Title

Neuromorphic processors with memristive synapses: Synaptic interface and architectural exploration

Authors

Wang, Qian
Kim, Yongtae
Li, Peng

Publication Date

2016

Peer reviewed

Neuromorphic Processors with Memristive Synapses: Synaptic Interface and Architectural Exploration

QIAN WANG, Texas A&M University
YONGTAE KIM, Intel Corporation
PENG LI, Texas A&M University

Due to their nonvolatile nature, excellent scalability, and high density, memristive nanodevices provide a promising solution for low-cost on-chip storage. Integrating memristor-based synaptic crossbars into digital neuromorphic processors (DNPs) may facilitate efficient realization of brain-inspired computing. This article investigates architectural design exploration of DNPs with memristive synapses by proposing two synapse readout schemes. The key design tradeoffs involving different analog-to-digital conversions and memory accessing styles are thoroughly investigated. A novel storage strategy optimized for feedforward neural networks is proposed in this work, which greatly reduces the energy and area cost of the memristor array and its peripherals.

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: Neural networks, digital integrated circuits, analog-digital conversion, memristors, reconfigurable architectures

ACM Reference Format:
Qian Wang, Yongtae Kim, and Peng Li. 2016. Neuromorphic processors with memristive synapses: Synaptic interface and architectural exploration. J. Emerg. Technol. Comput. Syst. 12, 4, Article 35 (May 2016), 22 pages.
DOI: http://dx.doi.org/10.1145/2894756

1. INTRODUCTION

The human brain is the control center for all our body movements, thinking functions, emotions, cognitive activities, and other complex tasks. Although most real-world applications that involve the processing of sensory inputs and pattern recognition are still difficult tasks even on a supercomputer, a human brain can solve these problems easily and show even better performance with great energy and space efficiency. In contrast, conventional von Neumann machines may require tremendous energy consumption and space resources to achieve the same if it is all possible [Arthur et al. 2012]. Brain-inspired neuromorphic computing provides an appealing architectural solution for the above problems and shows good energy efficiency, potentially improved scalability, and great suitability for processing complex tasks such as image recognition, classification, and language learning. Meanwhile, the inherent error resilience

Author's addresses: Q. Wang, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843; email: qwangku@tamu.edu; Y. Kim, Intel Corporation, Santa Clara, CA 95054; email: yongtae.kim@intel.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/10.1145/2894756

and fault tolerance offered by the brain-inspired architectures are very suitable for large-scale integration in VLSI technologies.

Traditionally, analog circuits are used to implement the silicon neurons [Mitra et al. 2009; van Schaik 2001]. However, they are difficult to reconfigure and intrinsically sensitive to process, voltage, and temperature (PVT) variations. In addition, large-scale integration of spiking neurons is hindered by the use of area-consuming capacitors as to keep synaptic weights [Indiveri et al. 2006]. Recently, Merolla et al. [2011] and Seo [2011] have demonstrated two digital reconfigurable neuromorphic chips. These two designs support up to 256 programmable digital neurons as well as 1024×256 binary synapses by means of an SRAM (Static Random-Access Memory) crossbar array. However, the corresponding binary synapses are updated by a probabilistic scheme, which may degrade the learning performance. Moreover, the SRAM array occupies a significant portion of the entire chip area.

Memristive nanodevice provides a promising solution for on-chip storage thanks to its nonvolatile nature and high integration density reaching $10Gb/cm^2$ [Ho et al. 2009; Merkel et al. 2011]. Several recent studies have suggested leveraging memristive nanodevices for building synaptic arrays [Jo et al. 2010a; Snider 2008]. A high-density, fully operational hybrid crossbar/CMOS (Complementary Metal-Oxide Semiconductor) system composed of a memristor crossbar array has been demonstrated in Kim et al. [2011], which can reliably store complex binary and multilevel data. Chen et al. [2014] proposes a memristor crossbar array system for image processing and demonstrates a good performance for noise reduction. Meanwhile, efficient hardware implementations of neural networks based on Resistive Random Access Memory (RRAM) crossbar arrays have been demonstrated in Hu et al. [2012] and Li et al. [2015].

A brain-inspired reconfigurable digital neuromorphic processor (DNP) architecture for large-scale spiking neural networks is presented in Kim et al. [2012] and Wang et al. [2014], which supports the spike timing-dependent plasticity (STDP) learning mechanism. This design is implemented in a commercial 90nm CMOS technology and leverages the memristor nanodevice to build a 256×256 crossbar array to store multi-bit synaptic weight values with significantly reduced area cost. Realizing memristor array-based DNPs entails addressing a number of critical issues pertaining to the memory access styles, analog-to-digital (A/D) conversion, and optimized storage organization. However, a systematic analysis of the above issues is lacking in the previous works. The main goal of this work is to investigate critical design decisions and identify key tradeoffs between energy and area for DNPs with different synapse readout schemes and storage organizations.

The memristor crossbar array has many advantages over SRAM and DRAM (Dynamic Random-Access Memory) in terms of high integration density and nonvolatile nature, but the synaptic weight values stored in the memristor array are essentially continuous-valued analog signals (i.e., conductance and current), which cannot be directly processed by the digital arithmetic components in the DNP. Typically, in such mixed-signal systems, the analog-to-digital converters (ADCs) make up a large portion of the total power consumption and chip area. Therefore, an efficient analog-to-digital conversion scheme for synapse readout plays an extremely important role in the design of DNPs. Crucial design choices and tradeoffs involving different memory access styles and different types of ADCs are systematically investigated in this work.

Figure 1 compares the most popular ADC architectures in terms of number of bits and the sampling frequency range [Murmann 1997]. From the application point of view, the ADC sampling rate should be consistent with the frequency of the neuromorphic chip, which is 1MHz. The synapse readout from the memristor crossbar array can be realized by using either an array of low-resolution ADC (3 bits) or high-resolution ADC (over 12 bits).

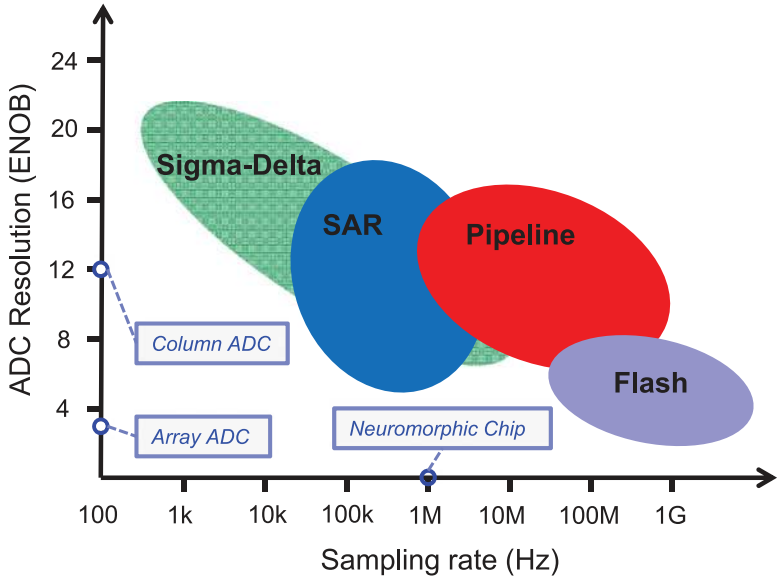

Fig. 1. Comparison of ADC architectures vs. resolution and sampling rate.

Two memory access styles are proposed, which are referred to as the columnwise scheme and the rowwise scheme. Hence, there exists a large design space for optimization of area and energy consumption. While targeting spiking neuron processors with N neurons in a commercial 90nm CMOS technology, our design analysis shows that among the fully reconfigurable architectures that are based on $N \times N$ synapse crossbar arrays, utilizing a high-resolution Sigma-Delta (SD) ADC for the column readout provides the best energy efficiency, while the column readout scheme based on a flash ADC array shows the smallest chip area. Our analysis also highlights the tradeoffs involved in various other ADC strategies available for synapse readout.

In addition, this work proposes an optimized synapse storage scheme for a wide class of feedforward spiking neural networks, which reduces the energy consumption by 70% compared with those based on a full $N \times N$ memristor array. The corresponding area cost of memories is also much smaller. These encouraging results suggest the great potential of the proposed architecture for building DNPs with high energy and area efficiency for large-scale feedforward neural networks. While focusing on 256- and 891-neuron DNPs in 90nm CMOS, the presented architectural design exploration can be adopted for large networks at other technology nodes.

