse

A bi-memristor synapse consists of two memristors connected between the pre-neuron and the post-neuron as shown in Figure 1 [14]. When the pre-neuron “spikes”, the incoming spike is translated as voltages of opposite polarity (V_{DD} and V_{SS} in this case) on the pre-neuron end of the memristors while the post-neuron end is held at a ground voltage, as the post-neuron is in the accumulation phase. The weight of this synapse is directly proportional to its effective conductivity, given by:

$$G_{eff} = \frac{1}{M_p} - \frac{1}{M_n}$$

The bi-memristor synapse implements non-volatile synaptic weights that can be both positive and negative. When $M_p < M_n$, the weight is positive whereas it is negative when $M_p > M_n$. A positive weight leads to a net positive current flowing into the post-neuron and vice-versa.

The current flowing into/out of the post-neuron leads to a change in the membrane voltage V_{mem} , which is compared

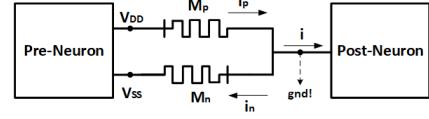


Figure 1. Current flow through a bi-memristor based synapse when the pre-neuron spikes and the post-neuron accumulates [14].

against a threshold voltage (V_{th}). When the membrane voltage exceeds the threshold, the neuron sends out an action potential/spike at its output and a feedback voltage is applied at its input node, that causes the synaptic weight to update depending on the pre- and post-neurons’ spike timing [14].

3.2. Basic Shape Recognition Network

3.2.1. Operation Principle. We consider a simple shape recognition network here that can identify one of the shapes among a set of four shapes as shown in Figure 2. In this case, the goal for the network is to identify a triangle shape input. This implies that the network’s output neuron must spike appropriately when a triangle shape is given as the input, but not for the other shapes.

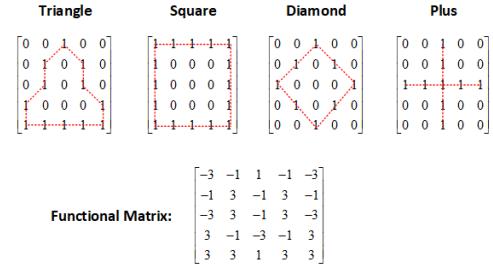


Figure 2. Matrices used to denote the input shapes to the network and the functional matrix used to set the synaptic weights therein.

In order to build such a network, we first start with calculating the functional matrix that consists of the synaptic weights of the network. This matrix is generated by multiplying the triangular matrix by 4 and subtracting the shape matrices for the other shapes (because these must not be recognized). The resultant matrix is shown in Figure 2.

Using this functional matrix, the network is built as shown in Figure 3. Here, the input notation at the neuron p/q stands for *weight/delay*, where *weight* is the synaptic weight and *delay* denotes the number of clock cycles the input signal is delayed before being passed on to the synapse. It may be observed from Figure 2 that columns 1&5 and 2&4 in the functional matrix are identical. Hence, in the network in Figure 3, the neuron N0 is responsible for the columns 1&5, N1 for columns 2&4 and N2 for column 3. This is evident by the two connections N0&N1 have to the output neuron N5, each corresponding to one column in the matrix. Also, the figure shows the thresholds for each neuron (the numbers inside them), implying that a neuron with a threshold of m will spike when it receives an input current corresponding to a net weight greater than or equal to m .

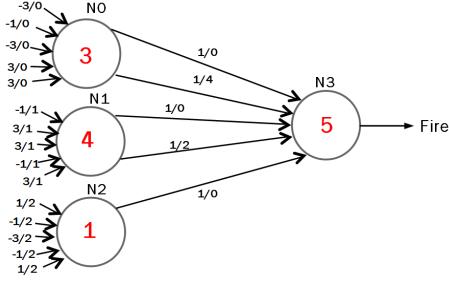


Figure 3. The network topology for the simple shape recognition task.

