Breaking Through the Speed-Power-Accuracy Tradeoff in ADCs Using a Memristive Neuromorphic Architecture

Loai Danial ⬤ , *Student Member, IEEE*, Nicolás Wainstein ⬤ , *Student Member, IEEE*, Shraga Kraus ⬤ , and Shahar Kvatinsky ⬤ , *Senior Member, IEEE*

Abstract—The analog-to-digital converter (ADC) is a principal component in every data acquisition system. Unfortunately, modern ADCs tradeoff speed, power, and accuracy. In this paper, novel neuroinspired approaches are used to design a smart ADC that could be trained in real time for general purpose applications and break through conventional ADC limitations. Motivated by artificial intelligent learning algorithms and neural network architectures, the proposed ADC integrates emerging memristor technology with CMOS. We design a trainable four-bit ADC with a memristive neural network that implements the online gradient descent algorithm. This supervised machine learning algorithm fits multiple application specifications such as full-scale voltage ranges and sampling frequencies. Theoretical analysis, as well as simulation results, demonstrate highly powerful collective properties, including reconfiguration, mismatch self-calibration, adaptation to dynamic voltage and frequency scaling, noise tolerance, and power consumption optimization. The proposed ADC achieves 8.25 fJ/conv FOM, 3.7 ENOB, 0.4 LSB INL, and 0.5 LSB DNL. These promising properties make it a leading contender for general purpose and emerging data driven applications.

Index Terms—Analog-to-digital conversion, adaptive systems, calibration, computational intelligence, energy efficiency, memristors, neuromorphic computing, reconfigurable architectures, supervised learning.

I. INTRODUCTION

THE rapid evolution of data-driven systems towards the internet of things era has paved the way to emergent interacting and varying applications where data converters are ubiquitous. With the advent of high-speed, high-precision, and low-power mixed-signal systems, there is an ever-growing demand for accurate, fast, and energy-efficient data converters. These systems operate on a broad range of real-world continuous-time signals; examples include medical imaging, biosensors, wearable devices, consumer electronics, automotive, instrumentation, and telecommunication [1].

Unfortunately, the intrinsic speed-power-accuracy tradeoff in analog-to-digital converters (ADCs) is pushing them out of the application band of interest [2], [3]. Furthermore, with the nonstop downscaling of technology motivated by Moore's law, this tradeoff has become a chronic bottleneck of modern systems design due to alarming deep sub-micron effects [4], [5]. Those effects are poorly handled with particular design techniques that overload data converters with tremendous overhead, exacerbating the tradeoff and degrading their performance dramatically [5]. Nowadays data converters lack design standards and are customized with sophisticated specific design flow and architectures for special purpose applications.

This paper takes a different approach to design general purpose ADCs. We propose that the converted data be used to train the converter in order to autonomously adapt to the exact specifications of the running application as well as to adjust to environmental variations. This approach will reduce the converter's time to market, efficiently scale with newer technologies, drastically reduce its cost, standardize the design flow, and enable a generic architecture for general purpose applications.

The proposed trainable ADC utilizes machine learning (ML) algorithms to train an artificial neural network (ANN) architecture [6] based on the promising technology of memristors. Memristors are now being widely adopted in the design of synapses for artificial neural systems [7], [8] because of their small footprint, analog storage properties, energy efficiency, and non-volatility. These characteristics allow for synapse-like behavior, where the conductance of the memristor is considered as the weight of the synapse [9]. We leverage the use of memristors as synapses to achieve high-precision, high-speed, low-power, a simple cost-efficient, and reconfigurable single-channel ADC architecture that breaks through the speed-power-accuracy tradeoff. The design methodologies are based on our previous work on online training of memristive synapses [10] and on digital-to-analog converter (DAC) [11].

