Neuromorphic Character Recognition System With Two PCMO Memristors as a Synapse

Ahmad Muqeem Sheri, *Student Member, IEEE*, Hyunsang Hwang, *Member, IEEE*, Moongu Jeon, *Member, IEEE*, and Byung-geun Lee, *Member, IEEE*

***Abstract*—Using memristor devices as synaptic connections has been suggested with different neural architectures in the literature. Most of the published works focus on simulating some plasticity mechanism for changing memristor conductance. This paper presents a neural architecture of a character recognition neural system using Al/$Pr_{0.7}Ca_{0.3}MnO_3$ (PCMO) memristors. The PCMO memristor has an inhomogeneous barrier at the aluminum and PCMO interface which gives rise to an asymmetrical behavior when moving from high resistance to low resistance and vice versa. This paper details the design and simulations for solving this asymmetrical memristor behavior. Also, a general memory read/write framework is used to describe the running and plasticity of neural systems. The proposed neural system can be produced in hardware using a small 1 K crossbar memristor grid and CMOS neural nodes as presented in the simulation results.**

***Index Terms*—Character recognition, memristors, neural classifier, neuromorphic.**

I. Introduction

NEUROMORPHIC devices and systems have seen an upsurge recently in research due to the interest in using memristive devices as synaptic connections. Neural networks, particularly in embedded form, are used in many industrial applications. In [1], neural networks have been used to solve the friction compensation problem, while in [2], pulse-coupled neural networks are presented for color image segmentation. Also, in [3], recurrent neural networks have been used for modeling predictive control of nonlinear dynamical systems. In [4], applications of neural networks in power electronics and motor drives are presented in detail. From an application point of view in industrial electronics, the need is always to have a fast and small form factor intelligent system, and therefore, such applications become prime targets for the use of neuromorphic devices in the near future. In [5], the first modeling of neural networks with memristors as synapses was presented, with emphasis on making "crummy" (large scale, easy to manufacture, and cheap but error prone) memristors and then using the fault tolerance of neural networks on large-scale problems to develop self-organized neural systems. In [6], a cosputtered Ag and Si active layer memristor with Ag/Si mixture ratio gradient was used to show spike time-dependent plasticity, using positive and negative pulses. A more detailed and biologically more plausible scenario is presented in detail in [7]. This work [7] uses models of actual action potentials produced in biological neurons and uses those pulses to produce simulation results on mathematically modeled memristors.

Some other works [8]–[11] also present how some other memristors can be used as synaptic junctions in neural networks in specific and in neuromorphic systems generally. In [8], a spiking neural network is presented using crossbar-based memristors, while [9] presents data about the immunity of memristors when used in neural systems. Another pulse-based programmable memristor circuit is presented in [10] while Seo *et al.* [11] detail spike timing-dependent plasticity characteristics in a TiO_2 bilayer resistive switching device. In all of the aforementioned works, the memristors either increase or decrease their conductance for a positive pulse and vice versa for a negative pulse. Although, in real devices, the increase and decrease in conductance are not identical for the same shape of positive and negative pulses, this can be handled through changing the pulsewidth or pulse amplitude. This behavior can be considered as symmetric because the use of positive pulses takes the memristor toward one side of its conductance spectrum while negative pulses produce the opposite effect, and both of these changes are gradual. This, in general, means that one can use positive pulses for one type of potentiation while negative pulses for the other, and also, a combination of a positive and a negative pulse can be established without any change in the conductance due to the canceling effect of each other.

This paper tackles the problem of using a praseodymium calcium manganese oxide ($Pr_{0.7}Ca_{0.3}MnO_3$, PCMO) device which is asymmetric. Fig. 1 shows a plot of 200 potentiation pulses of −4 V, 10 ms followed by 600 pulses of 1 V, 10 ms for a PCMO memristor device. As Fig. 1 shows, for negative pulses of −4 V, we have a gradual increase in conductance, while for the positive pulse of just 1 V, a very sudden decrease in conductance is evident. In fact, a 1-V pulse resets the memristor device to its minimum possible conductance. This kind of device therefore cannot be used in many of the proposed neural systems where the increase and decrease in conductance are

Manuscript received October 24, 2012; revised March 2, 2013 and June 18, 2013; accepted July 7, 2013. Date of publication August 1, 2013; date of current version December 20, 2013. This work was supported in part by the Systems Biology Infrastructure Establishment Grant (2013) provided by the Gwangju Institute of Science and Technology and the Pioneer Research Center Program through the National Research Foundation of Korea funded by the Ministry of Science, Information and Communication Technologies and Future Planning under Grant 2012-0009462.

A. M. Sheri and M. Jeon are with the School of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea (e-mail: amsheri@gist.ac.kr; mgjeon@gist.ac.kr).

H. Hwang is with the Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang 790-784, Korea (e-mail: hwanghs@postech.ac.kr).

B. Lee is with the School of Mechatronics, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea (e-mail: bglee@gist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2013.2275966

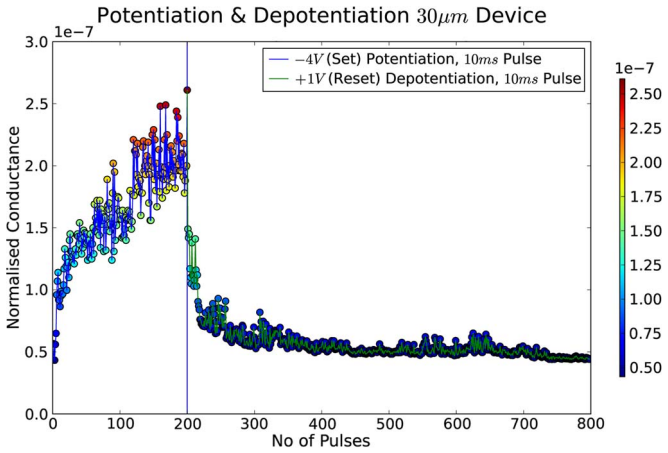

Fig. 1. Conductance change of the PCMO device under 200 potentiation and 600 depotentiation pulses.

achieved by applying opposite potential voltages relying on the fact that the memristor will react gradually in both directions. This paper presents a two-memristor structure as one synapse to solve this problem. A two-phase-change-memory synapse [12] has been suggested, which is proven energy efficient for neuromorphic architectures. This paper presents a model for a PCMO-based device with a simpler spiking model and also lists some interesting properties which can be achieved using this approach.