This network was implemented using the bi-memristor synapse and the integrate and fire neuron circuit mentioned in [14]. The CMOS circuits were built using a 65nm process design kit [15], whereas the memristor itself was modeled in Verilog-A [16]. The memristor's resistance range used here is $2k\Omega$ - $10k\Omega$, based on the values reported in [17]. The results of the simulation are shown in Figure 4.

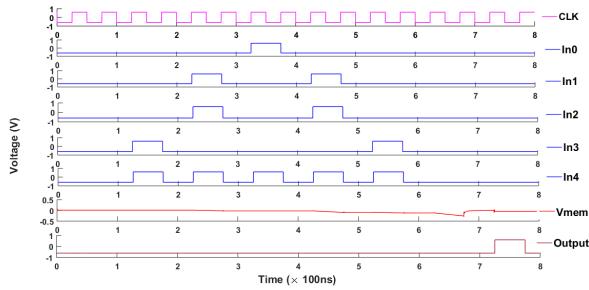


Figure 4. Simulation results for the shape recognition task. It is seen that the output neuron of the network spikes for a triangular shape input.

It can be seen from Figure 4 that the output neuron in the network produces a spike for a triangle input, as intended. Hence, this network can identify a triangle shape input.

3.2.2. Adaptability Test for the Analog Neuron. To test the adaptability of this network with integrate and fire neuron across multiple memristor types, we consider two more memristor types with their resistances as shown in Table 1. Figure 5 shows the simulation results for the network designed in Section 3.2.1 with the three memristor types.

TABLE 1. THE THREE MEMRISTOR TYPES CONSIDERED IN THIS WORK.

	M1 [17]	M2 [18]	M3 [19]
Low Resistance	$2k\Omega$	$5k\Omega$	$30k\Omega$
High Resistance	$10k\Omega$	$50k\Omega$	$300k\Omega$

It is seen from Figure 5 that the network designed to work with the memristor M1 does not work for the other two memristor types, since it does not produce any output spike for a triangle shape input. The reason for this failure can be seen from the graph plotted for the accumulated potential V_{mem} in the output neuron in Figure 5. It is seen that V_{mem} is too low (since the other two memristors' conductances are

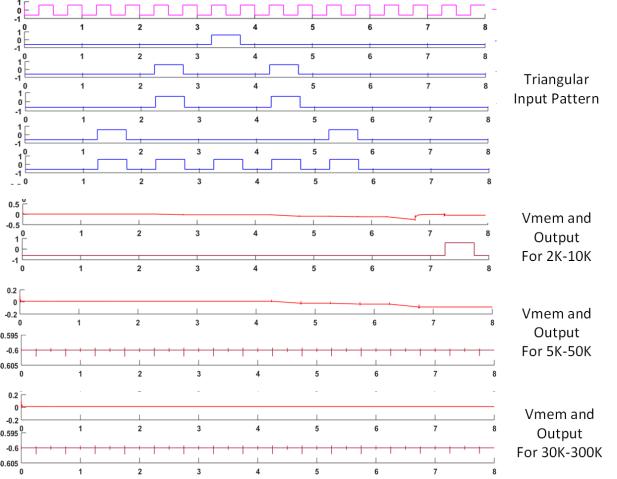


Figure 5. Simulation results for the network designed for M1 memristor being used with the other two memristors without redesigning the neuron.

lower) to cross the threshold of the neuron. This implies that for the network to function for a new memristance range, one of the two factors in the neuron have to change: either the neuron thresholds have to be adjusted to differentiate between the new weights (and hence the new V_{mem}), or the capacitance in the neuron has to be redesigned to adjust the accumulation rate to obtain reasonable threshold voltages that are sufficiently apart as in the earlier case.

Table 2 shows the analysis for the accumulated voltage in the neuron designed for a given memristor, when applied to other devices. This table shows that a neuron designed to work for a given memristance range accumulates too much voltage when used with other devices with lower resistance ranges and thereby saturates at the supply rail voltage. This makes it ineffective by not being able to differentiate between higher weight values. Similarly, it is also seen that a neuron designed for a lower memristance range (has a high capacitance value and hence) accumulates too small of a voltage on the capacitor to differentiate between the weights for higher resistance values.

