Title

A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing

Authors

Kim, Yongtae
Zhang, Yong
Li, Peng

Publication Date

2015

Peer reviewed

A Reconfigurable Digital Neuromorphic Processor with Memristive Synaptic Crossbar for Cognitive Computing

YONGTAE KIM, YONG ZHANG, and PENG LI, Texas A&M University

This article presents a brain-inspired reconfigurable digital neuromorphic processor (DNP) architecture for large-scale spiking neural networks. The proposed architecture integrates an arbitrary number of N digital leaky integrate-and-fire (LIF) silicon neurons to mimic their biological counterparts and on-chip learning circuits to realize spike-timing-dependent plasticity (STDP) learning rules. We leverage memristor nanodevices to build an $N \times N$ crossbar array to store not only multibit synaptic weight values but also network configuration data with significantly reduced area overhead. Additionally, the crossbar array is designed to be accessible both column- and row-wise to expedite the synaptic weight update process for learning. The proposed digital pulse width modulator (PWM) produces binary pulses with various durations for reading and writing the multilevel memristive crossbar. The proposed column based analog-to-digital conversion (ADC) scheme efficiently accumulates the presynaptic weights of each neuron and reduces silicon area overhead by using a shared arithmetic unit to process the LIF operations of all N neurons. With 256 silicon neurons, learning circuits and 64K synapses, the power dissipation and area of our DNP are 6.45 mW and 1.86 mm^2, respectively, when implemented in a 90-nm CMOS technology. The functionality of the proposed DNP architecture is demonstrated by realizing an unsupervised-learning based character recognition system.

Categories and Subject Descriptors: B.7.1 [**Integrated Circuits**]: Types and Design Styles—*Advanced technologies, memory technologies, VLSI (very large scale integration)*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Digital neuromorphic processor, memristor, reconfigurable, silicon neuron, spike-timing-dependent plasticity, spiking neural network, synaptic crossbar array

ACM Reference Format:
Yongtae Kim, Yong Zhang, and Peng Li. 2015. A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing. ACM J. Emerg. Technol. Comput. Syst. 11, 4, Article 38 (April 2015), 25 pages.
DOI: http://dx.doi.org/10.1145/2700234

1. INTRODUCTION

The human brain mediates and produces our thoughts, actions, memory, feelings and other complex tasks, which are all accomplished with great energy and space efficiency. In contrast, for the conventional Von Neumann machines to do the same would consume a tremendous amount of power, energy, and space resources, if not entirely impossible [Merolla et al. 2011].

To date, implementing the Von Neumann architecture in scaled VLSI technologies faces significant challenges as a result of growing process variations, device reliability and power consumption. Brain-inspired neuromorphic computing may offer a

This article is an extension to the preliminary work presented at the IEEE International System-on-Chip Conference (SOCC) [Y. Kim et al. 2012].

Authors' addresses: Y. Kim, Intel Corporation, Santa Clara, CA 95054; email: yongtae.kim@intel.com; Y. Zhang, Cadence Design Systems, San Jose, CA 95134; email: zyong@cadence.com; P. Li, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843; email: pli@tamu.edu.

DOI: http://dx.doi.org/10.1145/2700234

promising architectural solution to overcoming these challenges. In particular, the neuromorphic architecture is well suited for complex processing tasks such as character or image recognition, classification, and language learning with great power efficiency and scalability [Mitra et al. 2009; Seo et al. 2011; Merolla et al. 2011; Serrano-Gotarredona et al. 2009; Brink et al. 2013; Massoud and Horiuchi 2011]. Among these, spiking neural networks are of a particular interest since they more closely resemble biological brains than more traditional artificial neural networks. Furthermore, this non-Von Neumann paradigm comes with inherent error resilience and fault tolerance, which is appealing for large-scale integration in scaled VLSI technologies.

Traditionally, silicon neurons have been implemented with analog circuits that utilize the I-V characteristics of MOS transistors to mimic biological neurons [Mitra et al. 2009; van Schaik 2001; Wijekoon and Dudek 2008; Cosp et al. 2006]. Unfortunately, analog circuit implementations are intrinsically susceptible to process, voltage and temperature (PVT) variations and are difficult to reconfigure and interface. An analog implementation often necessitates the use of area-consuming capacitors to store synaptic weights, which hinders large-scale integration of spiking neurons [Indiveri et al. 2006; Serrano-Gotarredona et al. 2009].

Two digital reconfigurable neuromorphic chips and their building blocks have been recently demonstrated [Seo et al. 2011; Merolla et al. 2011; Imam et al. 2012; Arthur et al. 2012]. These two designs support up to 256 programmable digital neurons as well as 1024×256 binary synapses by means of an SRAM crossbar array, which occupies a significant portion of the entire chip area. Binary synapses with a probabilistic weight update scheme are adopted in Seo et al. [2011]. However, the low synaptic weight resolution can limit learning performance. On the other hand, one limitation of the work of Merolla et al. [2011] is its lack of on-chip learning capability.

Chua [1971] theoretically predicted the existence of memristor, as known as memory resistor, as the fourth fundamental passive circuit element. More recently, TiO_2 thin-film-based memristors have been demonstrated at the nanoscale [Strukov et al. 2008]. The memristive nanodevice has gained increasing research interest and become a promising solution for low-cost on-chip storage thanks to its non-volatility, excellent scalability, and high integration density of 10 Gb/cm^2 or greater [Ho et al. 2011; Yang and Williams 2013]. Several multibit hybrid CMOS/memristor memory architectures targeting high integration density and low power dissipation have been proposed to substitute conventional SRAM and flash memories that are confronted with fundamental technology scaling limits [Merkel et al. 2011; Manem et al. 2012].

Several recent studies have suggested to leverage memristive nanodevices for building neuromorphic synaptic arrays [Jo et al. 2010; Snider 2008; Pershin and Ventra 2010; K.-H. Kim et al. 2012; Hu et al. 2012]. In particular, ideas for implementing analog-based CMOS/memristor neuromorphic circuits have been proposed in the literature. Hu et al. [2012] proposes an analog-based Brain-State-in-a Box (BSB) recall hardware using a memristor crossbar array. Character recognition is demonstrated with a two-layer network on the proposed 256×256 memristor crossbar array. However, this design does not support on-chip learning and it is assumed that all memristors are preprogrammed or already trained. Additionally, the network topology is hardwired to be a fixed two-layer network. In Ebong and Mazumder [2012], hybrid CMOS and memristor circuits based on analog computation are discussed and compared with full CMOS alternatives for a 5×5 two-dimensional position detection network that has 25 neurons and 80 synapses. In Serrano-Gotarredona et al. [2013], the implementation of several spike-timing-dependent plasticity (STDP) learning rules using a combination of memristors and analog CMOS circuits is presented.

While initial good progress has been made in hybrid CMOS/memristor circuits, according to the authors' best knowledge, there is no existing work which describes

integration of digital leaky integrate-and-fire (LIF) neurons with CMOS/memristive synaptic crossbar arrays. The particular goal of this work is to integrate memristor-based synaptic crossbar arrays into a digital CMOS architecture. Different from the earlier work on analog-based neuromorphic systems, we aim to leverage the continuing scaling of digital CMOS devices for scalable realization of large spiking neural networks in modern CMOS technologies. In particular, we present a reconfigurable digital neuromorphic processor (DNP) architecture comprising a memristive crossbar, an array of digital LIF spiking neurons and on-line learning circuits that support STDP learning rules. The reconfigurability of our digital architecture renders it a flexible neuromorphic platform onto which a range of cognitive computing applications may be mapped.

A preliminary version of this work has been presented in Y. Kim et al. [2012]. In this article, we improve our neuromorphic circuits and architecture by addressing several key design issues. To implement a multilevel memristor synaptic crossbar, we systematically analyze the memristor device in terms of its programming time and discretization of memory storage. We also investigate memristor readout schemes to more efficiently perform digital LIF operations and present a low-cost digital pulse width modulation (PWM) scheme for writing the memristive crossbar. While our prior voltage-controlled-oscillator (VCO)-based column analog-to-digital converter (ADC) in Y. Kim et al. [2012] consumes a great deal of power, we address this limitation by introducing asynchronous counters to measure the VCO oscillation frequency in digital form. The proposed column ADC effectively accumulates the presynaptic weights of each neuron and allows a single adder and comparator to be shared among all N neurons to perform LIF operations without degrading throughput. This leads to considerable silicon area reduction. The digital implementation style of the ADC is also amenable to integration. In the proposed DNP, the $N \times N$ memristive synaptic array, which stores both multi-bit synapse values and network configuration data, is designed to be accessible both column- and row-wise to speed up the synaptic weight update process.

When implemented in a commercial 90-nm CMOS technology, a 256-neuron design with a 256×256 synaptic array based on the proposed neuromorphic architecture has an estimated area of 1.86 mm^2 and power consumption of 6.45 mW under a regular supply voltage of 1.2 V, respectively. The proposed memristor cell is $12.8\times$ more area efficient than the transposable 8T SRAM cell presented in Seo et al. [2011]. Our 256×256 memristive crossbar array occupies only 8.6% of the total chip area while supporting 3-bit synapses and storage of network configuration. Additionally, the VCO based column ADC with asynchronous counters reduces the power consumption by 24.8% when compared to the synchronous-counter-based ADC at the cost of only 1.6% area overhead.

The proposed neuromorphic architecture is rather flexible and can be configured to various network topologies to support a range of cognitive learning applications. To demonstrate its potential application, we configure our DNP to realize a two-layer spiking network with over two hundred silicon neurons for character recognition with unsupervised learning. The network is successfully trained to recognize all given letters. Furthermore, the impacts of process variations and noise on learning performance is analyzed for the proposed neuromorphic processor, which reveals the generally good robustness of the realized neuromorphic system.