2. THE DIGITAL NEUROMORPHIC PROCESSOR ARCHITECTURE

The leaky integrate-and-fire (LIF) model is adopted in this work for the silicon neurons to mimic the biological counterparts, which proves to be effective for a number of learning applications and is suitable for digital implementation due to its moderate hardware overhead [Indiveri et al. 2011]. Figure 2 depicts the overall block diagram of the DNP architecture with a $N \times N$ memristive synapse array. It consists of a synapse unit (SU), a learning unit (LU), a neuron unit (NU), and a LIF arithmetic unit (LAU). Let N denote the total number of neurons in the network. The SU employs an $N \times N$ memristor crossbar structure, which can represent a fully recurrent neural network topology and support N^2 possible synaptic connections among all the neurons. In this memristor array, a row and a column correspond to a dendrite and an axon,

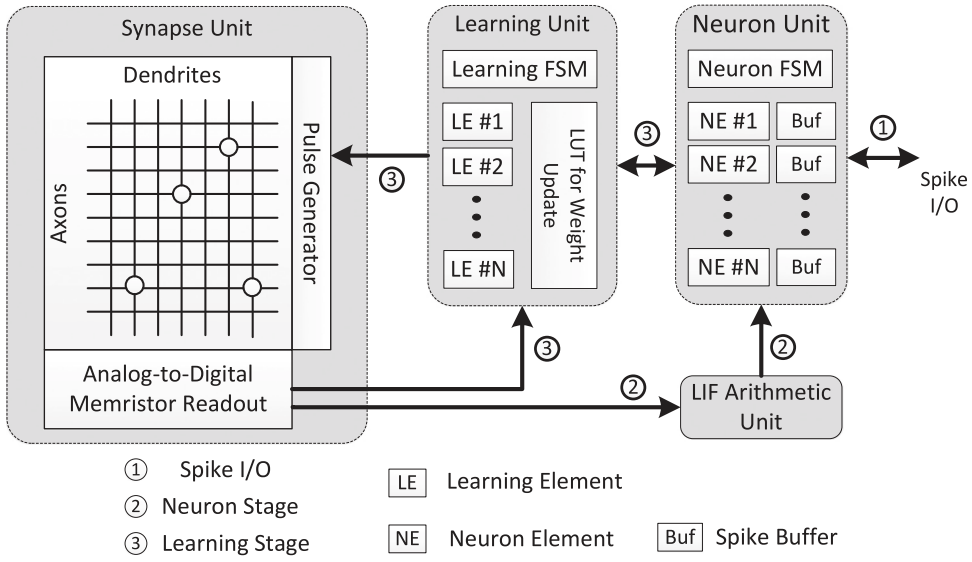

Fig. 2. Block diagram of the baseline digital neuromorphic processor architecture.

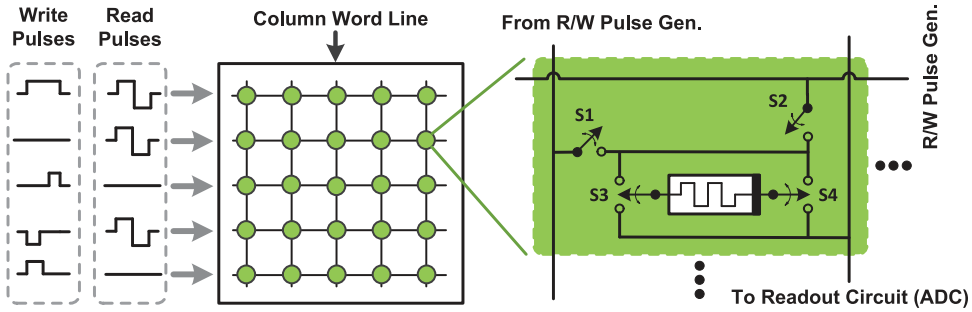

Fig. 3. Proposed synaptic crossbar array and CMOS/memristor hybrid synaptic cell. Parallel voltage pulses are generated by the R/W pulse generator and used for the read and write of all cells in the row (column).

respectively, for a biological neuron. Therefore, the connection between the (j)th row and (i)th column corresponds to the synapse between the (j)th and (i)th neurons.

The conductance of a memristive device can be incrementally adjusted by altering the pulse width of the constant input voltage [Jo et al. 2010b]. In other words, longer positive pulse duration leads to a larger increase of memductance. Therefore, an R/W (Read/Write) pulse generator is required for the access of either a column or a row of the memristor array. For the purpose of parallel read and write, the R/W pulse generator is designed to send out N parallel pulses simultaneously.

The proposed synaptic crossbar array and the synaptic cell are exhibited in Figure 3. The two switches S_1 and S_2 in the cell allow each memristive device to be accessed in both column and row fashion. When the row (column) driver activates a word line, S_1 (S_2) of all cells in the same row (column) are switched on, and the corresponding memristors are ready to be accessed. In order to allow the conductance of each memristor to be decreased by a negative voltage pulse, S_3 and S_4 are introduced to connect the two terminals of a memristor to either the ADC or the pulse generator, respectively.

The control flow of the DNP involves three processing stages, namely, the spike I/O (Input/Output) stage, the neuron stage, and the learning stage. As shown in Figure 2,

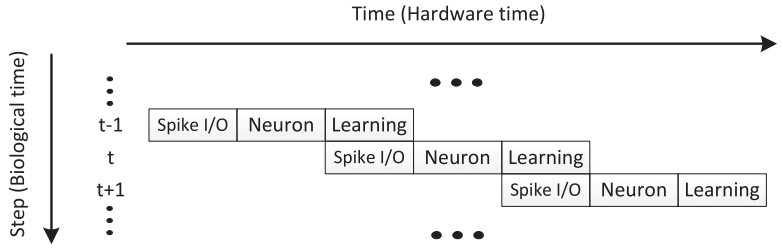

Fig. 4. Flow diagram of the digital neuromorphic processor.

during the spike I/O stage, input spike buffers in NU receive the spikes from the external environment. Meanwhile, the output spikes can be read off the chip to observe the output activities.

Then the neuron stage starts, where the following dynamics is implemented for each neuron element (NE) inside NU

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

where V_i is the membrane potential of neuron i, M is the number of pre-synaptic neurons, K_{SYN} is the synaptic weight parameter, w_{ji} is the synaptic weight between neurons j and i, S_j is the activity bit which indicates whether the neuron j fired (i.e., $S_i = 1$ if $V_i[t] \geq V_{Threshold}$), K_{EXT} is the external input spike parameter, E_i is the activity bit for the input spike, and V_{LEAK} is the leaky parameter. In this stage, the R/W pulse generator generates pulses for reading all pre-synaptic weight values. At the same time, the analog-to-digital readout block accumulates these pre-synaptic weights and transforms them into a digital quantity. This accumulation process can be realized in two ways. One is to sum the synaptic weights in the analog domain and then convert the summed result to a digital value with a high-resolution ADC. The other is to use an array of low-resolution ADCs to obtain all the digital weight values from a column (pre-synapses) and then accumulate them in digital domain. Finally, the accumulated presynaptic weights are sent to the LAU to perform the calculation of (1), and the NE updates its membrane potential based on the result from LAU. If the membrane potential exceeds the given threshold voltage, then the NE generates a spike event that indicates that the corresponding neuron fires.

After all the NEs have gone through the above process, the processing moves onto the learning stage according to the STDP rule. In this rule, each learning element (LE) measures the time difference between a pre-synaptic and a post-synaptic spike event to determine the synaptic weight change. The biological time of each neuron spike event is recorded by a time register inside each LE. If a neuron fires, then all of its pre-(post-) synaptic neurons' time registers are compared with a global timer representing the current biological time. The amounts of the synaptic weight changes are calculated by the corresponding LEs with a shared lookup table (LUT) inside the LU. Therefore, according to the amounts of the synaptic weight changes obtained by LU, the pulse generator produces parallel write pulses with different widths to update the internal states of memristors in a particular column (row).

The system controller manages the overall operations of the system through clocking-based synchronous control and the system operates in a synchronous manner, as shown in Figure 4. Each step corresponds to a biological time unit and consumes many hardware clock cycles. The three stages are executed in a pipelined manner in that the spike I/O and learning stages can work simultaneously because there are no data and control hazards between them.

3. THE PROPOSED ARCHITECTURES

As mentioned in the previous sections, the readout of the synaptic weights is a central problem associated with DNPs based on memristive synapses, which requires efficient analog-to-digital conversion and suitable memory access styles. The key problems such as column/row readout, choices of ADCs, and ways to improve storage utility are thoroughly studied and addressed brieflyin this section. Based on the baseline DNP design discussed in the previous section, we investigate a range of architectural design variants.

The memristor crossbar array can be accessed either columnwise or rowwise, and a range of ADC designs with different architectures and associated resolution, area, and power consumption tradeoffs can be used for the analog-to-digital conversion of the DNP. However, integrating one or more of such ADCs into the DNP requires a systemic investigation of memristive memory access styles to minimize power and area overhead as discussed below.

Two synapse storage strategies are developed for the proposed architectures: One is the full-size $N \times N$ memristor array, and the other is the optimized storage strategy for feedforward neural network topologies. Both can be implemented with different memory access styles and ADC architectures.

3.1. Memory Access Styles

In this work, we propose two different memristor array access styles. The first one is referred to as the columnwise readout, in which all the columns are sequentially accessed and the accumulated synaptic weights for each column is obtained one at a time. The integration element (IE) inside the LIF arithmetic unit is used for the calculation of (1). Although the columnwise approach shown in Figure 5(a) involves multiple IEs, these IEs do not work simultaneously. Therefore, the membrane potentials of the digital neurons are not updated in parallel. Since only one IE is needed to process the synaptic weights from a particular column, it is possible to have all the NEs inside the NU share only one IE, and this shared IE approach is illustrated in Figure 5(b), which requires a large N-input multiplexer (MUX). The readout scheme involving only one IE is referred to as the shared IE scheme, while the readout scheme involving multiple IEs is referred to as the nonshared IE scheme.