TABLE 2. THE NEURON ACCUMULATION BEHAVIOR FOR VARIOUS DESIGN AND USAGE COMBINATIONS.

	Used with M1	Used with M2	Used with M3
Designed for M1	Works	V_{mem} too low	V_{mem} too low
Designed for M2	V_{mem} saturates	Works	V_{mem} too low
Designed for M3	V_{mem} saturates	V_{mem} saturates	Works

The foregoing analysis dictates that an integrate and fire neuron designed to suit a given memristor does not accumulate charge at a desired rate to distinguish between the weights for other device types. Also, the accumulated voltages for these other device types are indistinguishable and hence cannot be differentiated by tuning the threshold voltages of these neurons on-chip. This leads us to the conclusion that these neurons must be custom designed to suit the specific memristor under consideration.

4. The Proposed Mixed-Mode Neuron

In this section, we propose a mixed-mode neuron that mitigates the need for custom neuron design for different memristor devices. This neuron takes-in an analog current from the synapses, senses the current that flowed in by comparing it to some references, and encodes it as a digital value. This value is then accumulated on a digital register and compared to a digital threshold value to produce a spike. The block diagram for this neuron is shown in Figure 6.

The circuit level complexity of the sensing block and the digital block are both a function of the chosen n . In our case, since the network in Figure 3 has a maximum of +5 as the neuron threshold, we choose to encode the synaptic weights from -5 to +5. Hence, $n = 4$.

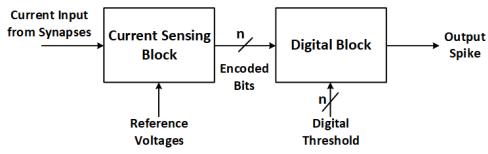


Figure 6. Block diagram for the proposed mixed-mode neuron.

4.1. The Current Sensing Block

Figure 7 shows the schematic for the current sensing block used as the first stage in the neuron. The incoming current from the synapses flows into the sensing transistor, which is biased in the linear region. A transistor biased in the linear region can have a range of currents flowing through it, depending on the V_{DS} of the device. Hence, the incoming current helps setup an appropriate DC bias point with a particular drain voltage at the sensing transistor. This drain voltage is used as the “sensed voltage” corresponding to the input current to the neuron.

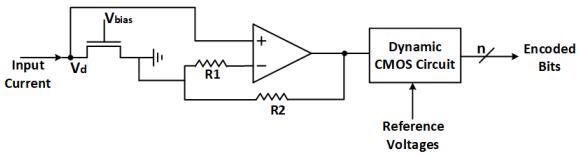


Figure 7. The circuit used to “sense” the incoming current. The V_{bias} here is chosen so as to maintain the sensing transistor in the linear mode.

Although the current-to-voltage conversion could be done with a resistor-divider network, the currents flowing into the neuron can vary up to two orders of magnitude, leading to a huge variation in the sensed voltages with such a system. However, in our case, we have biased the transistor’s source terminal with the mid-rail voltage (which in our case is *gnd!*). Hence, the sensed voltages for huge difference in currents will not be as huge.

The sensed voltage V_d at the drain of the sensing transistor is input to an op-amp based non-inverting amplifier to widen the gap between the sensed voltages for different

weight values. The amplified voltage is fed as the input to a dynamic CMOS circuit that encodes the sensed voltage as an n -bit (which is 4 in this case) signed digital value.

The dynamic CMOS circuit consists of a set of transistor pairs to categorize the neuron current as a corresponding synaptic weight. Each pair of transistors has a pre-charge transistor and a discharge transistor as shown in Figure 8. For positive weight detection, during the positive half of the clock cycle, the pre-charge transistor charges up the common node E_{px} (called the *evaluation node* here) to a supply rail voltage (V_{DD} in this case). During the negative half of the cycle, pre-charge transistor is cut-off, while the discharge transistor is responsible for the discharge of E_{px} . This discharge rate is dependent on the extent to which this transistor is switched on, which is in turn dependent on its V_{GS} . The gate voltage V_G is the sensed voltage coming from the output of the amplifier and the source voltage V_S is a reference voltage coming into the neuron from an on-chip pin. Hence, by adjusting V_S , V_{GS} can be modified to control the discharge rate of the node E_{px} . An analogous discussion holds true for negative weight detection nodes E_{nx} .