The remainder of this paper is organized as follows. In Section II, we explain the motivation behind our approach. In Section III, the proposed ADC architecture, theory, and training algorithm are described. In Section IV, circuit design and mechanisms of a four-bit ADC are detailed. In Section V, the circuit operation and learning capability are evaluated. In Section VI,
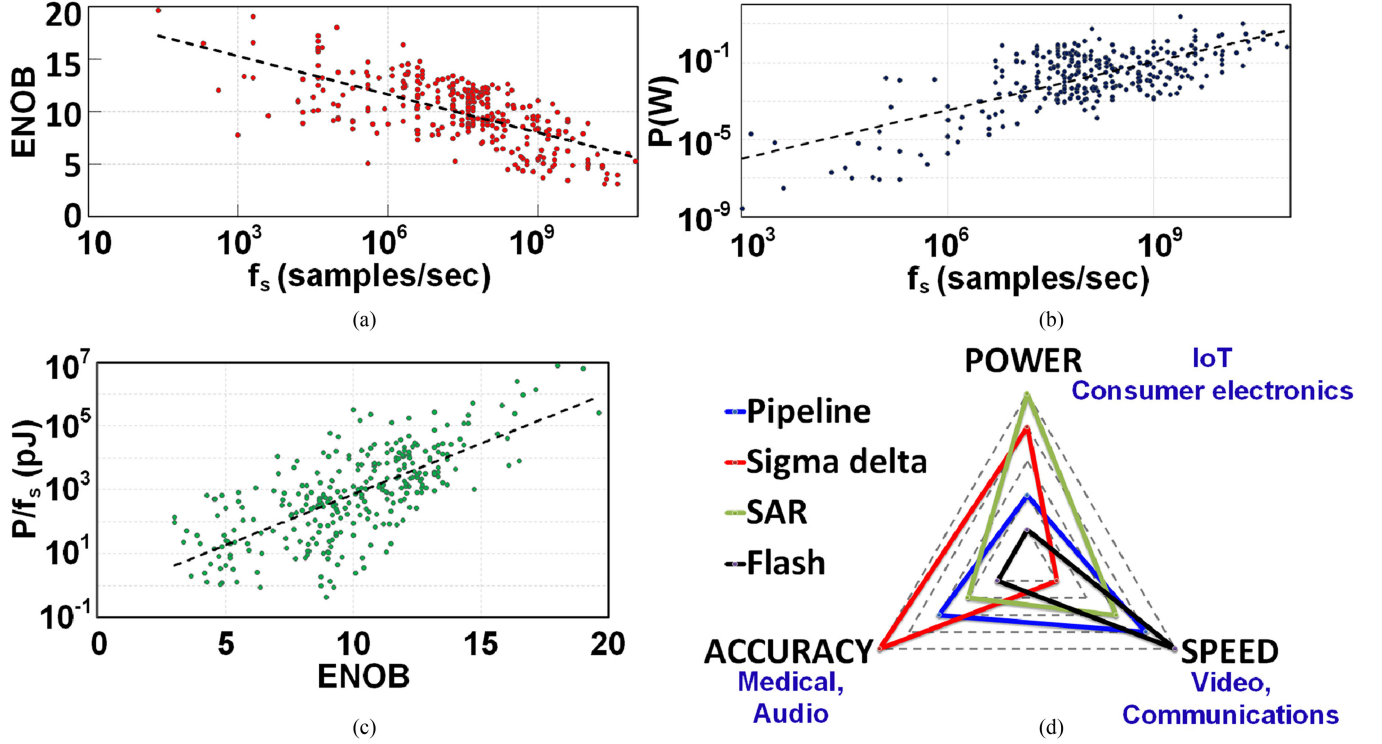
Fig. 1. Tradeoffs in conventional ADC architectures between (a) speed and accuracy, (b) speed and power, (c) accuracy and energy, as reported in [14]. Since the power-accuracy tradeoff depends on the limitations of the underlying architectures, the energy-accuracy is independent of the architecture and shows the tradeoff accordingly. (d) Spider diagram of ADC architectures (different color lines), design tradeoff, and associated applications (in blue).

design trade-offs and large-scale challenges are discussed. We conclude in Section VII.

II. Motivation

A. Speed-Power-Accuracy Tradeoff in ADC Architectures

While the analog domain is mainly characterized by its energy efficiency in data processing, its digital counterpart outperforms it in reliable computation [12]. ADCs are mixed-signal systems that inherently combine hybrid analog-digital principles along with the *pros* and *cons* of each domain. Therefore, these systems are optimally customized to fit a specific subset from a wide functional spectrum. Design tradeoff is an extreme case when the system is pushed toward its performance limits. The ADC comprises a signal sampler that discretely samples the continuous-time signal at a constant rate, and a quantizer that converts the sampled value to the corresponding discrete-time *N*-bit resolution binary-coded form. The quality of a system is considered ideal when it achieves high speed and accuracy with a low power drain. In practice, however, the resolution decreases as the conversion rate increases, and greater power consumption is required to achieve the same resolution.