Figure 5(c) shows the second memristor array access style proposed, which is referred to as the rowwise readout, where the memristor array is accessed row by row. Although only one synaptic weight is read out for each neuron, totally N synaptic weights are actually read out for all the N neurons for each row access. The neuron stage of the rowwise approach is further divided into two stages. In the first stage, N accumulators work in parallel to sum the synaptic weights in their corresponding columns, and N cycles are required to obtain the accumulated synaptic weights for all the neurons. Once all the rows have been accessed, the second stage starts and all the N membrane potentials are updated in parallel, which requires only one cycle. Therefore, the total number of cycles consumed by the neuron stage of the rowwise approach is the same as that of the columnwise approach.

3.2. Analog-to-Digital Conversion

Analog-to-digital conversion is essential for synapse readout in both the neuron stage and the learning stage.

During the neuron stage, the LIF arithmetic unit only needs the accumulated synaptic weights from a partcular column, instead of each individual synaptic weight in this column. Therefore, the synapse readout can be achieved in two ways. One way is to use N low-resolution ADCs to read out all the N synaptic weights from a column in parallel and then sum them with an N-input digital adder, as shown in Figure 5. The

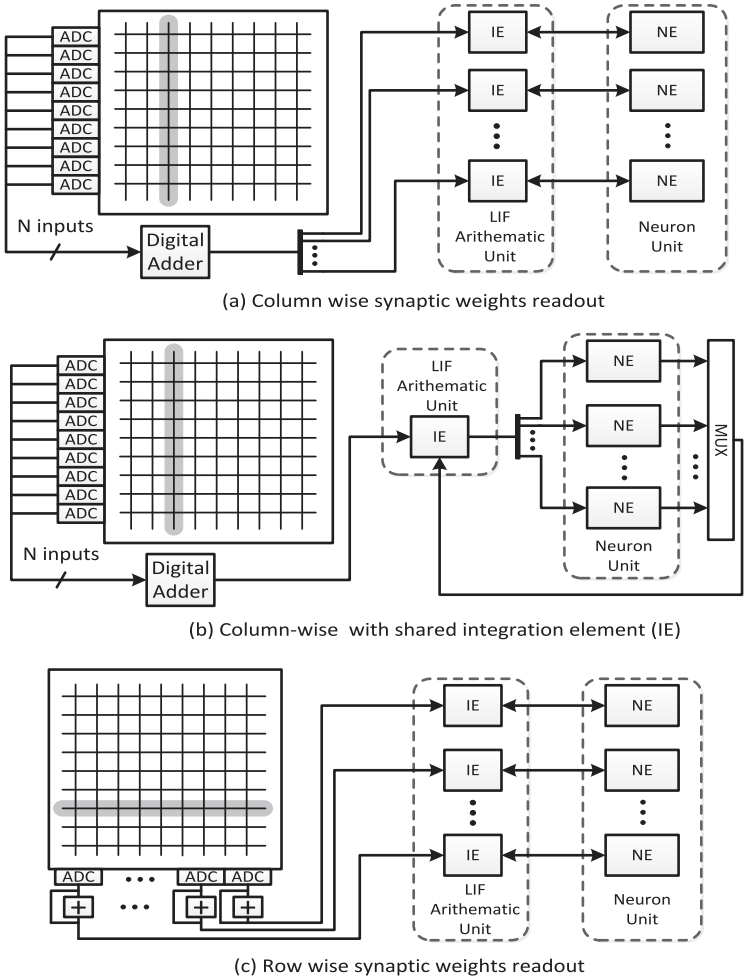

Fig. 5. Different memory access styles for neuron stage: (a) Readout synaptic weights column by column with N integration elements (IEs); (b) read out synaptic weights column by column with only one shared IE; (c) read out synaptic weights row by row with N low-resolution ADCs and N accumulators.

other way is to use a summing amplifier to obtain the sum of the synaptic weights in the analog domain and then convert it into a digital value with a high-resolution ADC (also called the Column ADC), as shown in Figure 7.

During the learning stage, however, we should know the corresponding memristor's current internal state, so the pulse duration to write the desired synaptic weight to each memristor in a row/column can be determined. In this regard, N low-resolution ADCs should be used to read all the pre-(post-) synaptic weights of each column (row) in parallel. Therefore, a low-resolution ADC array is indispensable to all the proposed architectures. Obviously, these N low-resolution ADCs can be reused during the neuron stage, according to the first accumulation scheme just mentioned.

This work focuses on five typical ADC architectures, and Figure 6 compares powers and areas of these mainstream ADCs with various resolutions, which was evaluated based on the models in Huang and Zhong [2004] with 90nm CMOS technology. According to Figure 6, the flash ADC architecture is obviously the best candidate for low-resolution analog-to-digital conversion (i.e., three-bit resolution), while the other

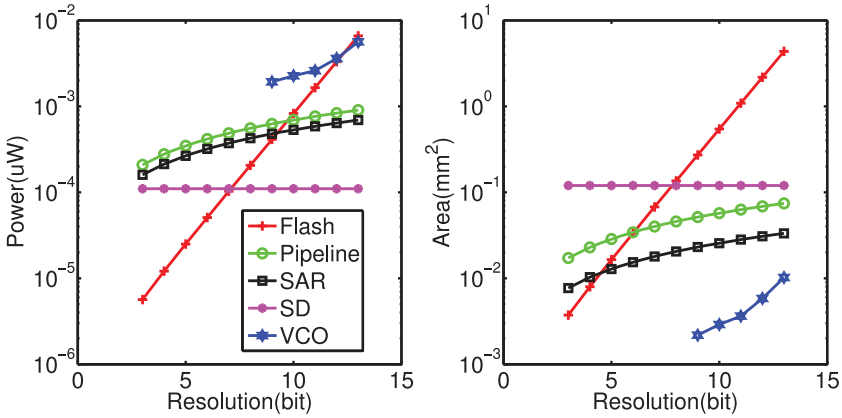

Fig. 6. Power and area for different ADCs of various resolutions.

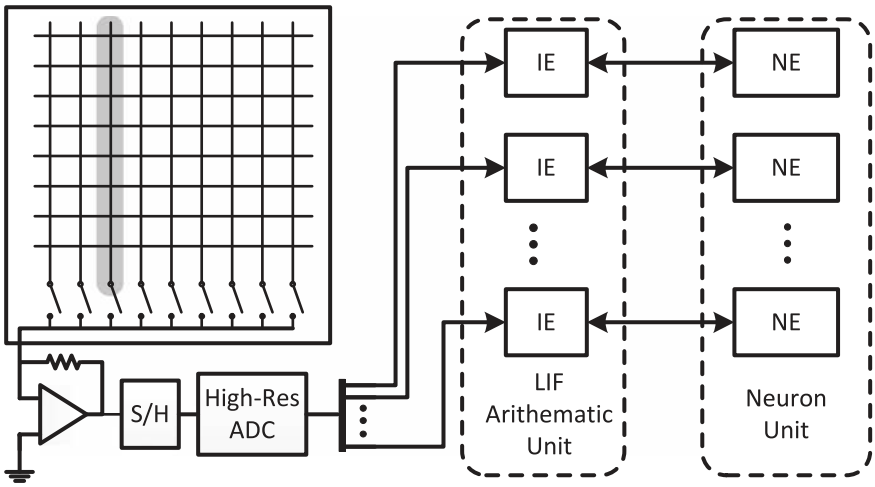

Fig. 7. Block diagram of the readout with column ADC.

ADC architectures are suitable for high-resolution conversion with different power-area tradeoffs.

For a flash ADC with resolution b, the power consumption can be estimated by Verbruggen et al. [2009],

$$P_{flash} = (2^b - 1)P_{cmp} + P_x, \tag{2}$$

where P_{cmp} and P_x are powers of comparator and encoder, respectively. Obviously, the flash ADC is a good choice for the low-resolution ADC array. However, it suffers from considerable power and area consumption for high-resolution A/D conversion, which prevents it from being used as column ADC.

In the columnwise approach with either shared IE or multiple IEs, alternatively, the neuron stage synapse readout can be achieved by using one high-resolution ADC. While compared with reusing the low-resolution ADC array for the neuron stage, introducing this additional high-resolution ADC may lead to lower energy consumption at the cost of minor area overhead. This high-resolution ADC is referred to as the column ADC. As shown in Figure 7, a summing amplifier (i.e., current-to-voltage converter) is used

to provide the linear summation of conductance of memristors in the analog domain. Then the obtained analog sum is converted to a digital value with a high-resolution column ADC. As with the readout scheme based on a low-resolution ADC array, in the column ADC-based readout scheme, the accumulation of the synaptic weights for a particular column can be finished within one clock cycle. The desired resolution of the column ADC is derived by

$$resolution = [log_2 N + log_2 L], \tag{3}$$

where N and L are the numbers of neurons and conductance levels of the memristor cell in the array, respectively. In Kim et al. [2012], a VCO-based column ADC is adopted for column readout. However, several other important choices exist.