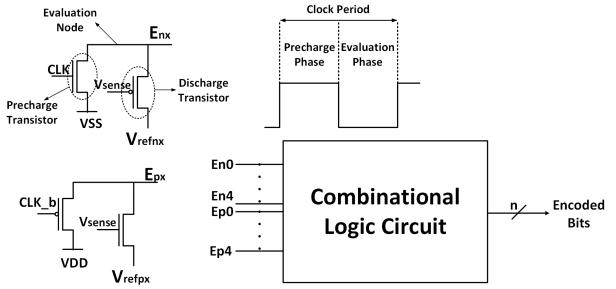


Figure 8. The dynamic CMOS circuit that encodes the sensed voltage. Here, $x = 0, 1, 2, 3, 4$ and hence consists of five positive and five negative evaluation nodes. Note that the combinational logic block here implements the truth table shown in Table 3.

The sensing block consists of 5 evaluation nodes for positive weights and 5 for negative weights. Each of these nodes’ extent of discharge is dependent on the input synaptic weight (which determines the gate voltage for the discharging transistors) and the external reference voltages. The reference voltages are applied (progressively) in such a manner that a certain number of nodes are fully discharged for a specific weight. Table 3 shows the truth table depicting the input synaptic weight versus the node voltages that are fully discharged in the evaluation phase of the clock and the corresponding digital value that is encoded for it. It is seen from this table that as the absolute value of the synaptic weight increases, more number of evaluation nodes are discharged. These nodes are pulled to either of the supply rails by a buffer and these buffer outputs are used by a combinational logic circuit to encode the digital bits according to the truth table shown in Figure 3.

The reference voltages V_{refnx} and V_{refpx} are in turn dependent on the sensed voltages for the particular synapse and are determined accordingly, to implement the scheme in Table 3. These sensed voltages change for a different kind

TABLE 3. THE TRUTH TABLE FOR THE OPERATION OF THE DYNAMIC CMOS CIRCUIT AND THE COMBINATIONAL LOGIC ACCOMPANYING IT.

Synaptic Weight	Nodes Discharged	Encoded Digital Output
+5	Ep5-Ep0	0101
+4	Ep4-Ep0	0100
+3	Ep3-Ep0	0011
+2	Ep2-Ep0	0010
+1	Ep1, Ep0	0001
0	Ep0, En0	0000
-1	En0, En1	1111
-2	En0-En2	1110
-3	En0-En3	1101
-4	En0-En4	1100
-5	En0-En5	1011

of memristive synapse. Hence, to implement the abstract scheme of Table 3, a new set of pin voltages are needed for each new device. Therefore, just by adjusting external reference voltages, we can still implement the same abstract idea of Table 3. Hence, our sensing block possesses on-chip tunability to modify the discharge rate of its sensing circuits and thereby maintain its abstract functionality. This is in contrast to the integrate and fire neuron, where the accumulation rate is non-tunable post-fabrication.

4.2. The Digital Block

The digital block performs a digital mode storage and comparison to produce the neuron's output spike. The topology used here is shown in Figure 9. The digital block takes in the encoded digital bits from the sensing block described above in Section 4.1. A digital adder is used to add these incoming encoded bits to the existing data stored in the memory register and the result is stored back on the register. The resultant value on the adder is then also compared against a digital threshold value in the digital comparator block (which is a combinational circuit comparing two n -bit signed values) to produce an output spike when the adder output is greater than or equal to the digital threshold value.

Additionally, the neuron must also spike when the result of the addition in the digital adder is a positive overflow, indicating that the result of the addition is greater than the threshold. Hence, the spike control logic here is a combinational circuit which produces the neuron spike by checking the overflow condition in the adder and the comparison result of the digital comparator. When the neuron spikes, the register is cleared asynchronously. This clear signal is active until the neuron spike is valid at the output, thus implementing the refractory period of the neuron.