Device mismatch is the dominant factor affecting system accuracy [4]. Larger devices are necessary to improve system accuracy, but the capacitive loading of the circuit nodes increases as a result and greater power is required to attain a certain speed. The maximal speed of the system is a function of the gain-bandwidth, but it is limited by the input pole. Aside from device mismatches, four loss mechanisms affect the ADC resolution and limit the *signal-to-noise-and-distortion ra-tio* (SNDR): quantization noise, jitter, comparator ambiguity, and thermal noise.

Quantization noise is the only error in an ideal ADC. Jitter is a sample-to-sample variation of the instant in time at which sampling occurred. Additionally, the conversion speed is limited by the ability of the comparator to make assertive decisions regarding the relative amplitude of the input voltage [3]. This limitation is called comparator ambiguity and it is related to the speed of the device used to fabricate the ADC. Device speed is measured as the frequency, f_T, at which there is unity current gain. As a result of these limitations, approximately one bit of resolution is lost each time the sampling rate doubles [3].

Whereas non-linear distortions, memory effects, and device mismatches can be somewhat compensated for, thermal white noise cannot; consequently, it is one of the more dominant limiters of ADC performance. It is modeled by *KT/C* noise, where *K* denotes Boltzmann's constant, *T* denotes temperature, and *C* denotes sampler capacitance. Lowering the noise floor by a factor of two in purely thermal-noise limited circuits would quadruple the power consumption [13]. The limit that device mismatch imposes on the power consumption is approximately two orders of magnitude higher than the limit imposed by thermal noise [4]. The speed-power-accuracy tradeoff is illustrated in Fig. 1(a)–(c); it is based on data that we have processed from Stanford's ADC survey [14], which includes papers published during the last two decades.

The need to digitize so many signal types has produced a broad range of data converters diverse in their resolution, sampling rates, and power consumption budget. These considerations profoundly affect system architectures and their performance. The speed-power-accuracy tradeoff has resulted in a

wide range of ADC architectures optimized for special purpose applications, from high-speed, to high-resolution, to low-power applications. Fig. 1(d) specifies the widely used ADC architectures, each mapped to its market applications, on the basis of data collected from [14].

B. ADC Figure-of-Merit (FOM)

When comparing ADCs with different specifications, a numerical quantity known as a figure of merit (FOM) is used to characterize the performance of each ADC relative to its alternatives. Two or more metrics can be combined into a single *FOM* that accurately reflects the merits of the ADC in a certain context and for a specified purpose. One of the most widely used FOMs [3] is defined as

$$FOM = \frac{P}{2^{ENOB} \cdot f_s} \left[\frac{J}{conv} \right], \qquad (1)$$

and relates the ADC power dissipation during conversion, *P,* to its performance in terms of sampling frequency, f_s, and effective number of resolution bits *(ENOB)*. Lower *FOM* values will result in better ADC performance. The *ENOB is* calculated from the SNDR as

$$ENOB = \frac{SNDR\,(dB) - 1.76}{6.02}. \qquad (2)$$

The aforementioned *FOM* best captures the fundamental speed-power-accuracy tradeoff [15]. The ongoing saga of CMOS technology trends toward smaller transistor dimensions has resulted thus far in ultra-deep submicron transistors [16]. The *FOM* evolution also best describes Moore's law of ADCs. Technology scaling improves sampling frequencies, because f_T allows for faster operation [5]. However, the speed of sampling frequency is limited by the comparator ambiguity. In the same context, the impact of technology scaling on power dissipation optimization is also limited by the supply voltages, and by leakage currents that inevitably lead to an increase in the power consumption required to maintain SNDR [16]. These limitations, along with manufacturing process variations and device mismatches in ultra-deep submicron technologies, are the biggest obstacle to achieving high linearity, wide dynamic range, and high-resolution converters [16]. Thus, the speed-power-accuracy tradeoff is becoming dramatically more severe with technology downscaling, pushing future data converters out of the application band of interest [2], [4], [5].

The FOM evolution is shown in Fig. 2 and is based on data collected from [14] that we have analyzed. The figure shows an overall improvement in the FOM over the technology nodes. This improvement is due to low-resolution converters that benefit from technology scaling. However, the improvement has slowed down significantly and ADC performance has recently saturated, as anticipated in [5]. The noise-floor has saturated during the last decade, indicating that *future ADCs could very well fail to maintain even the current state-of-the-art in noise performance* [5].