2. DIGITAL NEUROMORPHIC PROCESSOR ARCHITECTURE

2.1. Overall Processor Architecture

Figure 1 depicts the overall block diagram of the proposed DNP architecture for an N-neuron network. It consists of a synapse unit, a learning unit with a global timer, a

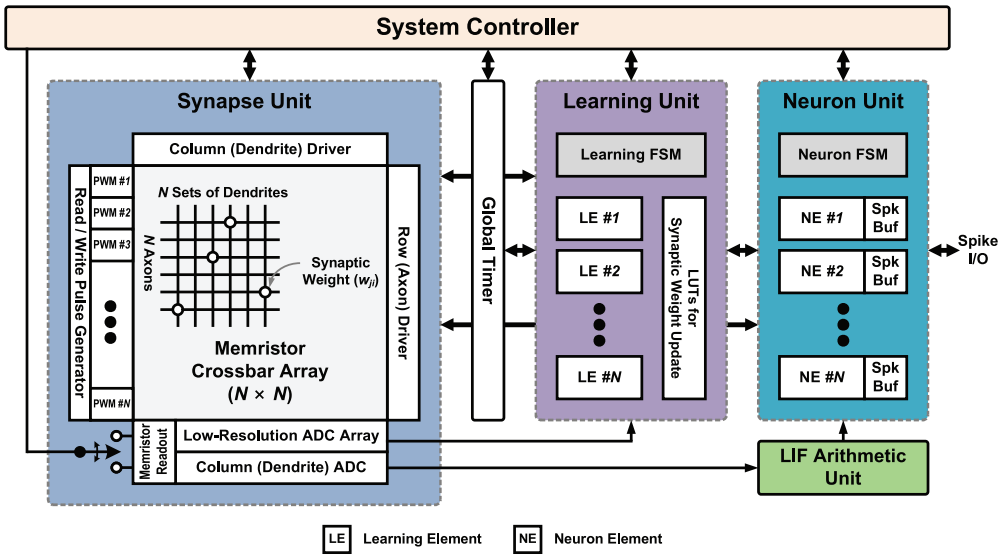

Fig. 1. Block diagram of the proposed digital neuromorphic processor architecture.

neuron unit, a LIF arithmetic unit and a system controller. The synapse unit consists of the proposed $N \times N$ memristive crossbar array, a column ADC, a flash ADC array, a read/write pulse generator and other interface circuits. The crossbar can represent a fully recurrent network topology and store N^2 all possible synaptic weights among the N neurons.

A biological neuron has multiple dendrites and a single axon. It receives input spikes from its presynaptic neurons and transmits output spikes to its postsynaptic neurons. The axon may connect with the dendrites of multiple postsynaptic neurons. In the crossbar array, a row and a column contain the axonal and dendritic connections, respectively, of a mimicked biological neuron. The connection between the jth row (axon) and ith column (dendrite) is represented by the synaptic weight w_{ji} between the jth and ith neurons. Each employed memristor device in the synaptic array keeps not only a multibit synapse value but also the network connectivity information. The proposed crossbar is fully reconfigurable in the sense that network connectivity can be programmed for any N-neuron network. Since the full set of $N \times N$ connections may not be necessary for a targeted network topology, we utilize the lowest storage level of each memristor to represent the absence of connection between the corresponding pair of the neurons. The detailed memristor cell utilization is presented in Section 3.

We adopt the LIF model to realize the dynamics of silicon neurons. The LIF model has been shown to be effective for a number of learning applications. It is suitable for digital implementation due to its moderate hardware overhead, that is, it can be realized using a few arithmetic components including an adder and a comparator [Indiveri et al. 2011]. The neuron unit, which consists of a finite state machine (FSM) and N neuron elements, emulates the LIF neuron dynamics while interfacing with the column (dendrite) ADC and LIF arithmetic unit. The proposed DNP has two different memristor readout circuits: a low-resolution ADC array and a column ADC, as described in Section 4.2. Each neuron element tracks the membrane potential and has spike buffers to store both the external input spikes that are fed by the off-chip environment and each output spike that is generated when its membrane potential reaches a given threshold. Each neuron element can be configured to be either excitatory or inhibitory, which potentiates or

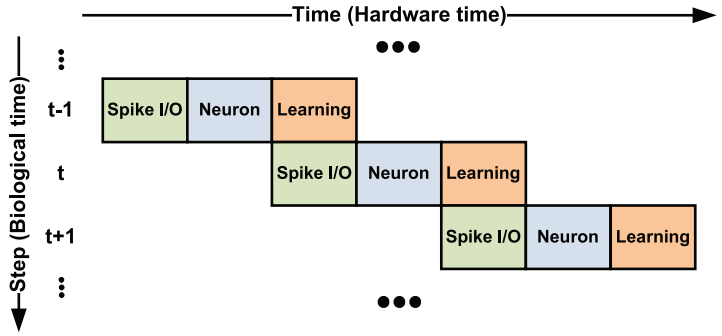

Fig. 2. Flow diagram of the proposed neuromorphic processor.

depresses the membrane potential of its postsynaptic neurons, respectively. In addition, each neuron can be configured as either an input, an output or an internal neuron of the network.

The learning unit is responsible for performing on-chip learning. It contains N learning elements that interface with the corresponding neuron elements and an FSM to control the overall synaptic weight update process. Each learning element has a register to maintain the corresponding neuron's spike timing, which is used to calculate the spike time difference between a presynaptic and a postsynaptic neurons. The learning unit updates the synapse values in the crossbar based on spike time difference to realize a given STDP learning rule. The STDP rule is programmable through the use of look-up tables (LUTs) where synaptic weight change as a function of timing difference is stored. These LUTs are shared by all N learning elements, thereby reducing silicon area significantly. Our design allows for parallel STDP updates of synaptic weights.

The communication between the proposed neuromorphic processor and the external environment is performed through the use of input and output spikes. External stimuli are applied in the form of input spikes while the output response of the network is sent off the chip in the form of output neuron spikes. In a character recognition system, for example, input letters are encoded into sequences of input spikes that are applied to the input neurons. The recognition (classification) result is identified from the spikes of the output neurons that are outputted.

2.2. Flow Control of the Neuromorphic Processor

The system controller manages the overall operations of the processor through clocking-based synchronous control as shown in Figure 2. Each step corresponds to a biological time unit and consumes many hardware clock cycles. The membrane potential of every neuron and the corresponding synaptic weights are updated per each biological step. The processing flow consists of three stages named as: (1) spike input/output (I/O), (2) neuron, and (3) learning. These stages are executed in a pipelined manner. The spike I/O and learning stages are processed simultaneously because there are no data and control hazards between them.

During the spike I/O stage, the spikes from the external environment are read into the input spike buffers of input neuron elements. Meanwhile, the spikes of the output neurons are read off the chip to observe as the output of the network. After receiving/transmitting all the input/output spikes, the neuron stage starts, where the membrane potential of each neuron is updated by the LIF arithmetic unit according to the governing neuronal dynamics. The neuron whose membrane potential exceeds the given threshold voltage produces a spike. These spikes are read off the DNP as the output during the spike I/O stage. After the neuron stage, the processing continues

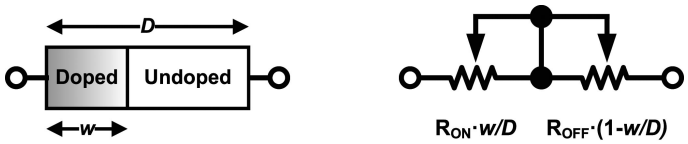

Fig. 3. Memristive device structure (left) and variable resistance model (right) [Strukov et al. 2008].

onto the third processing stage (i.e., the learning stage), where the synaptic weights are updated according to the STDP learning rule. In this rule, the time difference between a presynaptic and a postsynaptic spike events is measured to determine the synaptic weight change. The synaptic array access schemes for updating synaptic weights and membrane potentials are detailed in Section 3.

2.3. Neuron Dynamics of the Neuromorphic Processor

Each silicon neuron of the proposed processor implements the following LIF neuron dynamics

$$V_i[t] = V_i[t-1] + K_{SYN} \sum_{j=1}^{M} w_{ji} S_j[t-1] + K_{EXT} E_i[t-1] - V_{LEAK}, \tag{1}$$

where V_i is the membrane potential of the ith neuron, M is the number of pre-synaptic neurons, K_{SYN} is the synaptic weight parameter, w_{ji} is the synaptic weight between the jth and ith neurons, S_j is the activity bit that indicates whether the jth neuron fired, K_{EXT} is the external input spike parameter, E_i is the activity bit for the input spike of the ith neuron, and V_{LEAK} is the leaky potential.

If the membrane potential exceeds the given threshold voltage, the neuron element generates a spike event and its membrane potential is reset to the resting potential. The spiking activity bit of the ith neuron S_i is set according to

$$S_i[t] = \begin{cases} 1 & \text{if } V_i[t] > V_{TH} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where V_{TH} is the threshold voltage.

The detailed hardware realization of the neuronal dynamics is discussed in Section 4.3.

3. MEMRISTIVE SYNAPTIC CROSSBAR ARRAY

In this section, we first briefly introduce the memristor model and two readout schemes that are suitable for processing LIF operations of silicon neurons and synaptic weight update process. Then, the proposed memristive synaptic crossbar array and CMOS/memristor hybrid cell are presented. Additionally, we propose a new digital PWM scheme for both reading and updating memristive synapses. While an analog PWM scheme has been conceptualized for implementing the STDP earning rule [Snider 2008], the presented digital design is more amenable to integration into a digital system architecture.

3.1. Memristor Model

A memristor is a two-terminal electronic device. Conceptually, it has two divided regions: a doped and an undoped ones as shown in Figure 3. The memristor device model can be mathematically expressed by Ho et al. [2011]

$$R(x) = x \cdot R_{ON} + (1-x) \cdot R_{OFF}, \quad where \;\; 0 \le x \le 1. \tag{3}$$

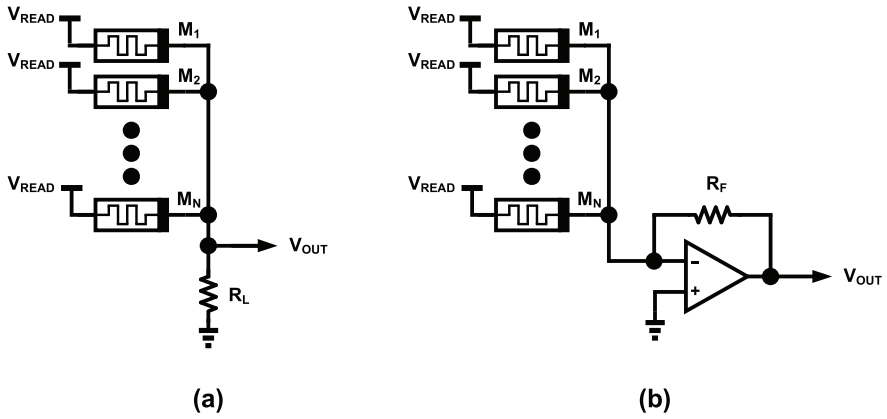

Fig. 4. Memristor sensing schemes by (a) load resistor and (b) summing amplifier.

In (3), R_{ON} and R_{OFF} are the fully doped (lowest) and fully undoped (highest) resistances of the memristor, respectively, and x is the internal state variable defined by the proportion of the memristive device length D that is doped (i.e., $x = w/D$).

The memristor's internal state x varies dynamically with the external input. A recent experimental study has shown that the conductance of a memristive device can be incrementally adjusted by altering the pulse width of its constant input voltage [Jo et al. 2010]. In other words, a longer positive pulse duration leads to a larger increase of memductance. Alternatively, the memristor state can be also changed by modulating the amplitude of the applied voltage input [Ho et al. 2011]. However, realizing pulse amplitude modulation (PAM) requires programmable analog circuits which are complex to integrate into a digital system. Hence, we realize the PWM scheme using digital logic and include it as part of our digital architecture. In this work, we model the memristors with parameters $R_{ON} = 10\text{K}\Omega$ and $R_{OFF} = 500\text{K}\Omega$. And the read voltage V_{READ} and write voltage V_{WRITE} are considered to be 1.2 V [Xu et al. 2011].