4.3. Shape Recognition with the Proposed Neuron

As described in Section 4.1, the proposed neuron's accumulation and firing characteristics can be controlled and tuned on-chip using reference voltages that control the discharging rate of the evaluation nodes in the sensing block. We demonstrate this by using the proposed neuron

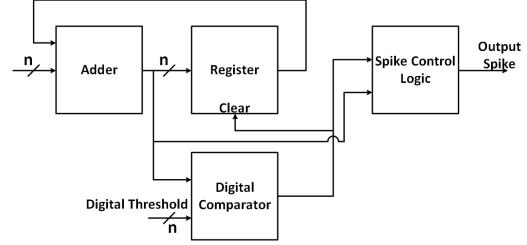


Figure 9. The digital half of the neuron, performing digital mode accumulation and comparison.

to perform the shape recognition task described in Section 3.2.1. The simulation results are shown in Figure 10.

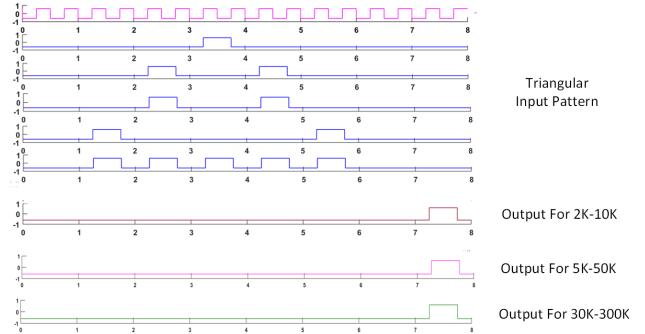


Figure 10. Simulation results for the shape recognition task performed using the proposed mixed-mode neuron. Note that the shape recognition task was successful regardless of the memristor device used.

The results shown in Figure 10 demonstrate that the proposed neuron allows for the task to be performed using all the three cases. In all the three cases, the network was able to perform the triangle-pattern recognition task. To adapt to the varying synaptic currents in each case, the reference voltages to the proposed neuron were applied so that the discharge rate of the evaluation nodes in the sensing block remains the same. Hence, Figure 10 illustrates the fact that the proposed neuron allows for on-chip tunability.

5. Discussion and Conclusion

Table 4 compares the merits of the proposed neuron as opposed to an integrate and fire neuron. To evaluate the area efficiency, we have compared the layout area for both the circuits when implemented with a 65nm process design kit. Table 4 shows that the estimated area of the proposed neuron is $2010.9 \mu\text{m}^2$ and is independent of the synapse since the neuron's design does not change across implementations. However, the capacitance used in the analog neuron here changes with the memristor used, and hence the area. The capacitors used in the designs used for the memristors M1, M2 and M3 are 8pF, 2.8pF and 0.467pF respectively. The layout area for these designs were $2382 \mu\text{m}^2$, $1992 \mu\text{m}^2$ and $1817 \mu\text{m}^2$ respectively, amounting to an average area of $2063.67 \mu\text{m}^2$. Hence, the area of the analog neurons here is comparable to that of the proposed neuron.

TABLE 4. COMPARISON TABLE FOR THE PROPOSED NEURON VERSUS THE INTEGRATE AND FIRE NEURON.

Design Aspect	Proposed Neuron	Integrate and Fire Neuron
On-chip tunability	Yes	No
Need for custom design	No	Yes
Recurring mask cost	No	Yes
Capacitor-based circuit	No	Yes
Estimated area (μm^2)	2010.9	2063.67
Pin Count (m neurons)	11	m
Energy per spike (pJ)	5.327	23.292

Although the proposed neuron increases the pin count for a discrete neuron, it actually reduces it for a large scale system. This can be explained as follows: for a system that employs m neurons, the analog neuron needs m threshold pins. Whereas in case of the proposed neuron, we need 4 pins for threshold per neuron, implying a total of $4m$ pins in addition to the 10 reference voltages to set the discharge rate. However, since threshold for this neuron is a digital value, all these $4m$ values for the thresholds can be input to the system using a serial-in, parallel-out shift register. Thereby, a carefully controlled data transfer protocol can help reduce the pin count to $1 + 10 = 11$ for the whole system. Hence, for large scale systems where $m > 11$ is typically valid, the proposed neuron helps reduce the pin count of the system.