C. Trainable ADC for General Purpose Applications

Techniques for circumventing the tradeoff have recently been investigated, with the goal of achieving ultra-low-power con-
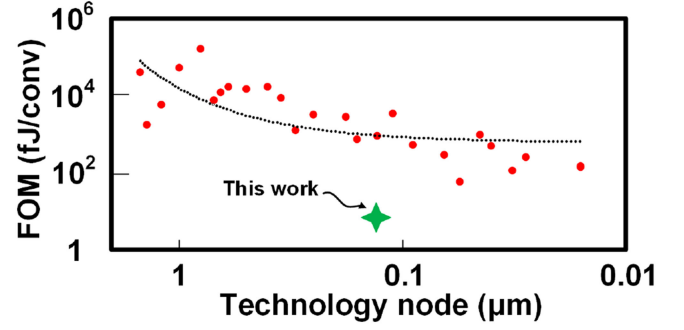

Fig. 2.   Average FOM evolution versus technology node scale-down of the different ADC architectures and specifications shown in Fig. 1 and reported in the ADC survey [14]. Overall, the FOM improves with the technology scale-down. However, the asymptotic slowdown in the last decade is shown by the trendline. The green star shows the achieved FOM of this work.

suming converters with high resolution through a combination of systematic, architectural and technological approaches. Examples of such methods are digitally assisted background calibration, time-interleaving, pipelining, subranging, folding, interpolating, and oversampling [13], [16], [17]. These techniques have succeeded to postpone the FOM saturation.

Modern ADC architectures are custom designed circuits that are fine-tuned to optimize specific capabilities and design parameters up to the application's specification. Widely used methods are sophisticated, specific, and technology dependent, lacking standard design flow. These methods require exhaustive characterization, massive validation, and relatively long development time-to-market. Furthermore, a rapid increase in multi-channel ADCs has recently been observed. Multiple channels are monolithically integrated for diversity-based applications, increasing the total area, cost, design complexity and power consumption [18].

In the same context, reconfigurable architectures that dynamically select between a narrow range of different predefined design specifications have been developed [19], [20]. In contrast, minimalistic design approaches have been proposed to improve power efficiency and potentially increase speed by utilizing simplified analog sub-circuits [13]. Future collective improvements in the ADC *FOM* will most probably be derived from a combination of factors that will include novel architectures, an emerging technology device beyond CMOS, and a systematic approach beyond Moore's law.

The field of machine learning (ML) is devoted to the study and implementation of systems capable of learning from data using their evolving perceptual ability to make crucial decisions, predictions, and classifications based on examples learned from the past. Data conversion could be viewed as a special case of the classification optimization and signal restoration problem that could easily be solved using ML to learn from the data.

We propose a trainable ADC architecture for general purpose applications, as shown in Fig. 3. In our system, a set of parameters is determined to meet the requirements of the running application. First, the sampling frequency f_s is determined, followed by the number of resolution bits N, followed by the full-scale voltage V_{FS}, which specifies the ADC input dynamic range. Then, the ADC is trained by a ML algorithm
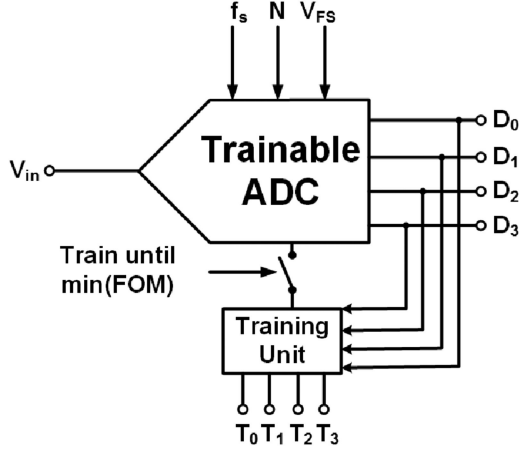
Fig. 3.   Scheme of trainable 4-bit ADC receives f_s, V_{FS}, N and is trained in real-time by providing a specific teaching dataset T_i. The training continues until the ADC achieves the optimal FOM.

in real-time to optimize the ENOB and power dissipation. This procedure is equivalent to a dynamic FOM optimization, which will be proven in this work to potentially achieve a much lower FOM (marked by a green star) than the trend-line in Fig. 2. The technique is not exclusive to reconfiguration, but can also be applied for device mismatch self-calibration, adaptation, and noise tolerance, using generic, standard methodology [10], [11]. Furthermore, the trainability of the architecture adds flexibility that makes it cost-effective and versatile, with a minimalistic design that uses one channel and an intelligent ML algorithm.