3.2. Memristor Readout Schemes

Two different ways to read the memristor internal state have been proposed by using either a load resistor or a summing amplifier (i.e., current-to-voltage converter), as depicted in Figure 4. The load-resistor-based sensing scheme is commonly adopted for both binary and multilevel memristor memories due to the ease of implementation [Ho et al. 2011; Merkel et al. 2011; Manem et al. 2012]. This scheme leverages a load resistor R_L, which is connected in series with the memristor, to form a voltage divider. Hence, the output voltage V_{OUT} of Figure 4(a) under a given read voltage V_{READ} is given by

$$V_{OUT} = \frac{\sum_{i=1}^{N}(1/R_i)}{(1/R_L) + \sum_{i=1}^{N}(1/R_i)} V_{READ} = \frac{\sum_{i=1}^{N} G_i}{G_L + \sum_{i=1}^{N} G_i} V_{READ}, \tag{4}$$

where N is the number of memristors attached to the sensing node, R_i and G_i are the resistance and conductance of memristor M_i, and R_L and G_L are those of the load resistor, respectively. This scheme is unable to integrate multiple memristor internal states in one step since the output voltage V_{OUT} in this equation does not represent a linear summation of either memristor resistances or conductances under a fixed read voltage V_{READ}. To update the membrane potential of a neuron based on a dynamical neuron model, what is possible with this scheme, though, is to accumulate all presynaptic

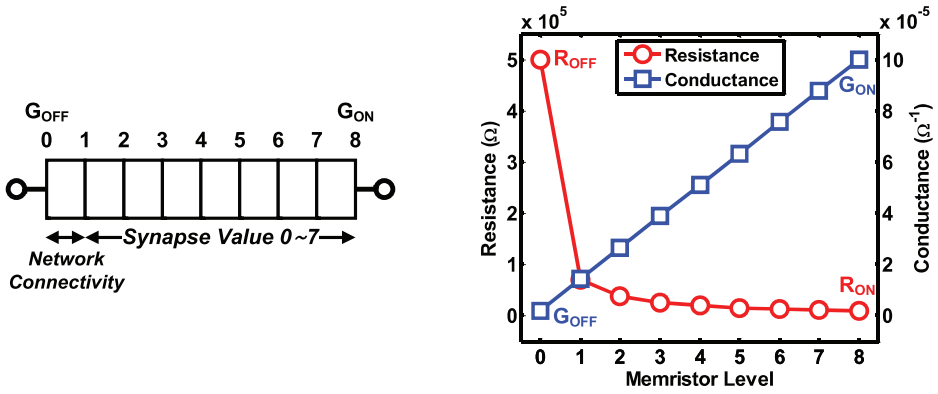

Fig. 5. Uniform partitioning of conductance (memductance) for multilevel synaptic weight storage.

Table I. Normalized Write Times to Change Memristor Conductance by One Level ($R_{ON} = 10K\Omega$, $R_{OFF} = 500K\Omega$, $V_{WRITE} = 1.2V$)

Level change	$0 \Leftrightarrow 1$	$1 \Leftrightarrow 2$	$2 \Leftrightarrow 3$	$3 \Leftrightarrow 4$	$4 \Leftrightarrow 5$	$5 \Leftrightarrow 6$	$6 \Leftrightarrow 7$	$7 \Leftrightarrow 8$
Time	8205	117	25	10	5	3	2	1

weights of the neuron, which are stored in N memristors, by spending N iterations with an additional adder. In other words, one memristor state can be readout and accumulated by the adder at a time, requiring a total of N^2 iterations to complete all N neurons' LIF tasks. Alternatively, to concurrently integrate the pre-synaptic weights of N neurons requires N adders and N iterations [Seo et al. 2011].

On the other hand, the summing-amplifier-based sensing scheme provides a linear summation of conductances of memristors such that it is possible to integrate all pre-synaptic weights of each neuron at once [Y. Kim et al. 2012]. This scheme forms a virtual ground at the negative input terminal of the amplifier. The current from each memristor flows into the virtual ground. Thus, the output voltage V_{OUT} of Figure 4(b) with the input voltage V_{READ} is given by

$$V_{OUT} = R_F \sum_{i=1}^{N} \frac{V_{READ}}{R_i} = R_F \sum_{i=1}^{N} G_i V_{READ}, \tag{5}$$

where R_F is the feedback resistor of the amplifier.

Note that both schemes can be employed to sense the state of a single memristor while the amplifier based scheme is more efficient for accumulating the internal states of multiple memristors. Hence, we leverage the summing amplifier based sensing scheme to accumulate all presynaptic weights of each neuron for LIF operations, whereas the resistor-based sensing is exploited to detect each memristor's current state in the synaptic weight update process.

3.3. Multilevel Memristive Synaptic Storage

Since the summing amplifier based sensing provides a linear summation of conductances of memristors, each memristor is designed to have uniformly spaced conductance (memductance) levels to represent a multilevel synaptic weight as shown in Figure 5. However, what is important to note is that the programming (writing) time required to perform a synaptic update has a strong dependency on the memristor current internal state [Manem et al. 2012]. Table I shows the write times needed to change the conductance value by one level under different internal states according to our adopted

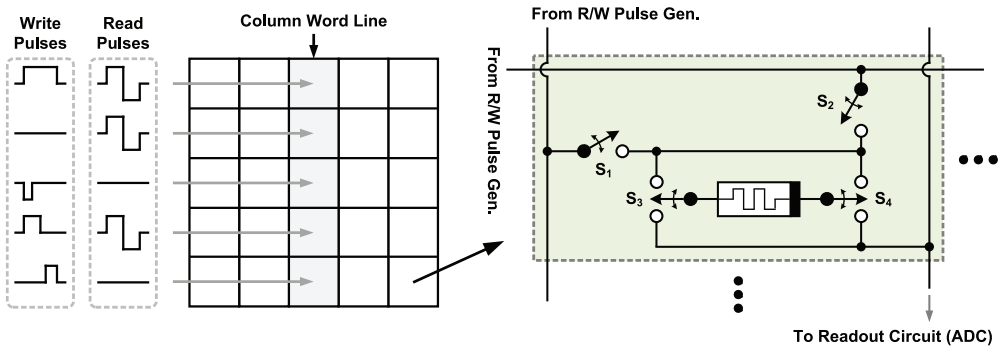

Fig. 6. Proposed synaptic crossbar array and hybrid CMOS/memristor synaptic cell.

memristor model [Xu et al. 2011]. These write times are normalized with respect to the time needed to move the memristor state from level 7 to level 8. Note that the write times are symmetric with respect to the direction (e.g., write time for changing from level 1 to level 0 and vice-versa are the same). The programming time required to move from level 0 to level 1 is over 8000× longer than that from level 7 to level 8. The excessive programming time required to change the state between levels 0 and 1 may notably slow down the overall on-chip learning speed during the training phase. To minimize performance delay by avoiding long writing times, we utilize the lowest level of the memristor state to store the network connectivity information and other higher levels to represent the actual synapse value (i.e., 3-bit synapse) if there exists a synaptic connection. As an example, if the memristor in the ith column and the jth row of the crossbar is at level 0, there is essentially no connection between the jth and the ith neurons. On the other hand, memristor conductance (memductance) level 4 indicates that the jth and ith neurons are connected with a synaptic weight of 3.

3.4. Memristive Crossbar Array and Cell Design

Figure 6 exhibits the proposed synaptic crossbar array and the CMOS/memristor hybrid synaptic cell. The two switches S_1 and S_2 in the cell are introduced to allow each memristor to be accessible in both the column and row fashions. When the row (column) driver activates a word line, S_1 (S_2) switches of all cells that lie in the same row (column) are turned on and ready to be accessed. Parallel voltage pulses are generated by the read/write (R/W) pulse generator and applied to read or write all cells in the row (column) as shown in Figure 6. To read from a cell, a fixed positive voltage pulse is applied to the memristor cell when S_3 and S_4 are connected to the pulse generator and the ADC (memristor readout circuit) lines, respectively. The current generated by the cell due to the applied positive voltage pulse flows out and goes into the ADC line and is converted to a digital value, reflecting the memductance of the cell. Unfortunately, the applied positive pulse disturbs each memductance [Ho et al. 2011]. Therefore, a flipped (i.e., negative) voltage pulse following the positive one is injected to each memristor to restore its memductance, resulting zero net flux injection for the memristor. This is effectively done by connecting S_3 and S_4 to the ADC and the generator lines, respectively.

In the write operation, the cells in either one row or column are accessed and updated in parallel. A write voltage pulse is injected to each memristor cell and its memductance is altered depending on the pulse duration. The write operation latency varies with respect to the value to be written into the cell. It is possible to either increase or decrease the memductance. For the latter, S_3 and S_4 are connected to the ADC and

the generator lines, respectively, to effectively apply a negative voltage pulse to the memristor cell.

3.5. Memristive Synaptic Crossbar Array Accesses

The synaptic crossbar array is accessed during both the neuron and learning stages. At each hardware time step in the neuron stage, through the column driver, a neuron element activates the corresponding column word line to access all its presynaptic weights. A R/W pulse generator, which contains N digital PWMs and is detailed in Section 4.1, produces parallel pulses for reading all the presynaptic weight values from the memristor cells in the corresponding column of the crossbar. These values are sent to the column ADC. The ADC accumulates these presynaptic weights and converts the sum into a digital quantity. Finally, the neuron element updates its membrane potential by adding up the accumulated presynaptic weights, the weighted external spike input and the leaky potential through the LIF arithmetic unit.

For the learning stage, the synapse values are updated according to the STDP learning rule. To do this, each learning element has a time register to keep track of the neuron's spike event time that is stamped by the global timer. For each fired neuron, the learning unit conducts a presynaptic and a postsynaptic weight updates in a row. If a neuron fires,

(1) all its pre- (post-) synaptic neurons' time registers are compared with the global timer and the corresponding learning elements determine the amounts of synaptic weight update according to the prestored STDP LUT;
(2) the column (row) driver activates the memristor crossbar array's column (row) word line that is associated with the dendrites (axon) of the fired neuron;
(3) the R/W pulse generater generates a read pulse word to the corresponding column (row) to sense each memristor's current internal state (i.e., current synaptic weight) through the low-resolution ADC array;
(4) based on the current memristors' states determined in step (3), the learning elements calculate the pulse durations needed to produce the desired synaptic weight changes as determined in step (1).
(5) all pre- (post-) synaptic weights of the neuron are updated by means of the R/W pulse generator with the durations determined in step (4).