Additionally, the energy per spike of the proposed neuron was evaluated to be 5.327pJ when using a 50ns clock period, whereas the integrate and fire neuron built in the same technology was found to use 23.292pJ per spike. Thus, the proposed neuron is also energy efficient in comparison to a traditional analog neuron.

In conclusion, this work has demonstrated the incompatibility of custom made analog neurons across multiple memristive synapse implementations. The generic mixed-mode neuron proposed here alleviates the need for rigorous analog design, does not require redesign/re-fabrication effort/cost and has been illustrated to be suitable for large scale memristive neuromorphic systems. It is worth to note that the discussion here draws some parallel with the advantages a field programmable gate array (FPGA) has over an application specific integrated circuit (ASIC).

Acknowledgments

This material is based in part on research sponsored by Air Force Research Laboratory under agreement number FA8750-16-1-0065. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

- [1] Y. Tang, J. R. Nyengaard, D. M. De Groot, and H. J. G. Gundersen, "Total regional and global number of synapses in the human brain neocortex," *Synapse*, vol. 41, no. 3, pp. 258–273, 2001.
- [2] S. P. Adhikari, C. Yang, H. Kim, and L. O. Chua, "Memristor bridge synapse-based neural network and its learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1426–1435, 2012.
- [3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, p. 80, 2008.
- [4] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.
- [5] 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, no. 4, pp. 1297–1301, 2010.
- [6] G.-q. Bi and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.
- [7] X. Wu, V. Saxena, and K. Zhu, "A cmos spiking neuron for dense memristor-synapse connectivity for brain-inspired computing," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–6.
- [8] B. Linares-Barranco, T. Serrano-Gotarredona, L. A. Camuñas-Mesa, J. A. Perez-Carrasco, C. Zamarreño-Ramos, and T. Masquelier, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in neuroscience*, vol. 5, p. 26, 2011.
- [9] A. M. Sheri, H. Hwang, M. Jeon, and B.-g. Lee, "Neuromorphic character recognition system with two pcmo memristors as a synapse," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2933–2941, 2014.
- [10] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, p. 61, 2015.
- [11] J. M. Cruz-Albrecht, T. Derosier, and N. Srinivasa, "A scalable neural chip with synaptic electronics using cmos integrated memristors," *Nanotechnology*, vol. 24, no. 38, p. 384011, 2013.
- [12] X. Wu, V. Saxena, and K. A. Campbell, "Energy-efficient stdp-based learning circuits with memristor synapses," in *Machine Intelligence and Bio-inspired Computation: Theory and Applications VIII*, vol. 9119. International Society for Optics and Photonics, 2014, p. 911906.
- [13] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, p. 73, 2011.
- [14] G. Chakma, M. M. Adnan, A. R. Wyer, R. Weiss, C. D. Schuman, and G. S. Rose, "Memristive mixed-signal neuromorphic systems: Energy-efficient learning at the circuit-level," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2017.
- [15] S. Amer, M. S. Hasan, and G. S. Rose, "Analysis and modeling of electroforming in transition metal oxide-based memristors and its impact on crossbar array density," *IEEE Electron Device Letters*, vol. 39, pp. 19–22, 2018.
- [16] S. Amer, S. Sayyaparaju, G. S. Rose, K. Beckmann, and N. C. Cady, "A practical hafnium-oxide memristor model suitable for circuit design and simulation," in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–4.
- [17] J. J. Yang, M.-X. Zhang, J. P. Strachan, F. Miao, M. D. Pickett, R. D. Kelley, G. Medeiros-Ribeiro, and R. S. Williams, "High switching endurance in tao x memristive devices," *Applied Physics Letters*, vol. 97, no. 23, p. 232102, 2010.
- [18] 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.
- [19] M. Uddin, M. B. Majumder, and G. S. Rose, "Robustness analysis of a memristive crossbar puf against modeling attacks," *IEEE Transactions on Nanotechnology*, vol. 16, no. 3, pp. 396–405, 2017.