The successive approximation register (SAR) ADC holds the analog input signal on a sample/hold [Harpe et al. 2011]. It then converts this analog signal into a digital value via a binary search through all possible quantization levels. The estimated power consumption for a SAR ADC with b bits resolution is

$$P_{SAR} = \frac{b}{m}\left(P_{S/H} + mP_{DAC} + (2^m - 1)P_{cmp}\right) + P_x, \tag{4}$$

where m is the number of bits per cycle and P_{cmp}, P_{DAC}, $P_{S/H}$, and P_x correspond to Comparator, Sub-DAC, Sample-and-Hold, and Control Logic/Register, respectively. SAR ADCs can provide the lowest hardware cost, but each conversion requires multiple clock cycles to converge to the required resolution.

Pipelined ADCs distribute the conversion process over multiple stages in sequence, and the overall throughput is close to one sample per clock cycle if the pipeline is fully occupied [Huang and Lee 2011]. For a N_{sta}-stage pipelined ADC with b-bit resolution, the estimated power is

$$P_{pip} = N_{sta}(P_{S/H} + (P_{DAC} + P_{gain})b/N_{sta} + (2^{b/N_{sta}} - 1)P_{cmp}) + P_x, \tag{5}$$

where $P_{S/H}$, P_{DAC}, P_{cmp}, P_{gain}, and P_x correspond to the Sample-and-Hold, Sub-DAC, Comparator, Gain stage, and the digital part.

Since, for our application, the LIF cannot start until the analog-to-digital conversion is completed, the advantage of pipeline is not utilized. In order for the other parts of the DNP to still operate at 1MHz, the clock rate of the SAR and the pipelined ADCs should be KMHz, assuming the required ADC resolution is K-bit.

The SD ADC achieves high resolution by oversampling the input at a frequency higher than the Nyquist rate [Shettigar and Pavan 2012]. The input analog signal passes through the integrator followed by a comparator. Then the output of the comparator is fed back via a sub-DAC to the input for summation. The output of the comparator also passes through the decimation filter at the output of the SD ADC. For a N_{order}-order SD ADC with b-bit resolution, the estimated power is

$$P_{SD} = R_{oversample}(N_{order}P_{intg} + P_{cmp} + P_{DAC}) + P_x, \tag{6}$$

where $R_{oversample}$ is the oversampling rate and P_{intg}, P_{cmp}, P_{DAC}, and P_x correspond to integrator, comparator, sub-DAC, and decimation circuits.

Comparator, Sample-and-Hold, Sub-DAC, Integrator, and Gain-stage are the five major component building blocks for ADCs. According to Huang and Zhong [2004], a universal function with different parameters can be employed to model the power consumptions of these blocks. The power modeling function is

$$P_i = \frac{\alpha_i \cdot V_{DD} - \beta_i \cdot V_{swing}}{\eta_i} \cdot V_{DD} \cdot L_{min} \cdot f_{sample}, \tag{7}$$

Table I. Coefficients in Power Modeling Function

	α_i	β_i	η_i
S/H	0.5	0.25	14.6×10^3
Comparator	0.5	0.30	32.1×10^3
Sub-DAC	0.5	0.20	27.5×10^3
Gain	0.5	0.20	28.7×10^3
Integrator	0.5	0.15	9.8×10^3

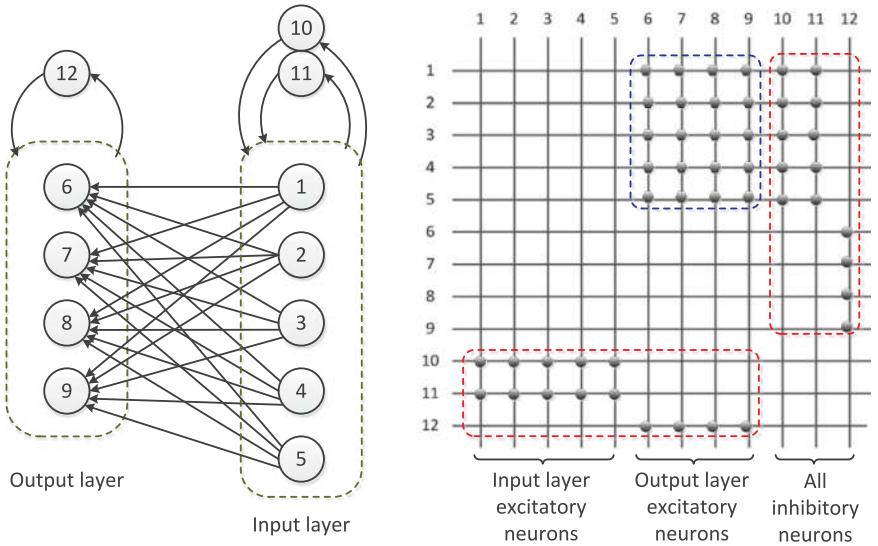

Fig. 8. An example of two-layer feedforward neural networks and its corresponding crossbar array.

where V_{DD} is the supply voltage and V_{swing} is the maximum signal voltage swing. L_{min} is the feature size of a particular CMOS technology, and f_{sample} is the sampling frequency. The values of α_i, β_i, and η_i vary for different building blocks, which are summarized in Table I. These coefficients are obtained from experimental data fitting and they have been validated with different commercial ADCs [Huang and Zhong 2004].

Clearly, these ADC architectures define a large design space. In this work, by properly modeling the area and power of each ADC as a function of the targeted technology, conversion speed, and resolution, we systematically evaluate the design tradeoffs associated with each choice. The detailed analysis is presented in Section 4.

3.3. Optimized Storage Strategy for Feedforward Neural Networks

All architectures in the previous sections are based on the $N \times N$ synaptic array which is fully reconfigurable. However, in reality, the neural network topologies are usually much sparser. Figure 8 shows an example of a typical two-layer feedforward neural network and the distribution of its synapses inside a conceptual $N \times N$ synaptic array. The neurons with indices 10, 11, and 12 are the inhibitory neurons, while all the other neurons are excitatory. Such network topologies have three important features:

(1) The synapstic weights involving inhibitory neurons are fixed so they are not updated during learning;
(2) The excitatory neurons within each layer are not connected to each other;
(3) There are no feedback synapses from the output layer neurons to the input layer neurons.

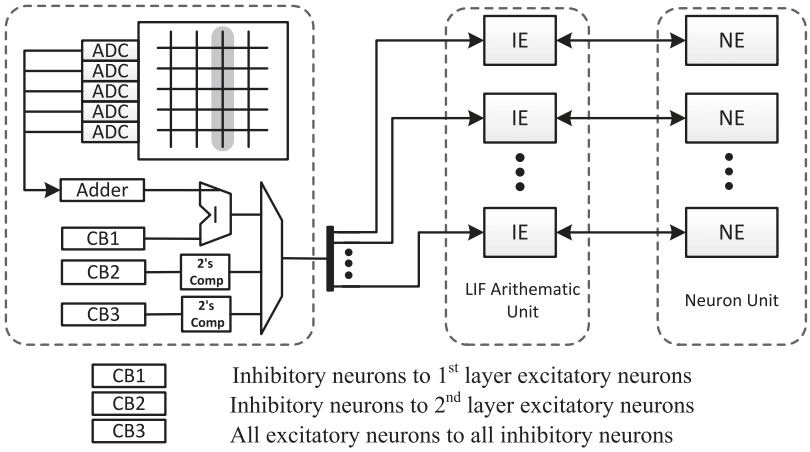

Fig. 9. The proposed storage organization optimized for a two-layer feedforward neural network. The constant blocks CB1, CB2, and CB3 are actually constants integrated into the digital design.

The first feature indicates that the synapse weights involving the inhibitory neurons can be simply integrated into the digital design rather than memristor arrays. The other two features guarantee that the synapses corresponding to the feedforwarding paths reside in a very small block of the conceptual $N \times N$ synaptic array so a full $N \times N$ memristor array is not necessary. In this case, the size of the flash ADC array and the size of the pulse generator will be greatly reduced. Moreover, the resolution of the column ADC may also be reduced by several bits.

Based on the above analysis, we propose an optimized architecture for typical feedforward neural networks, as illustrated in Figure 9. We only need to update the synaptic weights of the feedforward synapses this time, and all the other synapses are constants integrated into the digital design. The synaptic weights associated with the paths from inhibitory neurons to the input layer excitatory neurons are provided by the constant block CB1, while the synapses from the inhibitory neurons to the output layer excitatory neurons are provided by CB2. The weights of paths from all the excitatory neurons to the inhibitory neurons are provided by CB3. Each constant block is essentially combinational logic. Therefore, the hardware cost of CB1, CB2, and CB3 is trivial compared with other components in the system. Because the weights of synapses between the input layer and the output layer need to be updated during learning stage, they are stored in the memristor crossbar array. When columns 1 to 5 from the conceptual synaptic array in Figure 8 needs to be read out, we access CB1 for the desired synaptic weights. For the readout of columns 6 to 9, both the memristor array and CB2 are accessed. In the same way, we access CB3 for the desired synaptic weights in columns 10 to 12. Generally speaking, the readout procedure for this architecture is very similar to the architectures proposed in the previous section. However, fewer column accesses require analog-to-digital conversion and each analog-to-digital conversion involves fewer ADCs and smaller pulse generator. Therefore, the proposed architecture enjoys a significant improvement in energy efficiency for feedforward neural networks.