Note that the R/W generator produces parallel write pulses that have different widths according to not only the amount of each synaptic weight change but also the memristor's current internal state due to the nonlinear memristor device write-time characteristics described in Section 3.3. This synaptic update process is triggered only when the corresponding neuron fired in the neuron stage. In other words, if the membrane potential of the ith neuron does not exceed the threshold voltage, then both the pre- and postsynaptic weight update processes for the ith neuron are skipped. The entire learning stage is omitted when no neuron has fired during the preceding neuron stage. The proposed architecture processes spiking I/O tasks and the learning stage simultaneously.

4. IMPLEMENTATIONS OF BUILDING BLOCKS

4.1. Digital Pulse Width Modulation for Memristive Synaptic Cells

As discussed in Section 3.5, the synaptic crossbar array is accessed during both the neuron and learning stages through the R/W pulse generator, which contains N digital PWMs. For instance, in the learning stage the proposed DNP uses a parallel write pulse word, consisting of N binary pulses whose durations may be different from each

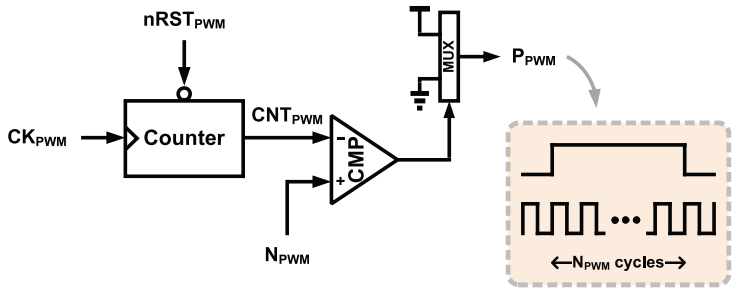

Fig. 7. Proposed digital pulse width modulator.

other, to update all the presynaptic and postsynaptic weights of each neuron that has fired.

Several PWM design considerations are worth discussing. The delay line based digital PWM requires a large number of delay cells to realize many different pulse widths and a large multiplexer to select one output from these cells, leading to remarkable area and power overheads [Syed et al. 2004]. Also, the delay cells are sensitive to PVT variations, introducing pulse width variability that may lead to failures in writing desired values to the memristors. Instead, we design a low-cost counter-based digital PWM to generate pulses with various durations as illustrated in Figure 7.

In this PWM design, the counter records the number of cycles of the clock signal CK_{PWM}. Its output CNT_{PWM} is compared with the desired number of cycles N_{PWM} by the digital comparator. The multiplexer outputs a "1" until CNT_{PWM} reaches N_{PWM} and after that it outputs a "0". The pulse duration is given by

$$t_{PWM} = N_{PWM} \cdot t_{CKPWM}, \tag{6}$$

where N_{PWM} and t_{CKPWM} are the desired number of cycles and the clock period, respectively, of the PWM clock CK_{PWM}. Note that CK_{PWM} does not have to be identical to the DNP operating clock. N_{PWM} is provided by the learning unit, where the amount of synaptic weight change is calculated by the time difference between a pre- and a postsynaptic firing events in accordance with the STDP rule in the learning stage. CK_{PWM} and N_{PWM} can be straightforwardly configured according to the range of synaptic weight change and device characteristics of the memristor. The R/W pulse generator includes N digital PWMs to simultaneously access the memristive synaptic cells in either one column or one row.

4.2. Memristor Readout

Figure 8 illustrates the proposed memristor readout block that includes a column ADC and a low-resolution ADC array. The column ADC works only in the neuron stage to conduct LIF operations while the low-resolution ADC array is activated during the learning stage to sense each memristor's internal state.

In Seo et al. [2011], each neuron circuit has its own adder and comparator to integrate all presynaptic weights and determine firing activity. It requires N iterations to complete all N neurons' LIF tasks. The proposed column ADC, which contains a summing amplifier, a sample-and-hold circuit and a high-resolution ADC, provides significant power and area reductions without degrading overall throughput. It allows a single adder and comparator (LIF arithmetic unit) to be shared by all N neurons. During a LIF operation, the R/W pulse generator injects a pulse word into the corresponding column to read all presynaptic weights from the memristor cells corresponding to the processed neuron. The current from each cell flows into the virtual ground and is

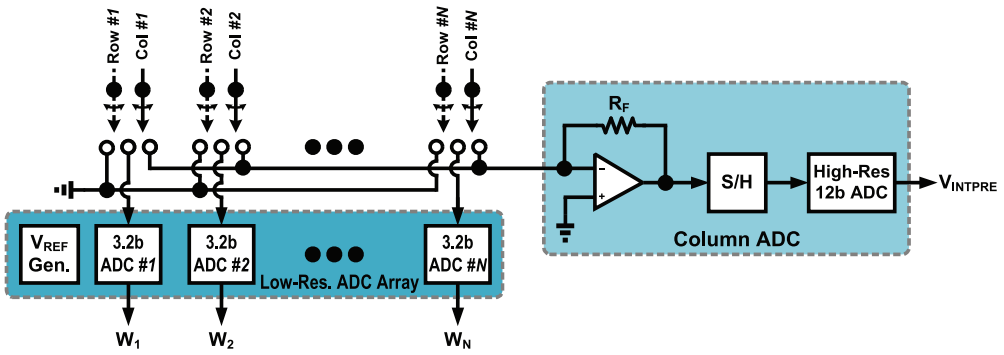

Fig. 8. Proposed memristor readout block consisting of a column ADC and a low-resolution ADC array.

summed at the negative terminal of the amplifier. The total current is converted to a voltage quantity through the feedback amplifier. The sample-and-hold circuit keeps the voltage and the high-resolution ADC transforms it into a digital value that corresponds to the term $\sum_{j=1}^{M} w_{ji} S_j[t-1]$ in (1). In this way, N-iterations are enough to complete the LIF operations for all N neurons with only one LIF arithmetic unit (Section 4.3).

It is important to determine the desired column ADC resolution, which can be found according to

$$resolution = \lceil log_2 N + log_2 L \rceil, \tag{7}$$

where N and L are the numbers of neurons and conductance levels of the memristor cells in the array, respectively. Obviously, the flash ADC architecture is not suitable for a high-resolution ADC because it requires $2^K - 1$ comparators to implement K-bit analog-to-digital conversion, leading to considerable power and area consumptions. The successive approximation register (SAR) and pipeline ADCs occupy large silicon area because of the employed area-consuming passive components. Moreover, the SAR and delta-sigma ($\Delta\Sigma$) ADCs are able to achieve a high-resolution but they unfortunately have a relatively slow conversion rate that is in the KHz range.

In principle, the ADC architecture and synaptic crossbar access styles shall be jointly optimized so as to optimally trade off between access speed, and power and area overheads. In the case of this work, we have found that a low-cost high-resolution ADC with a moderate conversion speed is a good choice. For this, we adopt a multiphase VCO-based ADC whose analog input alters the VCO frequency. The digital output of the ADC is produced by measuring the VCO frequency using counters [Yoon et al. 2008].

The VCO-based ADC is readily implemented with a few digital components such as counters, resulting in a small area overhead. Figure 9 shows the block diagram of the VCO based ADC, which consists of a ring VCO, counters, and a tree adder, and delay cells. We employ a 12-stage ring VCO with pseudo differential delay cells that are based on an inverter structure with a NMOS current source. A back-to-back inverter pair is used in each delay cell to produce a differential output. The VCO operating frequency is adjusted by controlling the current source (i.e., a higher current leads to a higher frequency). The clock phases of different VCO stages can be exploited to enhance the ADC resolution [Kim et al. 2010]. Generally, the use of more phases results in a higher resolution with higher power and area overheads. In our DNP, six phases (see Figure 9) are employed to achieve a 12-bit ADC resolution for 256 neurons and 9 conductance levels of the memristor cells under 1 MHz conversion rate as described in Section 5. These clock phases are connected to the respective digital counters whose

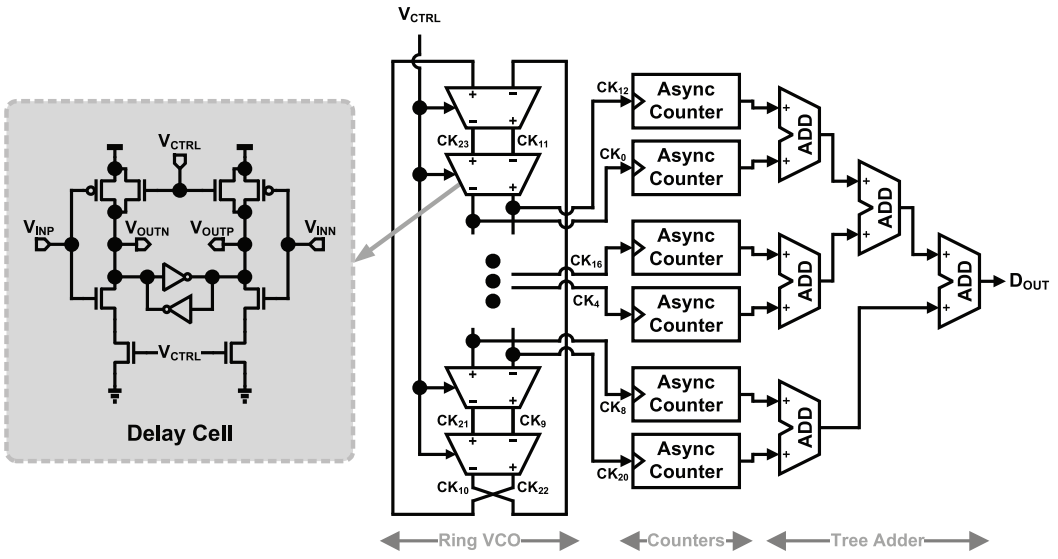

Fig. 9. Block diagram of the proposed VCO-based ADC and delay cell.

outputs are summed by the tree adder to obtain the final digital output. The oscillation frequency of our VCO is up to 1016 MHz. Thus, a 10-bit counter is used to measure the frequency of each clock phase with a 1 MHz sampling frequency. The VCO usually operates at a frequency of a few hundred MHz. Synchronous counters operating at such a high frequency can consume a considerable amount of power [Y. Kim et al. 2012]. Hence, we employ asynchronous counters to significantly reduce the ADC power.

The learning stage also require AD conversions. Before performing a synaptic weight update to set the weight to a specific value, the corresponding memristor's current internal state must be read out to determine the right pulse duration for writing the memristor cell. This is because that the required pulse width varies with respect to the current memristor state due to the nonlinear device characteristics as demonstrated in Table I. Note that here a low-resolution AD conversion is sufficient since each synaptic value needs to be read out individually. To do this, we employ an array of N low-resolution flash ADCs to read all pre- (post-) synaptic weights of each column (row) in parallel. Also, we adopt the load resistor based sensing scheme in Figure 4(a) to avoid using multiple amplifiers and thereby reduce area and power dissipation. Each flash ADC has eight comparators to detect each of nine internal state levels of the memristor cell and a digital logic circuit for encoding the output of the comparators. The reference voltage generator shared by the N flash ADCs is a resistor string that creates eight reference voltages for the comparators of each ADC. Importantly, this string does not have equally spaced resistor values since the state of memristor cells is equally sliced not by resistance but by conductance (see Figure 5).