III. NEURAL NETWORK ADC

Neuromorphic computing [6] is a mixed-signal design that inherently combines both analog and digital domains in its molecular, biophysical, behavioral, and functional abstraction levels. Extrapolating from electronics to neurobiology, the authors of [21] concluded that the brain computes efficiently in a hybrid fashion. Analogously, we propose to interpolate perceptual abilities from neurobiology to mixed-signal electronics to break through the derived design tradeoffs and utilize the advantages of both domains.

ANNs are receiving widespread attention as potential new architectures and model implementations for a diverse assortment of problems, such as pattern classification, object recognition, and signal processing [22]. Furthermore, ANNs are considered an efficient abstract platform for ML algorithms and big-data interpretation. The massively parallel processing power of the neural network lies in the cooperation between highly interconnected computing elements (neurons), connected by long-term memory elements (synapses). Furthermore, the trainable and adaptive capabilities of ML algorithms are considered novel intelligent features providing an impetus in specific areas where conventional computers perform poorly compared to our brain.

In this section, we propose a neural network ADC paradigm. We show its architecture, fundamentals, theory, and an ML algorithm to train the network.

A. Architecture

ANN architectures are distributed networks that collectively make decisions based on the adjustment of successive approximation weights. Strikingly, this mechanism precisely describes ADCs in time-scale with successive binary-weighted approximation, such the SAR ADC [23]. While bit comparison is equivalent to neural activation, each reference scale during the successive binary search algorithm is equivalent to a binary-weighted synapse. As a first step, we start with transforming the temporal binary search algorithm of 4-bit SAR conversion to a spatial neural network with binary-weighted synapses and pipelined [17] forward propagated neurons (MSB to LSB),

$$\begin{cases} D_3 = u\left(V_{in} - 8V_{ref}\right) \\ D_2 = u\left(V_{in} - 4V_{ref} - 8D_3\right) \\ D_1 = u\left(V_{in} - 2V_{ref} - 4D_2 - 8D_3\right) \\ D_0 = u\left(V_{in} - V_{ref} - 2D_1 - 4D_2 - 8D_3\right) \end{cases}, \quad (3)$$

where V_{in} is the analog input and $D_3 D_2 D_1 D_0$ is the corresponding digital form ($i = 3$ is the MSB), and each bit (neuron product) has either zero voltage or full-scale voltage. $u(\cdot)$ is denoted as the signum neural activation function, and V_{ref} is a reference voltage equal to the smallest discrete voltage quantum (LSB). Each neuron is a collective integrator of its inputs. The analog input is sampled and successively (by a pipeline) approximated by a combination of binary-weighted inhibitory synaptic connections.

The approximation procedure of determining each bit in the ADC is modular. The MSB voltage D_3 can first be determined independently of other bits by comparing to the middle of the full-scale voltage. When D_3 is known, it is bypassed to the second MSB, which can be found regardless of $D_1 D_0$. If D_3 is '1', then D_2 is compared to three-quarters of the full-scale; otherwise it is compared to one-quarter of the full-scale. Analogously, the LSBs are approximated based on the driving MSBs. The successive approximation flow is described by a binary-search tree with all the possible combinations, as shown in Fig. 4(a). Each neuron makes a decision, which takes t_d, and forwardly drives other neurons in an asynchronous pipeline and with strength (synaptic weight) proportional to its significance degree, during the read cycle after propagation time t_p. The total propagation time should be less than the read cycle duration.