The rest of this paper is organized as follows. First, the background is presented in Section II, where general neural systems are explained from a simplistic point of view so that they become easier to think about hardware implementations. Section III details the memristor type used in this paper. In Section IV a two-memristor-synapse model is presented to tackle the asymmetric nature of a PCMO memristor, followed by the details of how to read and write conductance in the two-memristor-synapse model. Then, the details of simulation are presented in light of a character recognition application in Section V.

II. Background

Neural architectures can always be seen as a system with two main parts. One is the neuron, and the other is the synapse connecting those neurons, which behaves as a message transmission line between neurons. The signals passing through these connections are modulated [13] in some way, which means that we can consider the connections as basic processing units with defined functionality. The neural bodies aggregate the signal coming through the synapses and perform further processing on all the inputs together to produce some output. The properties of the neural connection and neuronal bodies in light of neuromorphic considerations are presented next.

A. Neural Connections (Synapses)

Neural connections are links between two neurons, where one neuron behaves as a source and the other behaves as a sink. Sometimes, this source and sink classification is not very well defined, thus giving rise to bidirectional connections [13]. The source neuron sends a signal that passes through the connection which usually applies some function onto the signal, and then, the resultant signal is presented to the sink neuron. Mostly, the function applied by the neural connections to the message passing through is a simple scaling of the message by a known factor called the weight of the connection.

From a system's point of view, a neural connection has some memory element which stores the scaling factor or weight of that connection. It always scales the input signal by the weight stored in it. Furthermore, there is some external or internal mechanism available to change the weight stored in a connection. This last step is necessary for learning and is called the synaptic plasticity.

Let us consider a connection between two neurons of a strict feedforward network. In that case, it is customary to consider the neural connection as a weighted edge. The edges store a weight in its memory; when a signal is presented to the input node, the signal is multiplied by the weight of the connection, and then, the resulting signal is presented to the sink neuron as the output of the connection. This behavior is well established and is used extensively.

The contents of memory for the connection between two neurons is changed based on a learning function. This learning function considers the overall function of the network and then decides what type of weight change is necessary for one connection such that the overall behavior of a network can be optimized toward a required goal (target function).

In summary, we can say that the following functionality makes a usable neural connection.

1) Given an input signal from the source neuron, the connection should change the signal based on its memory content and then pass the resulting signal to the sink neuron. Also, this operation does not affect the memory contents of the connection.
2) Given a memory changing signal, the connection should change its memory contents.
3) Given no signal, the memory content should stay, and no output signal should be produced.

B. Neural Bodies (Neuron Cells)

Neural bodies (neurons) are aggregating units. They receive the input from all incoming connections, almost always aggregate the input, and then apply some nonlinear function to the aggregated sum [14]. After this aggregation and nonlinear transformation step, neurons usually compare the resultant value against a threshold. If the threshold is reached, the neuron is said to go in firing state. In the firing state, the neuron sends a signal to all its output lines, and sometimes, this firing signal is sent to both input and output connections. The firing of a neuron signifies the fact that the neuron was presented with an input which it has seen in the past.

Another aspect which must be mentioned here is the fact that neural systems are basically learning machines, i.e, when a neural network is presented with a set of inputs T, it tries to acquire some sort of experience E from T without remembering each and every instance of T such that, if the neural network is again presented with some examples from T, it would perform

better than before, based on a target function F. One interesting aspect in this definition is the fact that, sometimes, it is easy to distinguish between two different stages in the life of a neural network, namely, training stage and testing stage. This distinction makes the design and analysis of these systems a bit easier, although in most situations, transitions between training and testing phases can be done transparently with minimal or no change.

III. RRAM Memristors

A memristor is a two-terminal device whose conductance can be precisely modulated by the charge or flux through it. We need to understand this property of the memristor to use it as synaptic connections between neurons. Relating back to the summary points enumerated in Section II-A, memristor properties corresponding to each point are as follows.

1) The application of a voltage pulse across a memristor causes a current proportional to the conductance of the memristor to flow through it. This current can be used as a measurement of the conductance of the memristor.
2) All real memristors have a threshold voltage below which they do not change their conductance, but as soon as a voltage higher than the threshold is applied, the conductance of the memristor changes. Thus, for reading a memristor conductance, one can use any voltage below the threshold while a voltage over the threshold will change the conductance of the memristor, therefore giving a write mechanism.

We have used PCMO [15] thin layers for their uniformity as a material and a low power consumption to produce resistive random-access memory (RRAM) for use in neuromorphic hardware. These PCMO devices behave as the metal–semiconductor structure in the low resistance state (LRS) and as the metal–oxide semiconductor structure in the high resistance state (HRS). In the LRS, an inhomogeneous barrier is formed at the aluminum/PCMO interface with an effective barrier height (ϕ_{eff}) of $\sim$0.15 eV. In the HRS, by applying the Poole–Frenkel emission theory, the intrinsic trap energy level (Φ_t) of the AlO_x layer formed at the aluminum/PCMO interface was obtained as $\sim$1.4 eV. On the basis of characterization results of such devices, a simple $4F_2$ cross-point array with a Schottky barrier [16], formed through a redox reaction at the PCMO interface, can be fabricated without any additional fabrication steps. The data presented in this paper are from a 1 K cross-point device.