4.3. Neuron and LIF Arithmetic Units

As described in Section 3.2, we employ the summing-amplifier-based sensing scheme to accumulate all presynaptic weights of each neuron for LIF operations. In conjunction with this memristive array access scheme, the neuron unit interfaces with a LIF arithmetic unit to emulate the LIF neuron dynamics of (1) for all neurons during the neuron stage. The block diagram of neuron elements with the LIF arithmetic unit and the processing flow of the neuron unit are described in Figure 10. For each neuron, the

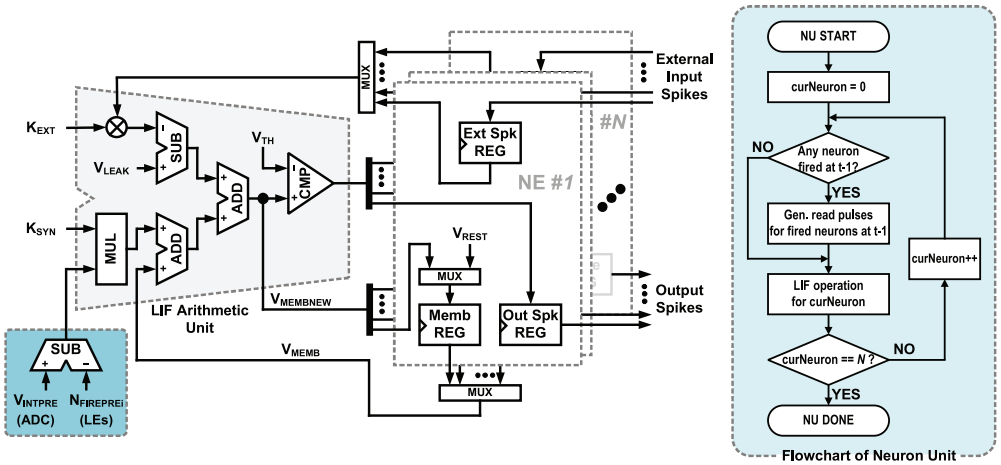

Fig. 10. Neuron elements with a LIF arithmetic unit (left) and processing flow of neuron unit (right).

neuron unit makes the R/W pulse generator produce a read pulse word for the corresponding column (dendrite) of the crossbar and sums up all the presynaptic weights of the neuron (V_{INTPRE}) through the column ADC. Importantly, only the synaptic weights associated with the presynaptic neurons that fired at the previous time step $t-1$ are read to save power. Consider a network of 10 neurons as an example. If the 3rd, 4th, and 5th neurons fired and the other 7 neurons did not fire at $t-1$, only the 3rd, 4th, and 5th PWMs in the R/W pulse generator produce a read pulse and the other PWMs output a "0". Similarly, this entire process of column reading and presynaptic weight accumulation is skipped when there was no firing activity across the network at time step $t-1$, that is, $\forall j, 0 \le j < N : S_j[t-1] = 0$ in (1). In this case, V_{INTPRE} is forced to zero due to the term $\sum_{j=1}^{M} w_{ji} S_j[t-1] = 0$ in (1) and only the leaky potential V_{LEAK} and the weighted external input (the K_{EXT} term) are considered to update the membrane potential.

Importantly, the column ADC output V_{INTPRE} should be adjusted since our synaptic cells also store the network connectivity information (e.g., the memristor conductance level of 4 in the jth row and ith column of the crossbar indicates a synaptic weight value of 3 between the jth and ith neurons). Therefore, the number of fired presynaptic neurons $N_{FIREPREi}$ for the ith neuron is subtracted from V_{INTPRE}. The former is evaluated by the corresponding learning element during the synaptic weight update process (i.e., the learning stage) at the previous time step $t-1$. The subtractor output $V_{INTPRE} - N_{FIREPREi}$ is added to the corresponding membrane potential V_{MEMB}, and the weighted external input spike and leaky potential V_{LEAK} are added as well. The adder's output $V_{MEMBNEW}$ is compared to the threshold voltage V_{TH} by the digital comparator and the result is sent to the respective neuron element through the demultiplexer and captured by its output spike register. Meanwhile, $V_{MEMBNEW}$ is sent to the neuron element and stored in the membrane register if it does not exceed V_{TH}. Otherwise, the register is reset to the resting potential V_{REST} via the multiplexer.

4.4. Learning Unit

The learning unit is designed to perform on-chip STDP learning by calculating the amounts of pre- and postsynaptic weight changes, determining the pulse durations to write the desired weights, and updating the memristive crossbar array through the R/W pulse generator.

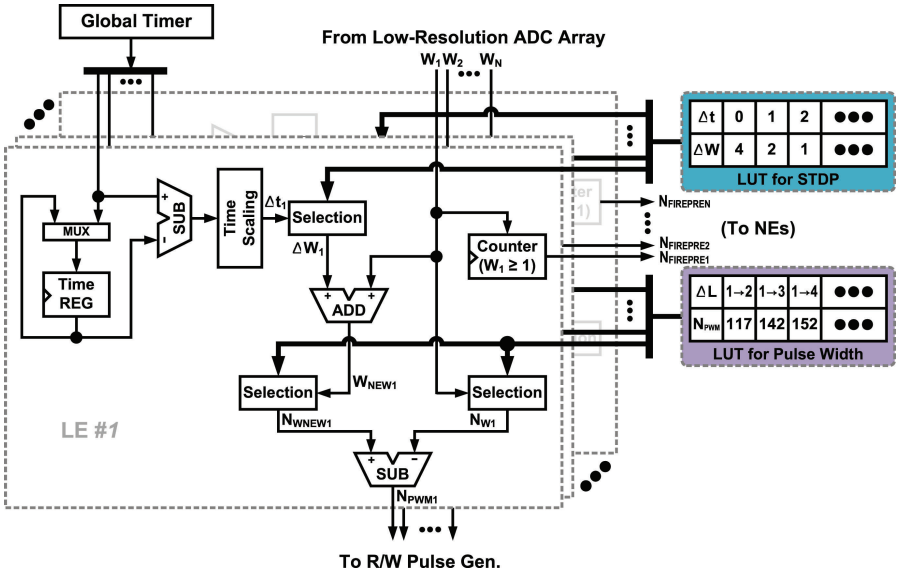

Fig. 11. Learning elements with shared LUTs.

Figure 11 exhibits the block diagram of the learning elements with two programmable LUTs. All learning elements contain their own time register to keep track of the corresponding neuron's spike event time and share two register-based LUTs, one for storing the STDP learning curve and the other write pulse widths for the memristors. The STDP LUT holds several pairs of spike time difference Δt and synaptic weight change ΔW while the other LUT stores the required number of cycles of PWM clock CK_{PWM} to write each desired value to a memristor.

The design of the second LUT involves important area considerations. For K-bit synapses, a brute-force implementation would require a large number of $\frac{2^K(2^K-1)}{2}$ LUT entries for all possible pairs of current and target memristor conductance levels. Instead, to reduce area and complexity of the selection logic, we design the LUT in such a way that only the numbers of CK_{PWM} cycles required to alter the memristor state from the lowest level to each target (i.e., from level 1 to levels 2, 3, 4, 5, 6, 7, and 8) are stored using $2^K - 1$ entries. With this area-efficient design, the actual pulse duration for a given update is determined by finding the difference between the stored numbers of cycles of the current and target memristor levels. For instance, the number of cycles required to increase the memristor level from 3 to 5 is obtained by subtracting the number of cycles needed for changing the memristor level from 1 to 3 from that for changing the level from 1 to 5. As a result, we attain $2^{K-1}\times$ area reduction (e.g., $4\times$ reduction in the proposed DNP). In addition, the learning unit supports a time scaling feature to provide additional programmability for the stored STDP rule. It is implemented with a shift operation of the time differences.

The processing of the learning stage for the entire network is performed as follows. The synaptic weight updates are processed by iterating over all fired neurons. To check each neuron's firing activity, the learning unit checks the output spike buffer of the corresponding neuron element, which is filled in the neuron stage. For each fired neuron, the learning unit runs the following two back-to-back parallel processes, one for presynaptic weight updates and the other postsynaptic weight updates.

The respective learning element for every fired neuron updates its time register with the global timer. Simultaneously, all learning elements calculate the scaled time differences $\Delta t_1, \Delta t_2, \ldots, \Delta t_N$ between the global timer and their time register values. This step basically determines the firing time differences between this fired neuron and all other neurons in the network. The synaptic weight changes $\Delta W_1, \Delta W_2, \ldots, \Delta W_N$ for $\Delta t_1, \Delta t_2, \ldots, \Delta t_N$ are selected from the STDP LUT in parallel. Meanwhile, all pre- (post-) synaptic weights $W_1, W_2, \ldots, W_N$ (before update) are read from the corresponding column (row) in the synaptic array through the R/W pulse generator and the low-resolution ADC array, and are finally fed into the respective learning elements, also in parallel.

The pre- (post-) synaptic weights to be written into the respective memristor cells $W_{NEW1}, W_{NEW2}, \ldots, W_{NEWN}$ are computed by the adder (i.e., $\forall i : W_{NEWi} = W_i + \Delta W_i$). W_i and W_{NEWi} correspond to the current and target levels of the ith memristor, respectively. Then, each learning element concurrently looks up the cycle counts N_{Wi} and N_{WNEWi} from the entries associated with W_i and W_{NEWi} from the pulse width LUT. The numbers of cycles $N_{PWM1}, N_{PWM2}, \ldots, N_{PWMN}$ required to update the pre- (post-) synaptic weight values to $W_{NEW1}, W_{NEW2}, \ldots, W_{NEWN}$, respectively, are determined by subtracting each N_{Wi} from N_{WNEWi} (i.e., $\forall i : N_{PWMi} = N_{WNEWi} - N_{Wi}$).

Finally, the pulse generator produces a parallel write pulse word with $N_{PWM1}, N_{PWM2}, \ldots, N_{PWMN}$ as in Figure 7. When creating the word, N_{PWMi} is set to "0" for zero-valued synaptic wights (i.e., $W_i = 0$) since no pre- (post-) synaptic connection exists between these neuron pairs. Additionally, for negative N_{PWMi} values, the generator inverts the polarity of the pulses, which is effectively done by manipulating the switches of the memristor cell as shown in Figure 6. Also, all learning elements record the numbers of fired pre-synaptic neurons $N_{FIREPRE1}, N_{FIREPRE2}, \ldots, N_{FIREPREN}$ during the postsynaptic weight update process as needed for the following LIF operations as detailed in Section 4.3. Note that the entire learning stage is skipped when there is no firing activity across the network.