In a real-time operation where non-ideal, non-linear, stochastic, and varying conditions affect the conversion accuracy, the correct weights are not distributed deterministically in binary-weighted style as in (3). In this case, the weights should be updated in real-time *in situ* by a training feedback. Four different binary-weighted weights are needed to implement a 4-bit ADC, and 2^4 different precise weights around each binary-weighted weight are required to fine-tune the LSB neuron. The interconnected synaptic weights of the network are described by a matrix W, and each element W_{ij} represents the weight of the connection from pre-synaptic neuron j to post-synaptic neuron i. The neural network ADC architecture including its building blocks (neurons, synapses, and feedbacks) is illustrated in Fig. 4(b).
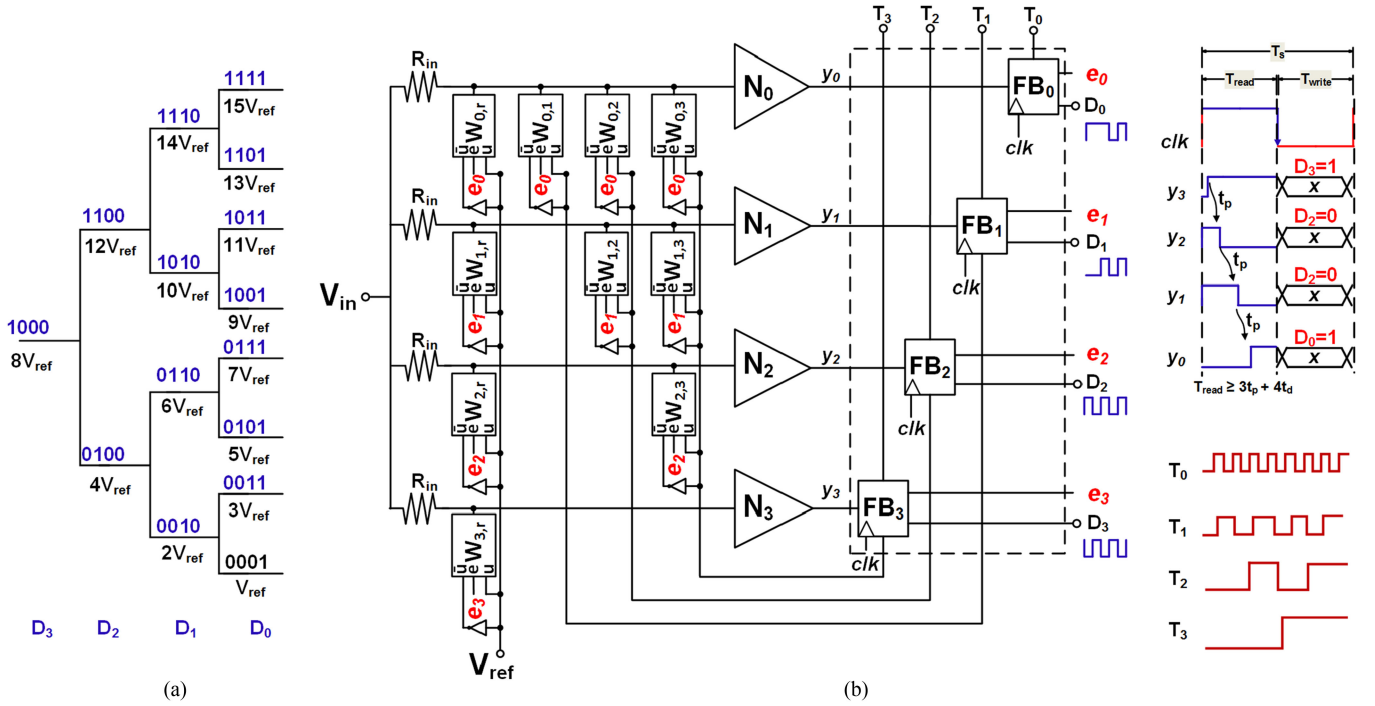
Fig. 4. (a) Successive approximation flow of the SAR-like neural network using a binary-search algorithm. (b) Neural network 4-bit ADC architecture including synapses $W_{i,j}$, neurons N_i, and feedbacks FB_i, in addition to a wave diagram of the neural activity forward propagation among bits. The propagation time of neural decisions should be less than the read cycle. The digital outputs D_i are sampled at the read-cycle end, and then are latched for the write cycle to compare with the teaching data-set T_i, which corresponds to the analog input ramp. Read and write dependent signals are marked in blue and red, respectively.