In the two-layer feedforward network discussed earlier, all the excitatory neurons in the input layer are connected to each excitatory neuron in the output layer. Therefore, this is not a sparse interconnection structure. In fact, this structure is one of the most commonly used neural network topologies in the real world.

The optimization for three-layer feedforward networks is the same with two-layer feedforward neuron networks. For three-layer feedforward networks, the synapses can

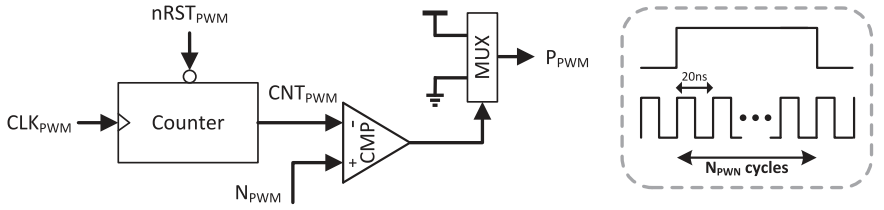

Fig. 10. Digital pulse width modulator: CLK_{PWM} is the 50MHz clock signal for the pulse generator. N_{PWM} is the desired number of clock cycles, which is compared to the output of the counter. The multiplexer outputs the pulse with duration of N_{PWM} clock cycles.

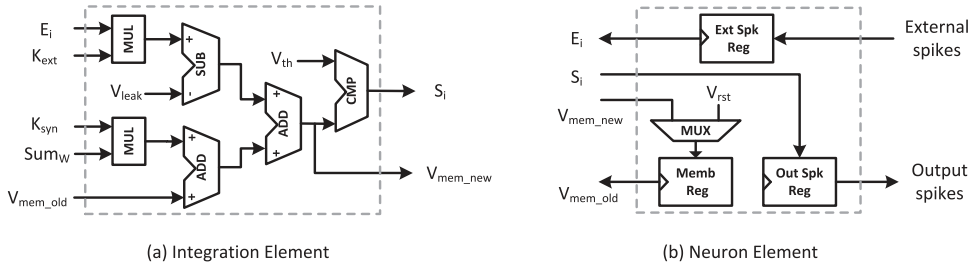

Fig. 11. Data flow of the integration element (IE) and the neuron element (NE). The signal $SumW$ corresponds to the term $\sum_{j=1}^{M} w_{ji} \cdot S_j[t-1]$ in (1), which is calculated by the readout circuits of the synapse unit.

still be divided into two categories, namely the inhibitory synapses and feedforward synapses. The inhibitory synapses are integrated into the digital design as constant values, and the feedforward synapses are stored in two separate small memristor arrays. Therefore, the proposed architecture is scalable for multi-layer feedforward neuron networks.

3.4. The Baseline Building Components of the Proposed DNP Architectures

In order to perform an accurate analysis for the hardware cost of each architecture proposed above, it is necessary to investigate the basic building blocks of different architectures and provide detailed information on each block. The design in Kim et al. [2015] is used as the baseline architecture in this work.

A memristor crossbar array is the central storage for all the architectures. To access the memristor crossbar array, a decoder (i.e., 8-to-256 decoder) is needed. Also, a pulse generator is needed to send out parallel voltage pulses for either the read or write operation. The pulse generator is implemented with an array of counters and comparators, which is illustrated in Figure 10. The digital counters in the pulse generator operate at 50MHz, although the other digital building blocks such as IE, NE, and LE operate at 1MHz. The required pulse width is determined by signal N_{PWN}, which is obtained from the learning unit.

The NU and LU involve arrays of digital processing engines (i.e., NE and LE), so they take up a large portion of the chip area. As discussed in the previous sections, a flash ADC array is necessary for the synapse update of all the architectures, and the corresponding hardware cost is also large.

The LIF arithmetic unit involves either one shared IE or an array of integration elements. If the architecture is based on the shared IE, then the cost of the LIF arithmetic unit itself is trivial, but a huge multiplexer (MUX) will be introduced.

Figure 11 illustrates the design details of IE and NE. As mentioned earlier, the function of IE and NE is to update the membrane potential of each neuron based on

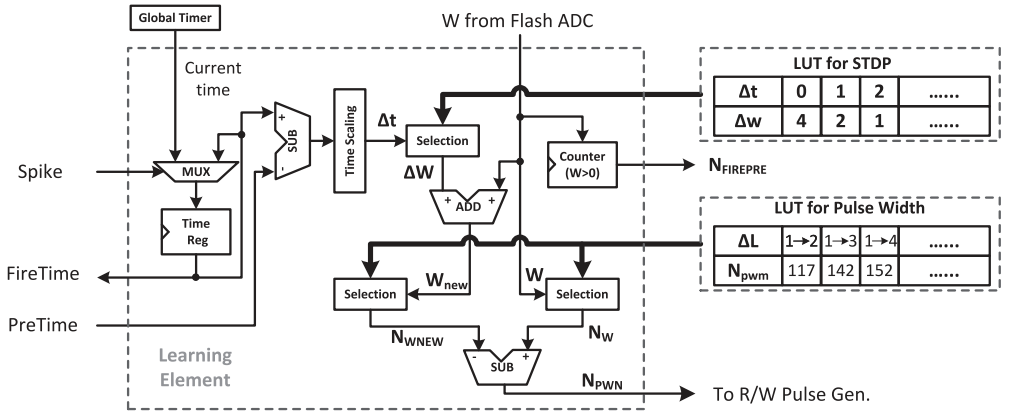

Fig. 12. Data flow of the learning element (LE). Lookup tables (LUTs) are used to calculate ΔW based on the STDP learning rule. Signal N_{PWN} controls the pulse generator to generate the required pulse widths.

(1) and then identify the firing activity. As shown in Figure 11, the IE reads out the membrane potential V_{mem} from the NE and sends the updated value back to the NE. The firing activity (spike) is calculated inside the IE and the spike bit is stored inside the NE.

Figure 12 illustrates the design details of LE. Every time this particular neuron fires, the current value of the Global Timer is recorded by the register *TimeReg*, which will serve as the most recent firing time of this neuron. When this neuron fires as a post-synaptic neuron, the firing times of its pre-synaptic neurons are received from the *PreTime* signal. As a pre-synapse neuron to some other neuron, the firing time of this particular neuron is sent out to its post-synaptic neuron through the *FireTime* signal. The calculation of ΔW based on Δt and the calculation of N_{PWN} for the pulse generator are both realized by lookup tables.

For the columnwise design using the flash ADC array for neuron-stage readout (see Figure 5(a) or (b)), a digital adder tree has to be introduced to realize the summation of the synaptic weights from one column. Although the total number of inputs is large, the adder tree only performs low-precision additions. Therefore, its hardware cost is not very large. According to Figure 5(c), the rowwise design requires an array of low-precision accumulators/adders. Since the proposed DNP works at a frequency range of KHz or MHz, each low-precision adder has a very small hardware cost.

3.5. The Parallel Neuron Integration

The proposed memristor crossbar array also supports parallel access of multiple columns. Therefore, the integration of multiple digital neurons can be performed simultaneously in a columnwise design using multiple IEs. The following figure illustrates an example of such a parallel scheme with a degree of parallelism of 2 (see Figure 13).

As the degree of parallelism is increased, more and more summing amplifiers and high-resolution ADCs are required. The area overhead introduced by this parallel processing scheme is mainly due to the duplicated summing amplifiers and the high-resolution ADCs, so it increases linearly with the degree of parallelism. Of course, the power consumption is also increased accordingly. The update of each individual synaptic weight during on-chip training is still realized by the flash ADC array. Because the hardware cost of the flash ADC array is much larger than that of the column ADCs when the network is large, it would not be very efficient if we duplicate multiple flash ADC arrays to support parallel synaptic weight updates, although it is possible.

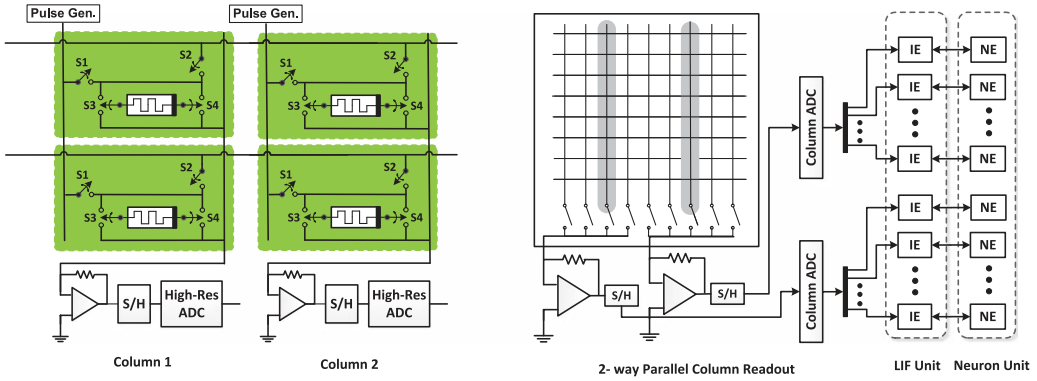

Fig. 13. Parallel processing with two-column ADCs: (a) the detailed connection between memristor cells and pulse generators; (b) the simultaneous access oftwo columns in a design with N digital neurons. Each column ADC accesses N/2 columns sequentially. Two Vmems can be calculated simultaneously.