5. SIMULATION RESULTS

To demonstrate the application of the proposed digital neuromorphic processor architecture, a DNP with 256 silicon neurons, learning circuits and 64K synapses has been implemented. Except for the memristor nanodevices, the proposed neuromorphic processor is designed using a commercial 90-nm CMOS technology under a regular supply voltage of 1.2 V. The digital blocks of the DNP are synthesized with a commercial standard cell library using Synopsys Design Compiler under 1-MHz main clock and 50-MHz PWM clock frequencies. The analog components including the column ADC (e.g., the ring VCO and summing amplifier) and low-resolution ADC array are custom designed. The memristors used in the crossbar array are modeled with the parameters in Xu et al. [2011]. Cadence Virtuoso and SOC Encounter are used for laying out the analog and digital blocks, respectively. Cadence Virtuoso is used for final layout integration to create the whole chip layout. Synopsys VCS and HSPICE are used to simulate the digital and analog circuits, respectively. The results presented in this section are based on prelayout simulation.

5.1. Column ADC Performance

To show the performance of the VCO-based ADC as proposed in Figure 9, we sweep the input voltage from 0.45 V to 1.15 V with a 0.05 V step under a sampling frequency of 1 MHz. The simulated input-to-output characteristic is shown in Figure 12(a). The digital output spreads over a range between 1,440 and 5,862 (i.e., 12.1-bit resolution) and exhibits excellent linearity with respect to the input ($R^2 = 0.9964$). Thus, it satisfies

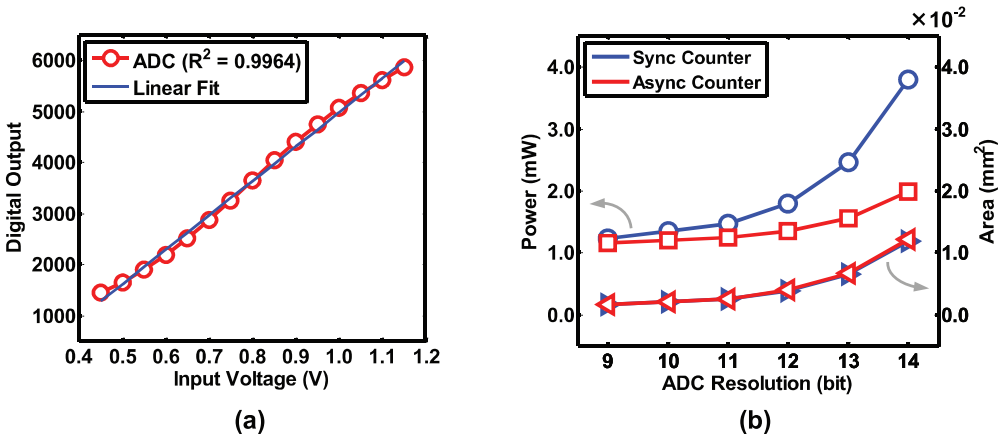

Fig. 12. Column ADC performance: (a) input-to-output characteristics, and (b) power and area as functions of counter type and resolution.

the resolution required in (7) to serve as a column ADC for the 256 silicon neurons and 256×256 memristive crossbar array with 3-bit synapses in the proposed architecture.

In Figure 12(b), we compare the power and area of the ADCs with asynchronous and synchronous counters under various ADC resolutions. Up to 24 counters can be attached to our 12-stage differential ring VCO in Figure 9, producing a ADC resolution between 9- and 14-bits. Note that the tree adder size also varies according to the number of counters. The ADC power consumptions are measured with an input voltage of 0.8 V, which is the mean input level of Figure 12(a). It can be observed from Figure 12(b) that the ADC with asynchronous counters is more power efficient than the one with synchronous counters while the areas of the two designs are almost the same. With a 12-bit resolution, the ADC adopting asynchronous counters dissipates 24.8% less power with only a 1.6% area increase compared with the synchronous-counter-based ADC, demonstrating the appealing low-power advantage of the former.

5.2. Overall Processor Performance

Figure 13(a) shows the layout of the proposed DNP with 256 neurons and 64 K synapses. The chip dimension is 1.45 mm×1.28 mm. In the chip layout, the memristor crossbar array is defined as an empty macro based on its estimated area. The area breakdown for the neuromorphic processor is shown in Figure 13(b). The synapse unit, which includes the column ADC, low-resolution ADC array, memristive crossbar, and pulse generator, occupies about 40% of the chip area. Despite the relatively small area of the memristive crossbar array (8.6%), realizing the parallel access scheme for updating the multibit memristive crossbar in the learning stage requires integration of several peripherals such as the array of low-resolution ADCs and multiple PWMs. As a result, the learning unit occupies a large portion (42.2%) of the chip area. Nevertheless, as a return, this parallel scheme expedites the synaptic weight update process significantly.

Figure 14 demonstrates the overall performance of the neuromorphic processor. The power consumption of the processor as a function of network size, which is evaluated based on the 90-nm CMOS technology and memristor parameters in Xu et al. [2011], is depicted in Figure 14(a). The required column ADC resolution is a function of network size as given in (7) (e.g., 9-bit column ADC for a 32-neuron design). For the range of network size considered, doubling the number of integrated neurons N increases the chip power by less than twice. The asynchronous-counter-based column ADC consumes over 21% of the overall chip power dissipation but its power does not increase much

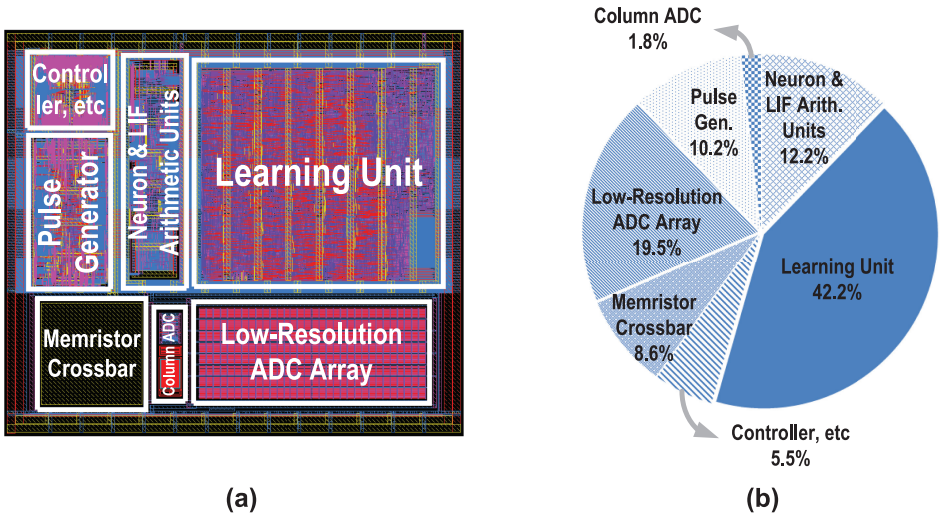

Fig. 13. Neuromorphic processor: (a) layout of the neuromorphic processor with 256 neurons and 65,536 synapse and (b) area breakdown.

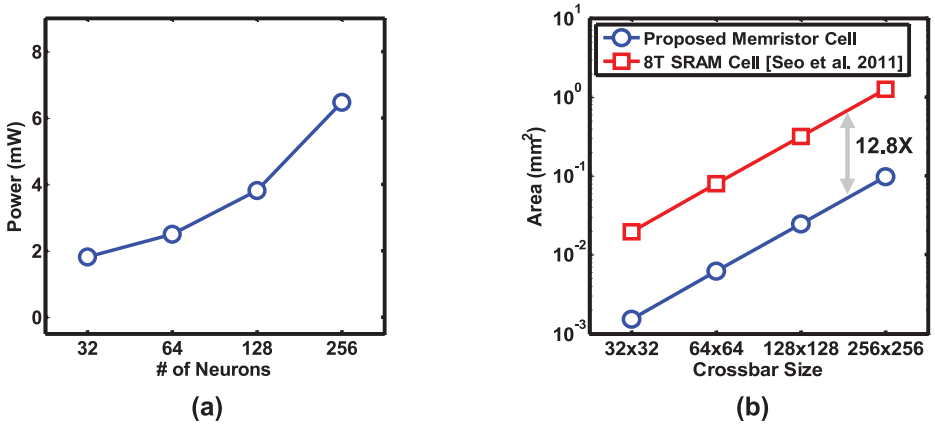

Fig. 14. Neuromorphic processor performance: (a) processor power as a function of network size, and (b) area comparison of the 8T-SRAM based synaptic crossbars [Seo et al. 2011] and the proposed memristive crossbars.

as the resolution increases (see Figure 12(b)). Additionally, for the 256-neuron design the power consumed by the synapse unit, which contains the memristive crossbar, column ADC, flash ADC array, R/W pulse generator and other interface circuits, reaches 5.16 mW or 80% of the entire processor power. The power consumed by the memristive crossbar is estimated by considering the average memristance and the supply voltage. Therefore, synaptic access schemes with a better power efficiency are of a particular interest, as will be explored in the future work.

Figure 14(b) compares the areas of the 8T-SRAM-based synaptic crossbars [Seo et al. 2011] with the proposed memristive crossbars. For a fair comparison, the areas of the 8T SRAM cell designs are scaled to the 90-nm technology node by considering the feature size difference. Note that the proposed memristive crossbars store 9 levels of synaptic weight values with the lowest level indicating network connectivity, leading to

Table II. Neuromorphic Processor Implementation Summary and Comparison

Item	[Seo et al. 2011]	[Merolla et al. 2011]	This work
Technology	45-nm SOI-CMOS	45-nm SOI-CMOS	90-nm CMOS
Synapse Storage	SRAM	SRAM	Memristor
Supply Voltage	$0.55 \sim 1.0$ V	$0.85 \sim 1.05$ V	1.2 V
Operating (PWM) Frequency	1 MHz	Event-Driven	1 (50) MHz
# of Neurons	256	256	256
# of Synapses	65,536	262,144	65,536
Synapse Resol.	1-bit	1-bit	3-bits (8-levels)
Neuron Model Param. Resol.	8-bits	8-bits	5-bits
Membrane Potential Resol.	8-bits	8-bits	16-bits
Neuron Model	Digital LIF neuron	Digital LIF neuron	Digital LIF neuron
Learning Rule	On-chip STDP	Off-chip learning	On-chip STDP
Synaptic Connection	Fully recurrent crossbar	Fully reconfigurable crossbar	Fully reconfigurable crossbar
Power Dissipation	N/A	N/A	6.45 mW
Area	0.8 mm^2	4.2 mm^2	1.86 mm^2

an effective resolution of 3.2-bits. The 8T SRAM crossbar designs only have a resolution of 3-bits (8 levels).