IV. Proposed Synapse and Architecture

In our PCMO RRAM devices, the long-term depression (LTD) (or simply depression) or HRS is not progressive with the migration of oxygen by using invariant or identical pulses as shown in Fig. 1. Therefore, a two-PCMO-memristor device model in which two PCMO devices constitute one synapse as shown in Fig. 2 is proposed here. The two devices make opposite contributions to the integrated state of a neuron. One is responsible for long-term potentiation (LTP) (or simply potentiation) as LTP contributing device, and the second device causes LTD (or depotentiation) as LTD contributing device.

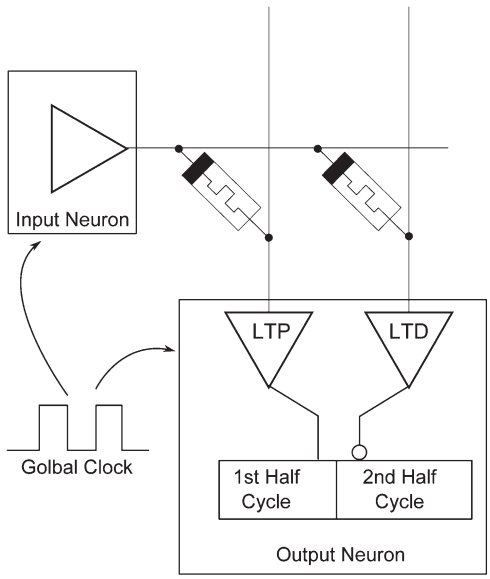

Fig. 2. Two-memristor schemes where one memristor contributes LTP current in the first half of a clock cycle while the second contributes current for LTD in the second half of a clock cycle.

With this scheme, when the synapse needs to be potentiated, the conductance of the LTP device is increased, and that of the LTD device is not changed, which causes an overall potentiation of the double-memristor synapse. Also, by using this same mechanism, moving the LTD contributing device toward the LRS will cause depression. Since all kinds of plasticity for this model will cause either a LTP or LTD device to move from HRS to LRS, we will see sooner or later that either one or both contributing devices saturate toward LRS. To handle this eventual convergence toward the upper bound of LRS for all memristors, we have introduced a sleep cycle mechanism.

Using the double-memristor structure gives us a few benefits, which need to be mentioned here. Memristor devices are almost always asymmetrical for potentiation and depotentiation pulses to some extent. Designing a neuromorphic system using such devices will always require some algorithmic way to minimize the effect of such asymmetry. The proposed two-memristor architecture is presented considering the case where memristor devices show a gradual response of change in conductance for potentiation while the change in conductance is pretty abrupt for depotentiation pulses. Therefore, by using the proposed scheme, it is possible to achieve symmetrical behavior by a memristor for potentiation and depotentiation pulses.

In biological neural systems, some synapses are excitatory which cause the increase in the membrane potential of the sink neurons, while others behave as inhibitory which cause the decrease in the membrane potential of their sink neurons. By the same token, in the neural network literature, neural connections have real numbered weights. Achieving synaptic weights which are evenly spread around zero is very hard if one memristor is used as a synapse. However, using the proposed architecture, real numbered weights between some bounds can be easily achieved. Also, the distribution of weights around zero

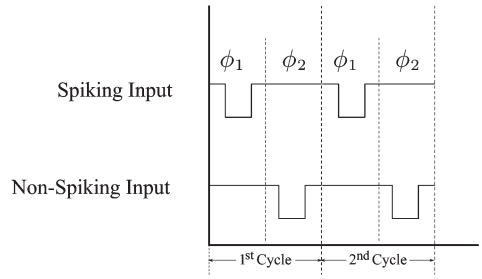

Fig. 3. Spiking input has a voltage pulse in the first half of a clock cycle while nonspiking input has a voltage pulse in the second half of a cycle.

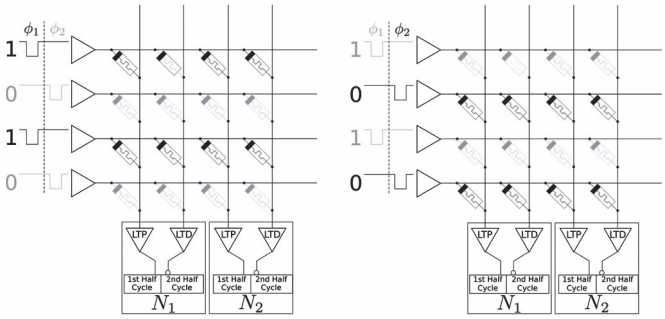

Fig. 4. Read cycle schematic structure: Left figure shows the status of the network in the first half of a clock cycle, and right figure shows the status in the second half of the clock cycle.

is very even in this case. Also, Suri *et al.* [12] suggest that this double-memristor model is more energy efficient.

For detailing the read operation, write operation, and sleep cycle operation, the following scenario is considered. We have two neurons N_1 and N_2. N_1 will be trained while N_2 will not be trained for the pattern 1010.

A. Read Operation

Real memristors have a threshold voltage under which they show change in conductance to a negligible degree [17], [18]. In our device, -2.4 V is the threshold voltage denoted as $V_{\text{threshold}}$. Therefore, we use pulses of -2 V denoted by V_{read} for reading memristive devices. As shown in Fig. 3, a spiking terminal will send a spike to the output neuron in the first half of a clock cycle ϕ_1 while a nonspiking terminal illustrated by 0 will send a pulse in the second half ϕ_2 of the global clock.

In ϕ_1, all currents collected from the LTP memristors will be summed to $I(LTP)_{\phi_1}$, and currents collected from LTD devices will be summed to $I(LTD)_{\phi_1}$ as shown in Fig. 4. The state of the neuron can then be calculated by $N_{\text{state}} = I(LTP)_{\phi_1} - I(LTD)_{\phi_1}$. As we are using pulses to identify nonspiking nodes in ϕ_2, we can actually think of them as spiking nodes in ϕ_1 and use the values as neuron states. Therefore, the values in ϕ_2 are basically a logical NOT for the input, and therefore, the calculating node states in ϕ_1 and ϕ_2 are equivalent.