4. RESULTS

In this section, we analyze the power and area costs of different DNP architectures, which involve different synapse readout schemes and various choices of ADCs. In addition, we also evaluate the performance of the new synapse storage scheme, which targets the mainstream feedforward neuron networks.

All the digital components such as neuron unit and learning unit are designed in the Verilog HDL and synthesized using a commercial 90nm CMOS standard cell library. The analog parts are designed and analyzed with HSPICE. A two-layer feedforward spiking neural network is proposed in this work, which can be configured for character and speech recognition. Inhibitory neurons are added in both the input layer and the output layer, which provide the winner-take-all mechanism for the neural activity.

Four different designs are used for architecture analysis in this work, which access one column or one row at a time. The power consumed by the memristive crossbar is estimated by considering the average memristance and the supply voltage. As mentioned earlier, there are eight different conductance levels for each memristor. The resistance values of level 4 and level 5, which are represented by R_4 and R_5, are used to calculate the average power.

The average power of the memristor is estimated by the following equations:

$$P_{read} = \frac{V_{DD}^2}{R_{read}} \cdot \frac{T_{read}}{T_{period}}, \qquad P_{write} = \frac{V_{DD}^2}{R_{write}} \cdot \frac{T_{write}}{T_{period}}, \tag{8}$$

where V_{DD} is the supply voltage and R_{read} and R_{write} are the resistances for read and write operations, respectively. T_{read} represents the time required by a read operation, while T_{write} represents the time required by a write operation to change the conductance between level 4 and level 5. T_{period} is the main clock period of the DNP. When using the high-resolution column ADC, R_{read} is simply equal to R_4. However, for the low-resolution ADC (i.e., 3-bit Flash ADC), $(R_4 + R_{load})$ is used as R_{read}, where R_{load} is the load resistance connected in series with the memristor to form a voltage divider. For the write operation, the average between R_4 and R_5 is used as R_{write}.

The average power consumptions of memristor crossbar arrays in these four designs are summarized in the following table. The first two designs are both based on the fully reconfigurable designs using $N \times N$ memristor array as synapse storage. However,

Table II. The Power Consumptions of Memristor Crossbar Arrays in Different Designs as Functions of the Size of the Arrays
The application-specific architectures only store the feedforward synapses in the memristor crossbar array.

Architecture	Columnwise Array Access	Rowwise Array Access
256 × 256 fully reconfigurable	1068.6 μW	1068.6 μW
891 × 891 fully reconfigurable	3721.5 μW	3721.5 μW
196 × 36 application specific	818.1 μW	150.3 μW
875 × 9 application specific	3654.7 μW	38.3 μW

Table III. Power and Area of the Baseline Components. NU and LU Represent Neuron Stage and Learning Stage, Respectively

	Power (uW)	Area (μm^2)	Stages
Integration element	88.65	430	NU
256-1 16-bit MUX	3680	24,950	NU
256-input adder tree	36.83	17,111	NU
3-bit accumulator	0.802	347	Rowwise
8-to-256 decoder	50.73	872	Both
Flash ADC array	1,446.4	211,700	LU or both
Learning Unit	968	551,391	LU
Neuron Unit	290	167,208	NU
Pulse Generator	1,079	120,393	Both
System Controller	29.7	19,157	Both
Memristor Array	/	100,489	Both
Pipelined ADC	835	68,600	NU
SAR ADC	639	30,800	NU
SD ADC	110	120,000	NU
VCO ADC	3,610	5,817	NU

the other two designs are based on the application-specific designs that consider only feedforward synapses in the memristor array.

When it comes to the hardware implementation, the neuromorphic chip for character recognition involves 256 digital neurons. Power and area of each baseline component in this work are illustrated in Table III. The powers of the ADCs are obtained by using the ADC power estimator discussed in the previous section. The corresponding areas are estimated using the reference ADC designs presented in Verbruggen et al. [2009] and SDADC, which are also based on a 90nm CMOS process. To estimate the ADC areas with different resolutions, we assume that: (1) The area of the flash ADC is exponential with the resolution, (2) the area of the Pipeline/SAR ADC is linear with the resolution, and (3) the area of Sigma-Delta ADC is linear with the filter order. The clock rate of the pulse generator is 50MHz, while the clock rate of the other digital part is fixed at 1 MHz. According to the control flow of the proposed neuron unit and learning unit, the time consumed in neuron stage and learning stage are 256μs and 512μs, respectively. The total energy consumed for processing all the 256 neurons is calculated from the power of each basic component and the corresponding processing time.

For the columnwise memory access scheme, the flexibility of using different high-resolution ADCs for the neuron stage provides a new way to trade off between energy and area. The power and area values of different ADCs are also summarized in this table.

We conduct a behavior-level digital simulation to demonstrate the functionality of the neuromorphic processors 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. All the key hardware features including the neuron

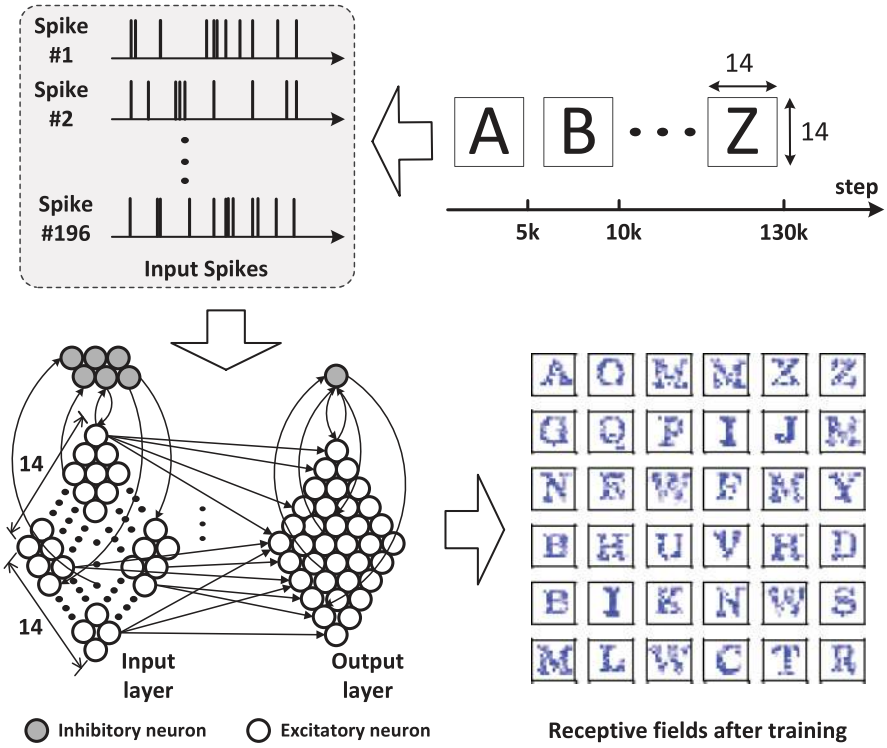

Fig. 14. The two-layer neural network designed for character recognition and the corresponding learning result. Each pixel input pattern is converted into 14 × 14 spike inputs to the input layer of the network.

dynamics and STDP rule are modeled in simulation. The proposed DNP is configured to be a two-layer learning network for character recognition, as illustrated in Figure 14. The network is designed to recognize the alphabets "A"–"Z" by unsupervised learning. Each exitatory input neuron receives a pixel value in the 14 × 14 pixel input pattern and projects its output to all excitatory output neurons through plastic synapses. The receptive fields of the network after the training demonstrate the learning result.

Using the fully reconfigurable 256 × 256 memristor array as the memory, the energy and area results of different architectures are listed in Table IV. According to Table IV, the rowwise memory access scheme has the moderate level of energy and area. But, as mentioned earlier, the neuron stage of the rowwise scheme can be further divided into two operating stages and the instantaneous peak power due to the parallel LIF units in the second operating stage can be large, which is a potential weakness of this readout scheme. The architectures based on the shared IE scheme are more energy consuming than those based on the nonshared IE approach, while their areas are smaller. To achieve a good balance between enery and silicon area, we also take into account the energy-area product (EAP) for each architecture.

As illustrated in Table IV, the designs involving VCO-based ADCs tend to suffer from higher energy consumption, although moderate areas can be achieved. On the contrary, the designs involving the pipelined ADC, SAR, and Sigma-Delta ADC tend to have a much lower energy consumption at the expense of a larger area. The lowest energy consumption is achieved by the design which utilizes Sigma-Delta ADC as the column ADC, although it has the largest area. The smallest area is achieved by the

Table IV. Fully Reconfigurable Designs Using 256x256 Memristor Array as Synapse Storage, Which Can Support Any Network Topology Involving 256 Neurons
Comparison of different architectures in terms of energy, area, and energy-area product (EAP).