As shown in Figure 14(b), the proposed designs are 12.8× more area efficient than the reference designs for various crossbar sizes and can store more information (i.e., 3.2-bits vs 3-bits). Furthermore, our 256×256 memristive crossbar occupies only 8.6% of the total chip area (see Figure 13(b)) while supporting 3-bit synapses and storage of network configuration. This is in contrast with the reference SRAM based design that contributes approximately 13% of the chip area while supporting only 1-bit synapses [Seo et al. 2011]. Obviously, the percentage of area occupied by the SRAM crossbar would further increase as synapse resolution increases. Hence, the memristive crossbar is certainly appealing for implementing on-chip multibit synaptic weight storage.

Table II summarizes the key specifications of the our neuromorphic processor design. According to the best knowledge of the authors, no memristive synaptic-array-based digital spiking neural networks have been reported in the literature yet. Therefore, we list the two SRAM crossbar-array-based designs with digital LIF neurons in Table II for comparison. These two chips have been implemented with a 45-nm SOI-CMOS technology. They employ supply voltage scaling to reduce power consumption. All three designs have 256 silicon neurons adopting the digital LIF model. Among the two compared reference designs, the design of Merolla et al. [2011] employs an asynchronous design technique which is achieved by event-driven communication while the one of Seo et al. [2011] adopts the standard synchronous clocking scheme with the same operating frequency of 1 MHz. The design of Merolla et al. [2011] integrates 1024×256 binary synapses whose values are programmed based on off-chip learning. This design consumes the largest area with 4× more synapses than other two designs which have 256×256 synapses. In contrast to the work of Merolla et al. [2011], our design and the design of Seo et al. [2011] support on-chip STDP learning. The proposed DNP has a 3-bit synaptic resolution while the other two only support binary synapses. In contrast with the proposed DNP, one key potential limitation of the design of Merolla et al. [2011] is its lack of on-chip learning capability. Furthermore, the learning performance of both reference designs ([Seo et al. 2011] and [Merolla et al. 2011]) may be limited by their low (1-bit) synaptic resolution. To fairly compare the area, we scale down our chip area approximately to the 45-nm technology node since our DNP is implemented

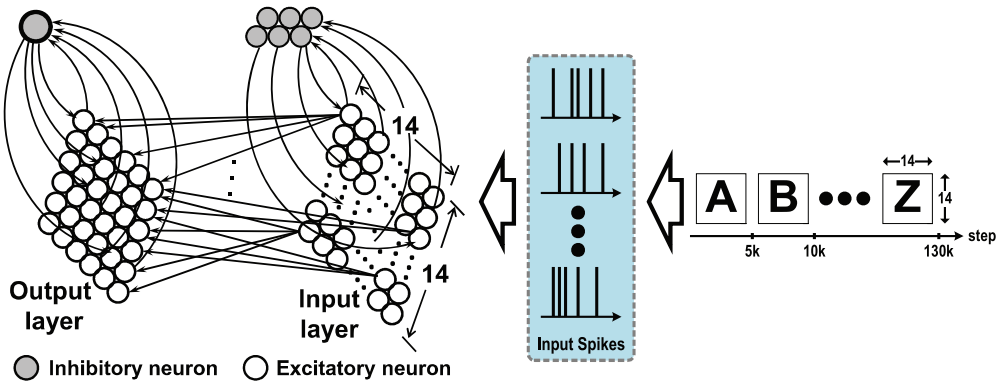

Fig. 15. Digital neuromorphic processor configured as a two-layer network for character recognition.

with a 90-nm technology. When the chip area is scaled with a factor of L^2 where L is the ratio of the feature sizes (i.e., 45 nm vs. 90 nm), our chip area would be 0.465 mm^2 at the 45-nm node, which is certainly more competitive than the other two reference designs.

5.3. Application of the Neuromorphic Processor to Character Recognition

We conduct behavior-level digital simulation of the chip to demonstrate the functionality of the neuromorphic processor designed in this article. The behavioral simulation is necessary as gate or transistor level simulation of long training processes requires huge CPU times, making it practically infeasible. To realistically capture the functionality of the designed processor and its dependencies on important hardware design choices, key network and design parameters including the digital LIF neuron dynamics, the STDP learning rule, bit-widths of the neuron model and synapses as in Table II are fully captured in behavioral simulation.

We specifically consider the case where the proposed DNP is configured to be a two-layer learning network for character recognition as illustrated in Figure 15 [Esser et al. 2010]. The network has an input-and-output layer structure with 232 excitatory and 7 inhibitory neurons and is designed to recognize alphabets "A" $-$"Z" by unsupervised learning. The input layer has 196 excitatory neurons, which form a two-dimensional array. Each excitatory input neuron receives a binary input representing a pixel value in the 14×14 pixel input pattern and projects its output to all excitatory output neurons through plastic synapses. In the input layer, the excitatory neurons project signals to 6 inhibitory neurons which provide negative feedback to modulate the firing frequencies of all excitatory neurons. The output layer consists of 36 excitatory neurons each of which receives input from all the input excitatory neurons. Structurally similar to the input layer, one inhibitory neuron is also employed in the output layer to provide strong negative feedback. The inclusion of this inhibitory neuron implements a winner-take-all (WTA) mechanism, where any firing output neuron activates the inhibitory neuron and thereby prevents other output neurons from firing through the negative feedback formed through the inhibitory neuron.

To train the network, we first convert each training letter, which is composed of a 14×14 pixel map, to 196 ($=14^2$) parallel input spike trains as the inputs to the network in Figure 15. The corresponding input neuron is either silent or active to encode a binary pixel. Then, for each alphabet from "A" to "Z", the corresponding input spike trains are applied to the respective input neurons for 5,000 biological time steps. As described earlier, network connectivity can be configured by properly programming

Neuron Index Mapping

Neuron Type	Neuron Index
Input Excitatory	1 ~ 196
Output Excitatory	197 ~ 232
Input Inhibitory	233 ~ 238
Output Inhibitory	239

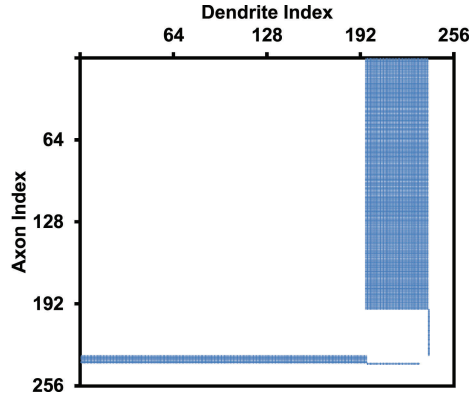

Fig. 16. Neuron index mapping and synaptic connections of the crossbar array.

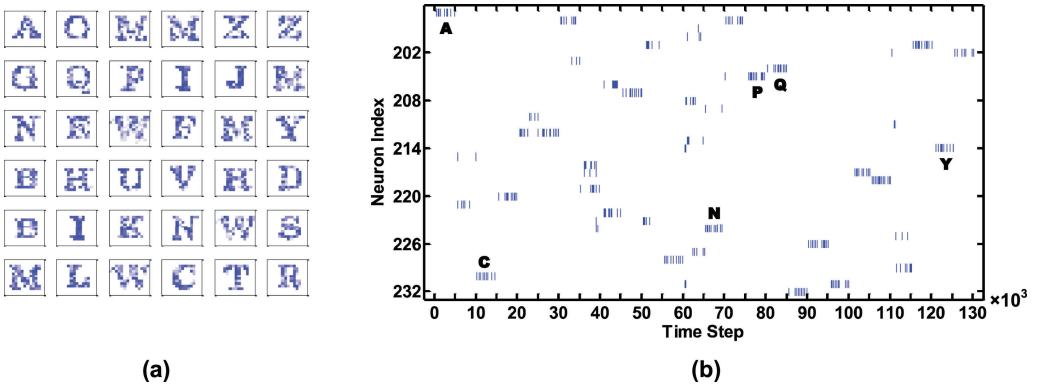

Fig. 17. Learning results for network: (a) Receptive fields after training, and (b) spike rasters for output neurons.

the memristive crossbar. For the configured two-layer character recognition chip, the index mapping of Figure 16 is used to identify each of these 256 neurons. The network connectivity matrix defined by the synaptic connections of the 256×256 memristive crossbar array is shown in Figure 16, where each dot represents the connection (i.e., a memristor conductance level greater than 0) between a pair of neurons.

The weights of all plastic synapses are set to random values before the training. With any input pattern, the net input received by each excitatory output neuron can be thought as the inner product of the presynaptic weight vector and a signal vector representing the activities of the excitatory input neurons. The weight vector of each output neuron corresponds to its receptive field, which describes the input pattern whose presence leads to excitation of the corresponding output neuron. During the training process, the network reshapes receptive fields of some excitatory output neurons to memorize each alphabet such that these neurons receive strong excitations and emit spikes with the presence of corresponding input pattern.

To demonstrate the recognition functionality of the proposed processor, we depict the simulated training results of the network in Figure 17. The receptive fields of the network after the training are shown in Figure 17(a). As can be seen, the receptive fields are well shaped by the training in the sense that every letter from "*A*" to "*Z*" appears at least once in the fields. This implies that during the recognition phase the presence of a letter is expected to excite at least one output neuron whose receptive

field closely resembles the presented letter, signifying the correct recognition of the letter.

The spike rasters for the 36 output excitatory neurons, which correspond to the neuron indices from 197 to 232, respectively, during the training process are plotted in Figure 17(b). Due to the implemented WTA mechanism, each input pattern has the tendency to excite only one or a few output neurons and all other output neurons are inhibited through the negative feedback. For instance, the letter "*A*" is presented from the first to the 5000th biological time steps in training. The 197th neuron's receptive field is trained to resemble "*A*" and this neuron is the only output neuron that actively fires in this period. In short, the 197th neuron is the winner when alphabet "*A*" is presented. For the training of some letters such as "*B*", it is possible to have a small number of winners. For example, more than one output neuron is trained to have a "*B*" shaped receptive field. Similarly, as shown in Figure 17(a), all other letters "*C*" −"*Z*" are memorized by the neural network. That is, when a letter is applied as input, the corresponding one or multiple output neurons fire, indicating correct recognition of the input letter. Therefore, the neural network achieves a recognition rate of 100%. In Figure 17(b), we mark a few representative output neurons that have fired appropriately to recognize the corresponding alphabets. For ease of visualization, only a subset of these neurons are marked in the figure.