B. Write Operation

The requirement of a write cycle is to achieve plasticity. For potentiation, the overall weight of a memristor pair should

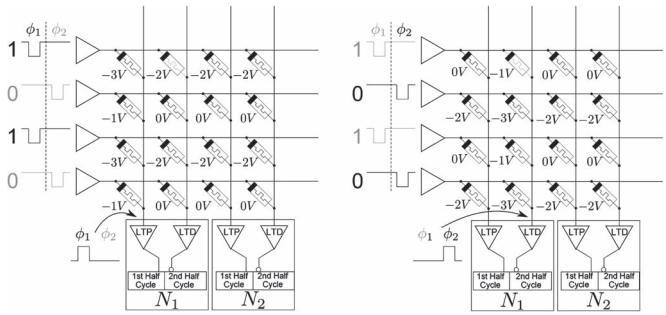

Fig. 5. Write cycle schematics: Left figure shows the pulse format in the first half of a clock cycle causing LTP while right figure shows the pulse format in the second half of a clock cycle which is responsible for LTD.

increase, while for depression, it should decrease. The increase in weight leading to potentiation is required when an input from the input neurons needs to cause the output neuron to fire with high correlation. On the contrary, if we want an input from input neurons not to affect the firing of the output neuron, we want to decrease its weight causing depression.

To further illustrate this, let us consider Fig. 5. Since neuron N_1 is being trained while neuron N_2 is not, only backspikes from N_1 will be produced. Now, in ϕ_1, a backspike of V_{back} will be generated. In our case, we produce a 1-V pulse as the backspike. As Fig. 5 shows, in ϕ_1, we will have $V_{\text{read}} - V_{\text{back}}$ applied to the spiking LTP devices while $-V_{\text{back}}$ will be applied to the nonspiking LTP devices. In our case, $V_{\text{read}} - V_{\text{back}} = -2 - 1 = -3$ V will be applied to the spiking LTP devices which is greater than $V_{\text{threshold}}$. Therefore, the LTP devices with spiking input will increase their conductance, while the memristors which had no pulse in ϕ_1 will experience $V_{\text{read}} - V_{\text{back}} = 0 - 1 = -1$ V, which is below $V_{\text{threshold}}$, resulting in no change of conductance. Similarly, the memristors on the LTD line for the spiking input in ϕ_1 will experience $V_{\text{read}} - V_{\text{back}} = -2 - 0 = -2$ V and nonspiking input $V_{\text{read}} - V_{\text{back}} = 0 - 0 = 0$ V. This tells us that, since both are below $V_{\text{threshold}}$, they undergo no change. Therefore, the net effect of the backspike in ϕ_1 will be an increase in LTP memristor conductance only.

Similarly, in ϕ_2, when spiking inputs become nonspiking while nonspiking ones become spiking, the roles are reversed. LTD devices of spiking input in ϕ_2 will increase their conductance which will result in the overall decrease in the conductance of a memristor pair. This will have the effect of training no input neurons with negative weights.

C. Sleep Operation

The conductance of LTP and LTD devices keeps on gradually increasing during the operation of the network, and therefore, a sleep mechanism is being introduced to make sure that the weights are adjusted between their bounds after a certain lifetime of the network. Biological neural systems are known to reconfigure and refresh memories during sleep time [19].

Let us consider that, for a given input, pulse size, and its duration, the average number of M steps is required for a device

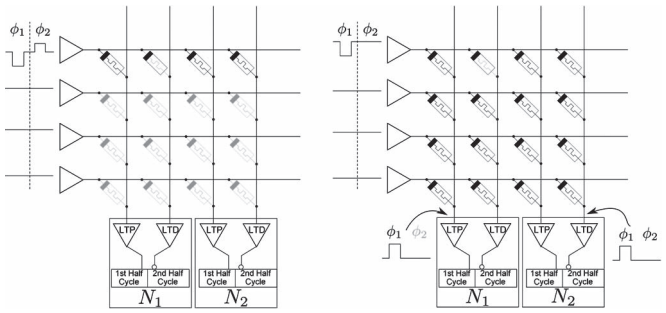

Fig. 6. Sleep cycle schematics: Left figure shows the read pulse followed by the reset pulse while right figure shows how again potentiation is performed.

to go from an HRS to LRS, which naturally entails that, after a minimum of M pulses to the LTP or LTD device, they might reach their maximum LRS. In experiments, it has been observed that our PCMO memristors are extremely fragile at these limits and easily render themselves useless if used at those limits. Therefore, we must run a sleep cycle for the network after every M pulses at maximum. In addition, Suri *et al.* [12] mention that running such cycle involves marginal time/energy cost and therefore can be run for a much smaller number than M. The sleep cycle is illustrated in Fig. 6. When a network goes into sleep mode, all nodes in the network know that the network is in sleep mode, and they run the following algorithm to adjust their weights during sleeping in which no input from outside the network is accepted. The following algorithm is used to reset all memristors' conductance in a sequence and, then, to repotentiate them.

In Algorithm 1, we simply move from one input neuron N to the other one by one. A read–reset pulse as shown in Fig. 6 with V_{read} in ϕ_1 and V_{reset} in ϕ_2 is applied. The read part of the input pulse causes a read operation on the LTP and LTD devices, and we gather $I(LTD)^i_{\text{read}}$ and $I(LTP)^i_{\text{read}}$ for each output neuron i and store their values. V_{reset}, which, in our PCMO device, is anything above $+1.3$ V, is applied to reset all memristors to their minimum states. After that, each output neuron i takes a difference $I^i_{\text{diff}} = I(LTP)^i_{\text{read}} - I(LTD)^i_{\text{read}}$, which tells if the output neuron i needs to backspike on the LTP line or LTD line. If I^i_{diff} is positive, a backspike pulse for the LTP device is generated. Otherwise, the LTD device is backspiked. After storing I^i_{diff}, the input neuron N spikes for M cycles in the ϕ_1 part of the cycle without any other concern, while the output neuron i uses the values of I^i_{diff} backspikes to approximately match the potentiations necessary to establish conductance that could have generated I^i_{diff}. Once M pulses have been applied, the next input neuron is selected, and the same procedure is repeated.