Memory access styles	ADC schemes		Energy (μJ)	Area (mm^2)	EAP
Columnwise	Nonshared IE	Pipelined ADC	3.26	1.350	4.40
		SAR ADC	3.21	1.312	4.21
		SD ADC	3.08	1.402	4.32
		VCO ADC	3.97	1.287	5.11
		Flash ADC array	3.42	1.282	4.38
	Shared IE	Pipelined ADC	4.20	1.265	5.31
		SAR ADC	4.15	1.227	5.09
		SD ADC	4.02	1.317	5.30
		VCO ADC	4.91	1.202	5.90
		Flash ADC array	4.35	1.197	5.21
Rowwise	Flash ADC array		3.43	1.299	4.46

Table V. Application-Specific Designs That Store Only Feed-Forward Synapses in the Memristor Array
Comparison of different architectures in terms of energy, area, and energy-area Product (EAP). All designs are based on the nonshared IE scheme.

Memory access styles	ADC schemes	Energy (μJ)	Area (mm^2)	EAP
Columnwise	Flash ADC array	0.955	1.114	1.06
	Pipelined ADC	0.941	1.183	1.11
	SAR ADC	0.934	1.145	1.07
	SD ADC	0.915	1.234	1.13
	VCO ADC	1.041	1.120	1.17
Rowwise	Flash ADC array	0.969	0.91	0.88

design which utilizes flash ADC array for column readout with only one shared IE, but its energy level is high due to the large multiplexer introduced by sharing a single IE.

All the designs discussed so far are based on the fully connected memristor array. This is the most flexible approach because any network with 256 neurons can be supported by the 256×256 synaptic array. However, this storage scheme suffers from bad storage utilization for sparser but more practical network topologies, which leads to a significant waste of energy and silicon area for very large scale neuron networks. To solve this problem, we propose an optimized synapse storage scheme for mainstream feedforward neural networks, which is discussed in detail in Section 3.3. According to Table V, on average, the designs with the new optimized storage scheme of the two-layer feedforward networks consume 70% less energy than the designs using a 256×256 crossbar array. This significant reduction of energy consumption is mainly due to the smaller number of three-bit flash ADCs and a smaller pulse generator, as well as fewer clock cycles to access the memristor array. The new memristor array only takes up 10.7% of the area of the original 256×256 memristor crossbar array. In addition, the optimized storage strategy can achieve up to $5\times$ reduction in the EAP when compared to the fully reconfigurable storage strategy.

In addition to character recognition, the proposed spiking neuron network can also be used for speech recognition. Figure 15 demonstrates the two-layer neural network designed to recognize short audio clips, such as “Two,” “Three,” and “Zero.” To apply the proposed spiking neuron network to speech recognition, the speech signals are converted into speech patterns with 35 frequency domain channels over 25 time units, where stronger signal in this 35×25 pattern corresponds to a higher input spiking rate for the corresponding input-layer neuron (pixel). The corresponding hardware implementation involves 891 digital neurons.

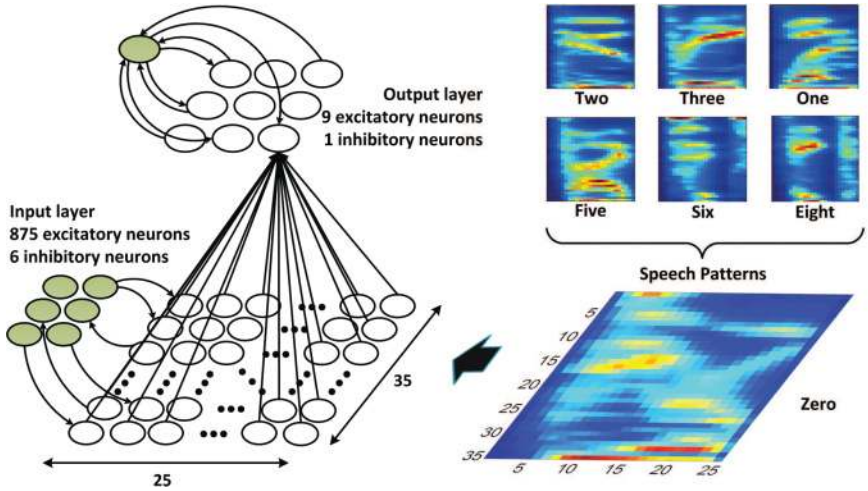

Fig. 15. The two-layer neural network designed for speech recognition. Each speech pattern is converted into 25×35 spike inputs to the input layer of the network.

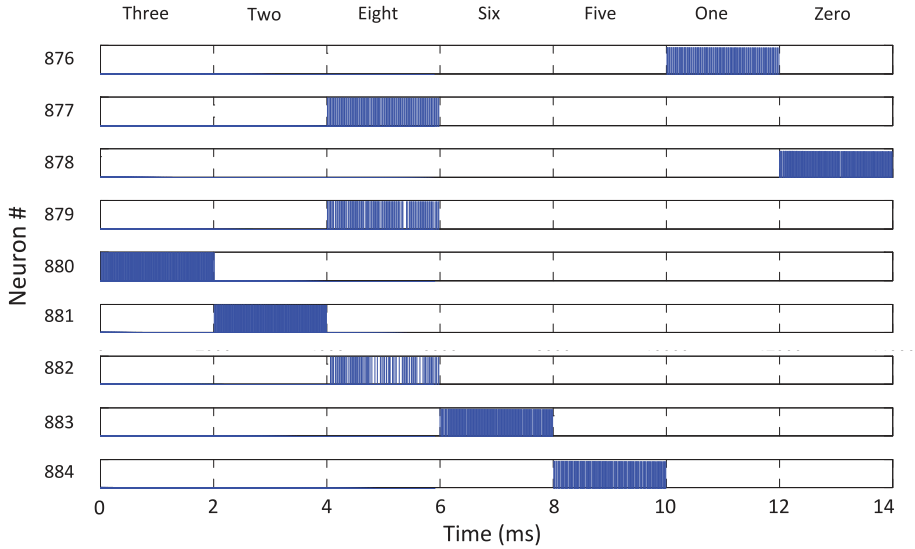

Fig. 16. The spiking events emitted by the output neurons (with neuron index from 876 to 884) as a function of time after training. Each neuron only responds to one particular speech pattern and shows high firing frequency for this speech pattern.

Figure 16 shows the activity of the output layer neurons after learning. Before the training is finished, the speech patterns enter the input layer one by one in random order, and the output layer shows no selectivity to different patterns. After the training, due to the winner-take-all property introduced by the inhibitory neuron, each output layer neuron only responds to one particular speech pattern. For example, the output layer neuron labeled 880 shows a high firing frequency for "Three," but it does not respond to any other input patterns.

For the hardware implementation of this spiking neural network with 891 neurons, the power and area of each baseline building component are listed in Table VI. The

Table VI. Power and Area of the Baseline Components
NU and LU represent the neuron stage and the learning stage, respectively. Since there are 891 neurons in this network, the resolution of the column ADC is changed to 13 bits.

	Power (μW)	Area (μm^2)	Stages
Integration element	88.65	430	NU
10-to-1024 decoder	203.09	3,519	Both
891-input adder tree	125.12	56,980	NU
Three-bit accumulator	0.802	347	Rowwise
Flash ADC array	5,034.15	736,815	LU or both
Learning Unit	3,370.10	1,919,099	LU
Neuron Unit	1,009.36	581,962	NU
Pulse Generator	3,755.42	419,025	Both
System Controller	29.7	19157	Both
Memristor Array	/	1,217,289	Both
Pipelined ADC	904	74,360	NU
SAR ADC	693	33,300	NU
SD ADC	110	120,000	NU
VCO ADC	5,630	10,200	NU

Table VII. Fully Reconfigurable Designs Using a 891x891 Memristor Array for Synapse Storage
Comparison of different architectures in terms of energy, area, and energy-area product (EAP). All designs are based on the nonshared IE scheme.

Memory access styles	ADC schemes	Energy (μJ)	Area (mm^2)	EAP
Columnwise	Flash ADC array	41.06	5.28	216.78
	Pipelined ADC	37.38	5.36	200.34
	SAR ADC	37.20	5.31	197.80
	SD ADC	36.67	5.40	197.64
	VCO ADC	41.59	5.29	219.99
Rowwise	Flash ADC array	41.88	5.30	221.95

energy and area results of different architectures are listed in Table VII. The architectures in Table VII are all based on the nonshared IE scheme, considering that the shared-IE scheme suffers from higher power due to the huge multiplexer introduced. Both the feedforward synapses and the synapses involving inhibitory neurons are stored in the memristor crossbar array, and they are processed in the same manner. These architectures are fully reconfigurable, which can support any neural network topology.

Figure 17 shows the synapse distribution of a conceptual 891×891 synaptic array. The number of the input-layer neurons (pixels) is much larger than that of the output layer neurons. This is a very common situation for two-layer feedforward neuron networks, because high resolution of the input pattern is required while the total number of the patterns to be recognized is usually limited.