5.4. Impacts of Device Variability and Noise on the Neuromorphic Application

Most electrical devices are prone to process variations and noise. This vulnerability is pronounced in scaled VLSI technologies and may degrade circuit performance and produce operational failures. As a nanodevice, memristors are also susceptible to process variations and noise [Hu et al. 2012; Niu et al. 2010; Querlioz et al. 2013]. Currently, there is a lack of quantitative memristor device variation data. To shed light on the impacts of device variability on the performance of the targeted neuromorphic application, we assume that these nonidealities of the memristors and analog circuits (e.g., ADCs, summing amplifier) may introduce certain degrees of errors in the synaptic crossbar read/write process during the neuron stage. We model the collective effect of these errors by adding an accumulated error at the column ADC output. On the other hand, we assume that our digital processing core is error free since it operates at a very low clock rate (e.g., 1 MHz). This is a safe assumption given the capability of modern CMOS technologies.

Specifically, we perturb the digital output of the column ADC randomly by a certain percent during the neuron stage. To have a semi-quantitative understanding of robustness of learning performance, three degrees of variability, i.e., 5%, 10%, and 20%, are considered at the ADC output. Note that these variability levels are fairly large. The effective number of bit (ENOB) of the ideal column ADC is 12-bits in the full data range. The losses of ENOB are then 8-, 8-, and 9-bits with 5%, 10%, and 20% variations of the column ADC outputs, respectively. The same character recognition system in Figures 15 and 16 is used to examine the resulting learning performance degradations in terms of the trained receptive fields as shown in Figure 18. The ENOB of the ideal column ADC is about 5-bits and the losses of ENOB are 2-, 2-, and 3-bits for 5%, 10%, and 20% of the ADC output variations, respectively, based on the actual processing workloads during the neuron stage.

Thanks to the built-in resilience of the neuromorphic architecture, the receptive fields have been trained to recognize a majority of the alphabets even with presence of the modeled variability. With 5% variation of the column ADC output, the trained receptive fields of the output excitatory neurons cover all alphabets except for "E" and "Q" as in Figure 18(a). 10% and 20% variations results in four ("I", "Q", "V", "X") and six ("E", "F", "G", "I", "O", "V") missing alphabets out of a total of 26 in the corresponding receptive

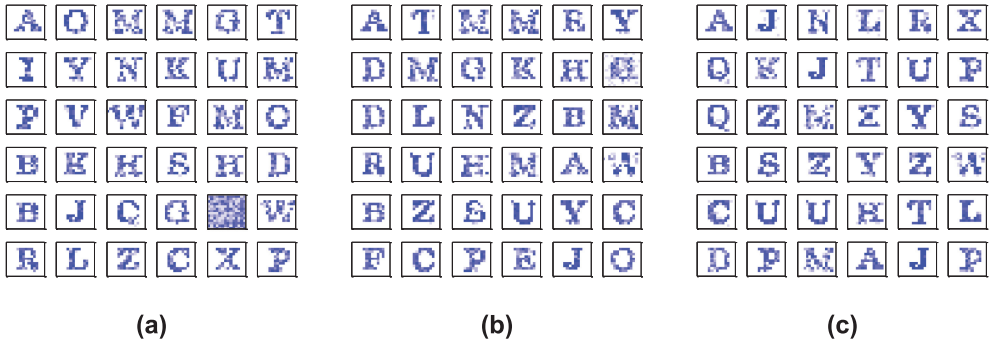

Fig. 18. Receptive fields after training under a variability degree of (a) 5%, (b) 10%, and (c) 20%.

fields, respectively, as shown in Figures 18(b) and 18(c). These results translate into a corresponding recognition rate of 92.3%, 84.6%, and 76.9%, respectively. It is clear that the variations do have an impact on performance. However, given that the levels of variations considered are fairly large and the resulting performance degradations are quite reasonable, we do observe good robustness of the realized neuromorphic system.

6. CONCLUSION

In this article, we have presented a scalable digital neuromorphic processor architecture for large scale integration of spiking neurons. The proposed architecture targets the realization of brain-inspired learning based on spike-timing-dependent plasticity rules. One of the main focuses of this work is to propose to use memristor nanodevices to support a high degree of connectivity between neurons, a desired key feature of neuromorphic computation. The proposed memristive synaptic crossbars store both multibit synaptic weight values and network connectivity information with low area overhead. The crossbar arrays are designed to be accessible both column- and row-wise. Efficient accessing schemes and interface circuits are developed to expedite read and write accesses that are key to the processing efficiency of the targeted neuromorphic architecture.

With 256 silicon neurons, learning circuits and 64K synapses, the power dissipation and area of our DNP are 6.45 mW and 1.86 mm^2, respectively, when implemented in a commercial 90-nm CMOS technology. The functionality of the proposed DNP architecture is demonstrated by realizing an unsupervised-learning-based character recognition system whose learning performance is validated by developing and adopting a behavioral digital simulation environment.

The robustness of our digital neuromorphic processor (DNP) architecture is studied by injecting high degrees of variations at the output of the column ADC, mimicking the effects of device variability and noise of the memristive crossbar and the analog-to-digital conversion. For the character recognition system, the resulting learning performance degradations are empirically examined. The presented neuromorphic system appears to be robust with respect to these nonidealities.

Our results have suggested that neuromorphic architectures employing biologically-inspired learning principles may be realized efficiently in silicon, in particular, with use of emerging nonvolatile memory devices (e.g., memristors). In addition, it is also suggested that neuromorphic architecture such as the one demonstrated in this article may provide a promising paradigm for building new generations of error-tolerant VLSI computing systems.

REFERENCES

J. V. Arthur, P. A. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, S. Chandra, S. K. Esser, N. Imam, W. Risk, D. B. D. Rubin, R. Manohar, and D. S. Modha. 2012. Building block of a programmable neuromorphic substrate: A digital neurosynaptic core. In *Proceedings of the International Joint Conference on the Neural Networks (IJCNN)*. 1–8.

S. Brink, S. Nease, P. Hasler, S. Ramakrishnan, R. Wunderlich, A. Basu, and B. Degnan. 2013. A learning-enabled neuron array IC based upon transistor channel models of biological phenomena. *IEEE Trans. Biomed. Circuits Syst.* 7, 1, 71–81.

L. O. Chua. 1971. Memristor - The missing circuit element. *IEEE Trans. Circuit Theory* 18, 5, 507–519.

J. Cosp, J. Madrenas, and D. Fernandez. 2006. Design and basic blocks of a neuromorphic VLSI analogue vision system. *Neurocomputing* 69, 16–18, 1962–1970.

I. E. Ebong and P. Mazumder. 2012. CMOS and memristor-based neural network design for position detection. *Proc. IEEE* 100, 6, 2050–2060.

S. K. Esser, A. Ndirango, and D. S. Modha. 2010. Binding sparse spatiotemporal patterns in spiking computation. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 1–9.

Y. Ho, G. M. Huang, and P. Li. 2011. Dynamical properties and design analysis for nonvolatile memristor memories. *IEEE. Trans. Circuits Syst. I*, (Reg. Papers), 58, 4, 724–736.

M. Hu, H. Li, Q. Wu, and G. S. Rose. 2012. Hardware realization of BSB recall function using memristor crossbar arrays. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC)*. 498–503.

N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D.S. Modha. 2012. A digital neurosynaptic core using event-driven QDI circuits. In *Proceedings of the IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. 25–32.

G. Indiveri, E. Chicca, and R. Douglas. 2006. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.*

G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Haliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen. 2011. Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5, 73.

S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu. 2010. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters* 10, 4, 1297–1301.

J. Kim, T.-K. Jang, Y.-G. Yoon, and S. Cho. 2010. Analysis and design of voltage-controlled oscillator based analog-to-digital converter. *IEEE. Trans. Circuits Syst. I* (Reg. Papers) 57, 1, 18–30.

K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu. 2012. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Letters* 12, 1, 389–395.

Y. Kim, Y. Zhang, and P. Li. 2012. A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning. In *Proceedings of the IEEE International SOC Conference (SOCC)*. 328–333.

H. Manem, J. Rajendran, and G. S. Rose. 2012. Design considerations for multilevel CMOS/nano memristive memory. *ACM J. Emerg. Technol. Comput. Syst.* 8, 1, Article 6, 6:1–6:22.

T. M. Massoud and T. K. Horiuchi. 2011. A neuromorphic VLSI head direction cell system. *IEEE. Trans. Circuits Syst. I* (Reg. Papers) 58, 1, 150–163.

C. E. Merkel, N. Nagpal, S. Mandalapu, and D. Kudithipudi. 2011. Reconfigurable N-level memristor memory design. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 3042–3048.

P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D.S. Modha. 2011. A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*. 1–4.

S. Mitra, S. Fusi, and G. Indiveri. 2009. Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *IEEE Trans. Biomed. Circuits Syst.* 3, 1, 32–42.

D. Niu, Y. Chen, C. Xu, and Y. Xie. 2010. Impact of process variations on emerging memristor. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC)*. 877–882.

Y. V. Pershin and M. D. Ventra. 2010. Experimental demonstration of associative memory with memristive neural networks. *Neural Netw.* 23, 7, 881–886.

D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat. 2013. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* 12, 3, 288–295.

J.-S. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha, and D. J. Friedman. 2011. A 45nm CMOS neuromorphic chip with a scalable architecture

for learning in networks of spiking neurons. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*. 1–4.

R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, Shih-Chii Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, and B. Linares-Barranco. 2009. CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking. *IEEE Trans. Neural Netw.* 20, 9, 1417–1438.

T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco. 2013. STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Front. Neurosci.* 7, 2 (2013).

G. S. Snider. 2008. Spike-timing-dependent learning in memristive nanodevices. In *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. 85–92.

D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. 2008. The missing memristor found. *Nature* 453, 80–83.

A. Syed, E. Ahmed, D. Maksimovic, and E. Alarcon. 2004. Digital pulse width modulator architectures. In *Proceedings of the IEEE Power Electronic Specialists Conference (PSEC)*, Vol. 6. 4689–4695.

A. van Schaik. 2001. Building blocks for electronic spiking neural networks. *Neural Netw.* 14, 6–7, 617–628.

J. H. B. Wijekoon and P. Dudek. 2008. Compact silicon neuron circuit with spiking and bursting behaviour. *Neural Netw.* 21, 2–3, 524–534.

C. Xu, X. Dong, N. P. Jouppi, and Y. Xie. 2011. Design implications of memristor-based RRAM cross-point structures. In *Proceedings of the Symposium on Design, Automation and Test in Europe (DATE), 2011*. 1–6.

J. J. Yang and R. S. Williams. 2013. Memristive devices in computing system: Promises and challenges. *ACM J. Emerg. Technol. Comput. Syst.* 9, 2, 11:1–11:20.

Y.-G. Yoon, J. Kim, T.-K. Jang, and S. Cho. 2008. A time-based bandpass ADC using time-interleaved voltage-controlled oscillators. *IEEE. Trans. Circuits Syst. I*, (Reg. Papers) 55, 11, 3571–3581.

Received July 2013; revised November 2013, March 2014, and May 2014; accepted September 2014