Algorithm 1 Algorithm for Sleep Cycle

for Input Neuron N **do**
  send read and reset pulse
  $I^i_{\text{diff}} = I(LTP)^i_{\text{read}} - I(LTD)^i_{\text{read}}$
  for $j = 0 \rightarrow M$ **do**
    Send read spike in ϕ_1
    if $I^i_{\text{diff}} \geq 0$ and j $\approx$ conductance required for I^i_{diff} **then**
      send back pulse to LTP device in ϕ_1
    end if
    if $I^i_{\text{diff}} \leq 0$ and j $\approx$ conductance required for I^i_{diff} **then**
      send back pulse to LTD device in ϕ_1
    end if
  end for
end for

V. Simulation Results

Considering the immediate requirements of designing a small testable network, which seems biologically plausible [20], a very small feedforward network with only 30 input nodes and 10 output nodes is simulated in the following section. This meant that a $30 * 10 * 2$ memristor array was required to test this network in the real world. The simulator was implemented in Python language, and the simulation was done with sleep cycles after every 300 clock cycles.

A. Memristor Model

It is necessary to discuss the memristor model used in simulation, as we have a number of data samples available with us for real memristor devices. The memristor model used in the simulation requires special mention here (for the detailed properties of the memristor, see [16]). A number of samples of data were available from real memristor devices to us. These data took different memristors from HRS to LRS by the application of small pulses, and one of them is shown in Fig. 7. An exponential function of the form

$$f(x) = -A \exp^{-Bx} + C \tag{1}$$

where A, B, and C are constants, was fitted to the provided data with

$$A = 0.96445349, \quad B = 0.00792457, \quad C = 1.09779073. \tag{2}$$

A linear model and an exponential model $f(x)$ that fit the data are shown in Fig. 7. Another function $g(x)$ was generated by a simple transformation of $f(x)$ with Gaussian noise $\mathcal{N}$ as follows:

$$g(x) = f(x) * \mathcal{N}. \tag{3}$$

The result of (3) fits the original memristor data the best and is shown last in Fig. 7. This detail is important because the exponential model $f(x)$ was used in the sleep cycle to estimate the number of backspikes to be generated, while the function $g(x)$ was used as the memristor growth function during simulations. Also, the minimum conductance and maximum conductance of the memristors were not considered as constants, Gaussian noise was added to the upper and the lower limit of memristor

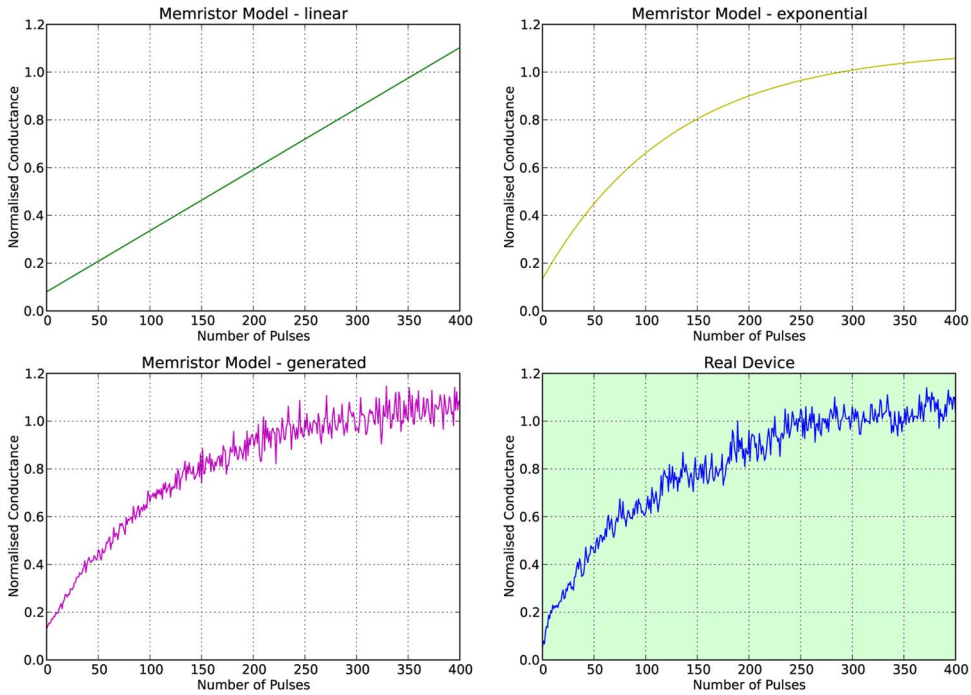

Fig. 7. Memristor conductance models: Three different memristor models (linear, exponential, and simulated) are shown in comparison to real memristor data. The graphs show how the conductance of a memristor device changes as pulses are applied for 400 clock cycles.