As illustrated in Figure 17, the feedforward synapses only exist in a narrow region inside the 891×891 synaptic array. Obviously, it would be a huge waste of hardware resource and processing cycles if the fully reconfigurable approach storing all 891×891 synapses was applied to such networks. When mapping this neural network to the proposed neuromorphic processors, the energy and area results can be obtained, as shown in Table VIII.

The fully reconfigurable synapse storage approach uses a 891×891 memristor array as storage, and each synapse in this array has to be accessed once for a single training iteration. However, according to Figure 17, there are only 9 columns and 875 rows that are associated with the feedforward synapses, so the optimized storage approach

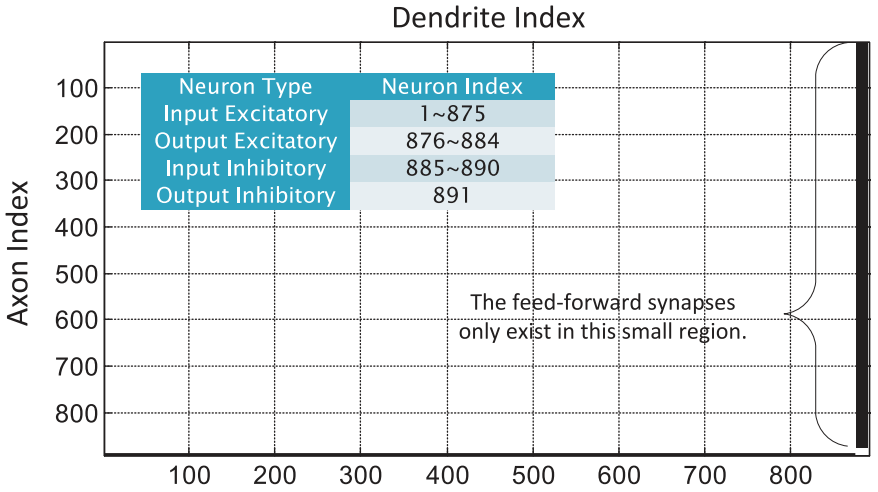

Fig. 17. The synapse distribution of the conceptual 891×891 synaptic array. Since there are only 9 excitatory neurons in the output layer and 875 excitatory neurons in the input layer, the feedforward synapses only exist in a very small region.

Table VIII. Application-Specific Designs That Only Update the Feedforward Synapses between the Two Layers

Comparison of different architectures in terms of energy, area, and energy-area product (EAP). All designs are based on the nonshared IE scheme.

Memory access styles	ADC schemes	Energy (μJ)	Area (mm^2)	EAP
Columnwise	Flash ADC array	7.94	4.05	32.15
	Pipelined ADC	7.90	4.13	32.62
	SAR ADC	7.90	4.09	32.30
	SD ADC	7.89	4.17	32.89
	VCO ADC	7.94	4.07	32.31
Rowwise	Flash ADC array	7.86	2.96	23.26

considering only feedforward synapses only needs to access 875 × 9 synapses for a single training iteration. Therefore, it has a much smaller energy consumption than the fully reconfigurable approach.

Updating only the feedforward synaptic weights is actually application-specific optimization, which works well for all the feedforward neural networks, as described in Section 3.3. In addition, for this particular neural network, if we choose a rowwise memory access style over a columnwise memory access style, then the number of flash ADCs can be reduced to 9 from 875, while the processing cycles will increase from 9 to 875. Therefore, the energy consumption will not changevery much, but the rowwise scheme introduces significant area reduction.

What needs to be noted here is that the rowwise scheme shows better results in Tables V and VIII, only because the number of output layer neurons is much smaller than that of the input layer neurons. If there are much more output layer neurons than the input layer neurons, then the columnwise scheme will become the better choice.

5. CONCLUSION

In this article, we have proposed two memory access styles for the memristor synaptic array-based DNP architectures. The architectures with various synaptic weight readout strategies and possible ADC schemes are thoroughly investigated, which provides

new insights into the tradeoff between energy and chip area of DNPs. In addition, a novel storage strategy optimized for mainstream feedforward spiking neural networks is presented, which proves to significantly improve the energy efficiency as well as the utilization of the memristive synaptic array.

REFERENCES

John V. Arthur, Paul A. Merolla, Filipp Akopyan, Rodrigo Alvarez, Andrew Cassidy, Shyamal Chandra, Steven K. Esser, Nabil Imam, William Risk, Daniel B. D. Rubin, and others. 2012. Building block of a programmable neuromorphic substrate: A digital neurosynaptic core. In *Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

Ling Chen, Chuandong Li, Tingwen Huang, Yiran Chen, and Xin Wang. 2014. Memristor crossbar-based unsupervised image learning. *Neur. Comput. Appl.* 25, 2 (2014), 393–400.

Pieter J. A. Harpe, Cui Zhou, Yu Bi, Nick P. van der Meijs, Xiaoyan Wang, Kathleen Philips, Guido Dolmans, and Harmke De Groot. 2011. A 26 W 8 bit 10 MS/s asynchronous SAR ADC for low energy radios. *IEEE J. Solid-State Circ.* 46, 7 (2011), 1585–1595.

Yenpo Ho, Garng M. Huang, and Peng Li. 2009. Nonvolatile memristor memory: Device characteristics and design implications. In *Proceedings of the 2009 International Conference on Computer-Aided Design*. ACM, New York, NY, 485–490.

Miao Hu, Hai Li, Qing Wu, and Garrett S. Rose. 2012. Hardware realization of BSB recall function using memristor crossbar arrays. In *Proceedings of the 49th Annual Design Automation Conference*. ACM, New York, NY, 498–503.

Yen-Chuan Huang and Tai-Cheng Lee. 2011. A 10-bit 100-MS/s 4.5-mW pipelined ADC with a time-sharing technique. *IEEE Trans. Circ. Syst. I: Regul. Pap.* 58, 6 (2011), 1157–1166.

Zhaohui Huang and Peixin Zhong. 2004. An architectural power estimator for analog-to-digital converters. In *Null*. IEEE, 397–400.

Giacomo Indiveri, Elisabetta Chicca, and Rodney Douglas. 2006. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neur. Networks* 17, 1 (2006), 211–221.

Giacomo Indiveri, Bernabé Linares-Barranco, and Teresa Serrano-Gotarredona. 2011. Neuromorphic silicon neuron circuits. Frontiers Research Foundation.

Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B. Bhadviya, Pinaki Mazumder, and Wei Lu. 2010a. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 4 (2010), 1297–1301.

Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B. Bhadviya, Pinaki Mazumder, and Wei Lu. 2010b. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 4 (2010), 1297–1301.

Kuk-Hwan Kim, Siddharth Gaba, Dana Wheeler, Jose M. Cruz-Albrecht, Tahir Hussain, Narayan Srinivasa, and Wei Lu. 2011. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Lett.* 12, 1 (2011), 389–395.

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

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 (2015), 38.

Boxun Li, Lixue Xia, Peng Gu, Yu Wang, and Huazhong Yang. 2015. Merging the interface: Power, area and accuracy co-optimization for RRAM crossbar-based mixed-signal computing system. In *Proceedings of the 52nd Annual Design Automation Conference*. ACM, New York, NY, 13.

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

Paul Merolla, John Arthur, Filipp Akopyan, Nabil Imam, Rajit Manohar, and Dharmendra S. Modha. 2011. A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. In *Proceedings of the 2011 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 1–4.

Srinjoy Mitra, Stefano Fusi, and Giacomo Indiveri. 2009. Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *IEEE Trans. Biomed. Circ. Syst.* 3, 1 (2009), 32–42.

B. Murmann. 1997. ADC performance survey. In *ISSCC & VLSI Symposium*, Vol. 2013.

J. S. Seo, et al. 2011. A 45nm CMOS Neuromorphic Chip with a Scalable Architecure for Learning in Networks of Spiking Neurons. CICC, pp. 1–4.

Pradeep Shettigar and Shanthi Pavan. 2012. A 15mW 3.6 GS/s CT-$\delta\sigma$ ADC with 36MHz bandwidth and 83dB DR in 90nm CMOS. In *Proceedings of the 2012 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 156–158.

Greg S. Snider. 2008. Spike-timing-dependent learning in memristive nanodevices. In *Proceedings of the IEEE International Symposium on Nanoscale Architectures, 2008 (NANOARCH'08)*. IEEE, 85–92.

André van Schaik. 2001. Building blocks for electronic spiking neural networks. *Neur. Networks* 14, 6 (2001), 617–628.

Bob Verbruggen, Jan Craninckx, Maarten Kuijk, Piet Wambacq, and Geert Van der Plas. 2009. A 2.2 mW 1.75GS/s 5 bit folding flash ADC in 90nm digital CMOS. *IEEE J. Solid-State Circuit.* 44, 3 (2009), 874–882.

Qian Wang, Yongtae Kim, and Peng Li. 2014. Architectural design exploration for neuromorphic processors with memristive synapses. In *Proceedings of the 2014 IEEE 14th International Conference on Nanotechnology*. IEEE, 962–966.

Received September 2015; revised January 2016; accepted February 2016