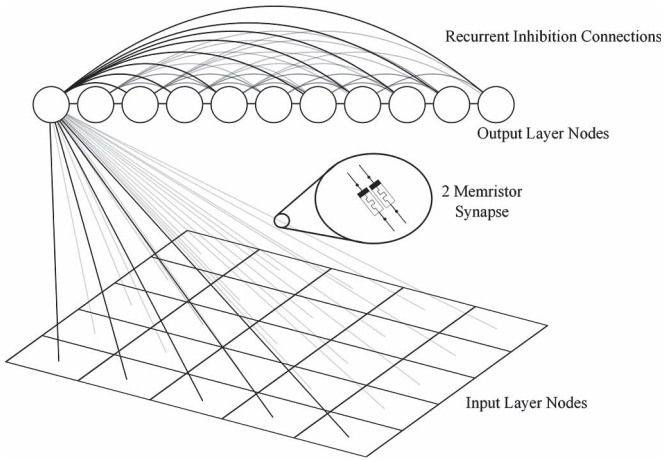

Fig. 8. Neural network structure: Input neural nodes are shown at the bottom which are fully connected to output nodes by synapses (where each synapse has two memristors). Ten output nodes are required to classify ten different input images. Each output node is also connected to all other output nodes with inhibitory connections to implement the winner-take-all scheme.

conductances, and the mean and variance of such noise were calculated by measuring real memristor devices.

B. Network Structure and Learning Stimuli

The topology of the network trained is shown in Fig. 8. The network has two layers, namely, input layers and the output layer. Input layer neurons produce a pulse with the probability of the pixel value being shown to it. That means a pixel value of 0 will be considered as a nonspiking input while a pixel value of 1 will produce a pulse as a spiking input at every clock cycle. The network uses the leaky integrate and fire neurons [21], [22]. These neural bodies integrate their input and fire if a threshold has been reached. Also, a recurrent connection is provided to all the neurons which act as a very strong inhibition signal. Once a neuron fires, it lets every other neuron know that it has fired, and therefore, all the integrate and fire neurons reset their internal state to minimum.

Supervised learning was administered in this networks, which means that, during training, we present input images, then tell the network which neurons to fire for the given input, and then run the network in the training mode. During training, the network adjusts the synapse weights. Algorithm 2 details the general algorithmic procedure followed during the training of the network.

Algorithm 2 Algorithm used for training the neural network. Each training image is given as input to the network for training a given output node.

for $epoch = 0 \rightarrow MaximumTrainingEpochs$ **do**
  for $j = 0 \rightarrow totalTTrainingImages$ **do**
    $trainImage \leftarrow trainImages[j]$
    Send read spikes and calculate node state N_{state}
    Send write spikes for learning node j
  end for
end for

A dynamic threshold scheme was simulated where each neuron would also learn the threshold that it has to fire at. However, in a different run of the simulations, a dynamic threshold had the same results as those of a fixed threshold for

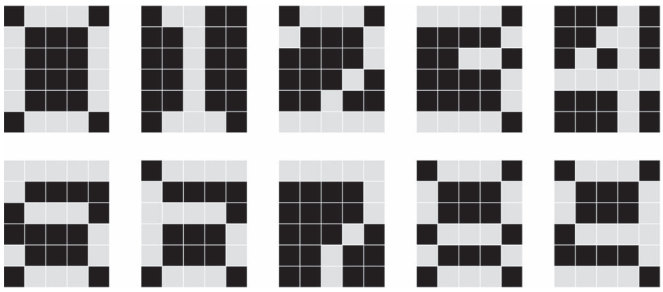

Fig. 9. Ten digit images which are used as input to the network for training.

TABLE I
RECOGNITION RATE FOR DIFFERENT NOISE LEVEL IMAGES

Noise Level	Correct Recognition
5%	99.8%
10%	99.6%
15%	99.1%
20%	98.3%
30%	72.5%

all neural codes. Therefore, for simplicity and easy hardware implementation, fixed threshold values are used, and the results are presented with a fixed threshold only.

C. Network Testing Results

Fig. 9 shows the 5 × 6 pixel input images for which the network was trained. Table I shows the recognition rate of the neural system when noise is added to images [23]. Fig. 10 shows how noise was added to input image 6 to generate noisy spike images. The original images for number 6 are shown in the left column, and uniform random noise mask images are shown in the middle column. Noisy images are acquired by flipping all the pixels in the character images wherever there is a white pixel in the noise mask image as shown in the right column of Fig. 10 for 10% noise. Algorithm 3 details the stepwise procedure used during the testing of the network.

Algorithm 3 Algorithm used for testing the neural network. An image is shown to the network until one of the output node spikes

for $j = 0 \rightarrow totalTestingImages$ **do**
  $testImage \leftarrow testImages[j]$
  while No Ouptut nodes spikes **do**
    Send read spikes and calculate node state N_{state}
    if Any node state N_{state} reaches threshold **then**
      produce output spike
      produce inhibitory spike
    end if
  end while
end for

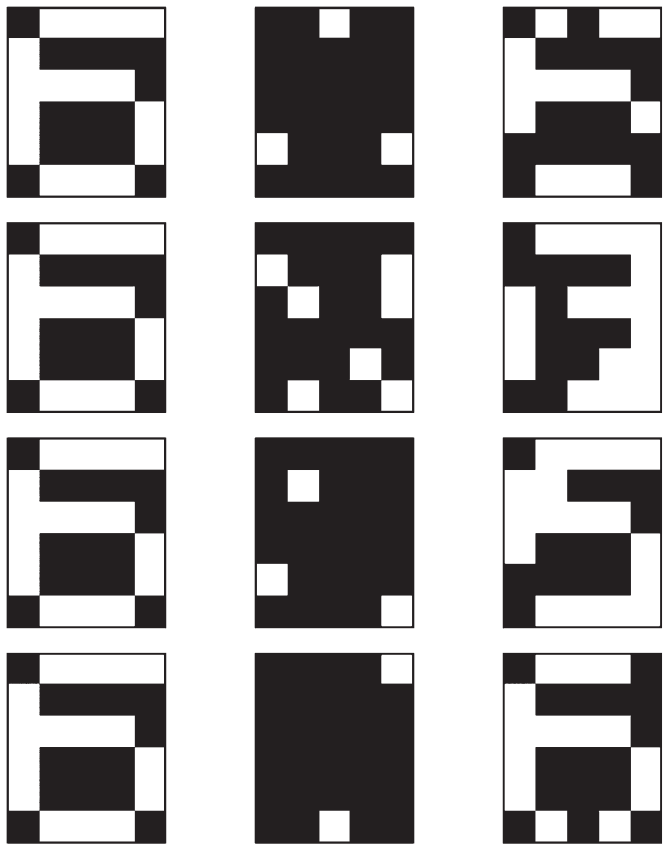

Fig. 10. Input image 6 in the left column is shown. Images with 10% random pixels as white are shown in the middle column to produce noisy image mask. Resulting noisy images used for testing are shown in the right column where all pixels which have noisy image mask as white are flipped.

Fig. 11 shows the internal states of the neurons during testing with 10% noise added to all images. Whenever a node state reaches the threshold, the node produces an output spike and an inhibition spike to all other neurons. Therefore, one can see all nodes resetting to almost zero level whenever a node spikes as shown by the spiking markers in Fig. 11. This is the expected behavior from this kind of feedforward neural network. The recognition rate should be 100% in such networks if noise is not considered in the system. However, because noise is considered for simulating real hardware conditions, the recognition rate falls with increasing noise in the system.

VI. CONCLUSION

Different methodologies have been presented in the literature to implement some form of plasticity in memristive synapses. All such works do present some simulation results of neural systems performing some task. As it is evident from the literature, the memristors behave differently for positive and negative voltages. This paper presented a character recognition system which takes into account this asymmetric property of memristors by using two memristors as one synaptic weight. Using two memristors gives us the benefit of easily creating synaptic weights which are excitatory as well as inhibitory. Also, the proposed system achieves the task of character recognition with

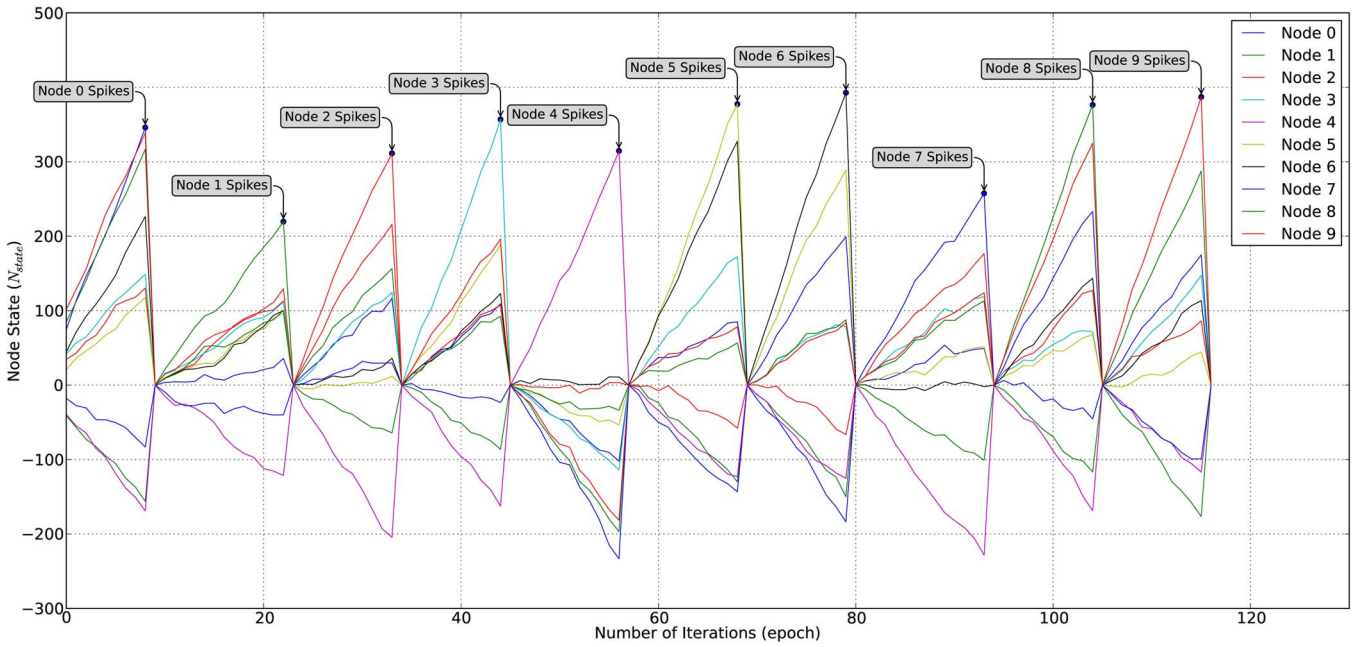

Fig. 11. Node states accumulate the current coming into a neural node in a leaky integrate manner. Whenever a node spikes, depicting that it has recognized an image, it produces an output pulse, resets its own state to zero, and sends an inhibition signal to all other nodes, resetting them to zero also.

noisy inputs, showing the feasibility of using two memristors in larger systems.

REFERENCES

[1] S. Huang and K. K. Tan, "Intelligent friction modeling and compensation using neural network approximations," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3342–3349, Aug. 2012.

[2] H. Zhuang, K.-S. Low, and W.-Y. Yau, "Multichannel pulse-coupled-neural-network-based color image segmentation for object detection," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3299–3308, Aug. 2012.

[3] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3089–3101, Aug. 2012.

[4] B. K. Bose, "Neural network applications in power electronics and motor drives—An introduction and perspective," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 14–33, Feb. 2007.

[5] G. Snider, "Self-organized computation with unreliable, memristive nanodevices," *Nanotechnology*, vol. 18, no. 36, p. 365 202, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1088/0957-4484/18/36/365202

[6] S. Jo, T. Chang, I. Ebong, B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010. [Online]. Available: http://dx.doi.org/10.1021/nl904092h

[7] C. Zamarreño-Ramos, L. Camuñas-Mesa, J. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B. Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers Neurosci.*, vol. 5, 2011. [Online]. Available: http://dx.doi.org/10.3389/fnins.2011.00026

[8] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," in *Proc. ECCTD*, 2009, pp. 563–566.

[9] D. Querlioz, O. Bichler, and C. Gamrat, "Simulation of a memristor-based spiking neural network immune to device variations," in *Proc. IJCNN*, Aug. 5–31, 2011, pp. 1775–1781.

[10] H. Kim, M. Sah, C. Yang, T. Roska, and L. Chua, "Neural synaptic weighting with a pulse-based memristor circuit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 1, pp. 148–158, Jan. 2012.

[11] K. Seo, I. Kim, S. Jung, M. Jo, S. Park, J. Park, J. Shin, K. Biju, J. Kong, K. Lee, B. Lee, and H. Hwang, "Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device," *Nanotechnology*, vol. 22, no. 25, p. 254 023, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1088/0957-4484/22/25/254023

[12] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *IEDM Tech. Dig.*, Dec. 2011, pp. 4.4.1–4.4.4.

[13] S. Cohen-Cory, "The developing synapse: Construction and modulation of synaptic structures and circuits," *Science*, vol. 298, no. 5594, pp. 770–776, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1126/science.1075510

[14] A. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, Jul. 2006. [Online]. Available: http://dx.doi.org/10.1007/s00422-006-0068-6

[15] S. Park, S. Jung, M. Siddik, M. Jo, J. Lee, J. Park, W. Lee, S. Kim, S. Sadaf, and X. Liu, "Memristive switching behavior in pr0. 7ca0. 3mno3 by incorporating an oxygen deficient layer," *Phys. Stat. Sol. (RRL)-Rapid Res. Lett.*, vol. 5, no. 10/11, pp. 409–411, Nov. 2011. [Online]. Available: http://dx.doi.org/10.1002/pssr.201190023

[16] M. Jo, D. Seong, S. Kim, J. Lee, W. Lee, J. Park, S. Park, S. Jung, J. Shin, and D. Lee, "Novel cross-point resistive switching memory with self-formed Schottky barrier," in *Proc. Symp. VLSIT*, 2010, pp. 53–54.

[17] J. Yang, M. Pickett, X. Li, D. A. Ohlberg, D. Stewart, and R. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nat. Nanotechnol.*, vol. 3, no. 7, pp. 429–433, Jul. 2008. [Online]. Available: http://dx.doi.org/10.1038/nnano.2008.160

[18] S. Jo, K. Kim, T. Chang, S. Gaba, and W. Lu, "Si memristive devices applied to memory and neuromorphic circuits," in *Proc. IEEE ISCAS*, 2010, pp. 13–16.

[19] G. Wang, B. Grone, D. Colas, L. Appelbaum, and P. Mourrain, "Synaptic plasticity in sleep: Learning, homeostasis and disease," *Trends Neurosci.*, vol. 34, no. 9, pp. 452–463, Aug. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.tins.2011.07.005

[20] S. Panzeri, E. Rolls, F. Battaglia, and R. Lavis, "Speed of feedforward and recurrent processing in multilayer networks of integrate-and-fire neurons," *Network*, vol. 12, no. 4, pp. 423–440, Nov. 2001.

[21] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *Proc. IEEE ISCAS*, 2009, pp. 2898–2901.

[22] A. Joubert, B. Belhadj, and R. Héliot, "A robust and compact 65 nm LIF analog neuron for computational purposes," in *Proc. IEEE 9th Int. NEWCAS*, 2011, pp. 9–12.

[23] E. Simoncelli and B. Olshausen, "Natural image statistics and neural representation," *Ann. Rev. Neurosci.*, vol. 24, pp. 1193–1216, Mar. 2001. [Online]. Available: http://dx.doi.org/10.1146/annurev.neuro.24.1.1193

Ahmad Muqeem Sheri (S'13) received the B.E. degree in computer software engineering from the National University of Science and Technology, Rawalpindi, Pakistan, and the M.S. degree in computer sciences from the Lahore University of Management and Sciences, Lahore, Pakistan, in 2008. He is currently working toward the Ph.D. degree in the School of Information and Communication, Gwangju Institute of Science and Technology, Gwangju, Korea.

Hyunsang Hwang (M'88) received the B.S. degree in metallurgical engineering from Seoul National University, Seoul, Korea, in 1988 and the Ph.D. degree in materials science from the University of Texas at Austin, Austin, TX, USA, in 1992.

From 1992 to 1997, he was with the LG Semicon Corporation, Korea, as a Principal Researcher. From 1997 to 2012, he was with the Gwangju Institute of Science and Technology, Gwangju, Korea, as a Professor. Since 2012, he has been with the Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea, as a Professor. In recent years, he has engaged in research on RRAM, Flash memory devices, and high-k dielectrics.

Moongu Jeon (M'07) received the B.S. degree in architectural engineering from the Korea University, Seoul, Korea, in 1988 and the M.S. and Ph.D. degrees in computer science and scientific computation from the University of Minnesota, Minneapolis, MN, USA, in 1999 and 2001, respectively.

As a Postgraduate Researcher, he worked on optimal control problems at the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2001–2003, and then moved to the National Research Council of Canada, where he worked on the sparse representation of high-dimensional data and the level set methods for image processing until July 2005. In 2005, he joined the Gwangju Institute of Science and Technology, Gwangju, Korea, where he is currently an Associate Professor in the School of Information and Communications. His current research interests are in machine learning, pattern recognition, and neurocomputing.

Byung-geun Lee (M'07) received the B.S. degree in electrical engineering from Korea University, Seoul, Korea, in 2000 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2004 and 2007, respectively.

From 2008 to 2010, he was a Senior Design Engineer at Qualcomm Inc., San Diego, CA, USA, where he had been involved in the development of various mixed-signal ICs. Since 2010, he has been with Gwangju Institute of Science and Technology, Gwangju, Korea, where he is currently an Assistant Professor in the School of Mechatronics. His research interests are analog and mixed-signal IC design.