B. Theory

Surprisingly, the proposed architecture is equivalent to a well-studied architecture with emergent collective computational properties [24]. A simple single-layer neural network was developed from a complex Hopfield neural network [25]. The originally proposed Hopfield network is considered a sub-type of recurrent neural networks with a parallel single layer that comprises fully-connected neurons with inhibitory feedbacks, bidirectional data traversal, and without auto-feedback. From a design point of view, a Hopfield network with symmetric connections is especially convenient for the ADC task [25]–[27]. Most interestingly, the ADC is based on an energy function that describes the macroscopic dynamics of the network. The energy function characterizes the energy minimization process and recursive convergence of the network from an initial state to a minimum energy in steady state [25]. The energy function is used as a network cost function customized for solving specific optimization problems. By defining the energy function, one can easily extract the corresponding weights that fit the network specifications and application demands.

Hopfield networks suffer, however, from several drawbacks that limit their use for practical applications. Due to the complex nature of the energy function, the solution of this symmetric network is highly dependent on its initial state. The energy function might decrease and then settle to one of the equilibrium points, called "spurious state" or "local minima," that does not correspond to the correct digital representation of the input signal and results in ADC characteristics that are far from ideal. Fortunately, these non-linearities can be eliminated using a modified Hopfield network with an additional self-correcting logic network and an extended resistive network [28], [29]. An-

other elimination technique is to use separate electronics that force the neurons to reset, alternately limiting the operational frequency and ADC speed [25], [30], [31]. Moreover, Hopfield networks also suffer from structural shortcomings, especially at a large scale: a large number of synapses, a high ratio between weights, and quantization errors. Recently, a level-shifted 2-bit Hopfield based ADC quantizer was proposed to overcome the original Hopfield network scaling shortcomings and eliminate the digital error that grows along with the number of bits [32].

Our proposed ADC architecture is equivalent to a particular class of asymmetric Hopfield-type networks. It has been designed to overcome the Hopfield network drawbacks and stability issues [24]. The equilibrium point is globally attractive, globally asymptotically stable and guaranteed [24]; that is, the system will converge toward this point for every choice of initial condition and for every choice of non-linearities. Furthermore, neural networks with lower block triangular interconnection structure for robust ADC application have been widely explored in the literature [33]–[38], including their mathematical justification, formalization, qualitative analysis, quantitative asymptotic constraints for stability, encoding techniques, and synthesis. Analogously to the Hopfield energy function, we describe the energy function of the proposed asymmetric architecture as

$$E = -\sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} W_{ij} D_i D_j - \sum_{i=0}^{N-1} D_i \left(V_{in} W_{i_{in}} + V_{ref} W_{i_r} \right), \quad (4)$$

where W_{ij} is a synapse (conductance) between a pre-synaptic neuron with index j and digital voltage D_j, and a post-synaptic neuron with index i and digital voltage D_i, as shown in Fig. 4(b).

The derivative of E according to D_i, which is equivalent to the inverting sum of neuron i input currents, is negative. Thus,

E is a monotonically decreasing function and achieves minimal value when D_i changes to guarantee a zero total current over the whole ramp input. The first component refers to the power dissipation of the interconnected synapses, taking the network asymmetry into consideration (j counts from i). The second component refers to the external dissipated power composed of the analog input voltage source and the reference voltage source. The strategy employed in creating (4) is to consider the ADC as an optimization problem implemented by the following error function E_Q, which is formalized analogously as

$$E_Q = \frac{1}{2}\left(V_{in} - \sum_{i=0}^{N-1} D_i 2^i\right)^2 - \frac{1}{2}\sum_{i=0}^{N-1}\left(2^i\right)^2\left[D_i\left(D_i - 1\right)\right], \quad (5)$$

where the first component is the power of the quantization error. It will achieve minimal value when the digital code corresponds to the correct analog input. The second component is added to eliminate diagonal elements (self-feedback), and its value is always zero. By reordering (5) as an energy-like function, similarly to (4) we get

$$E_Q =$$
$$2^N \cdot \left[-\sum_{i=0}^{N-1}\sum_{j=i+1}^{N-1}\left(-2^j\right)D_i D_j - \sum_{i=0}^{N-1}D_i\left(V_{in} - 2^{i-1}\right)\right], \quad (6)$$

where 2^N is a constant and does not affect the optimal weights for the ADC network. We extract the weights by comparing (6) to (4): $W_{ij(j>i)} = -2^j, W_{ij(j\leq i)} = 0, W_{i_{in}} = 1, W_{i_r} = -2^{i-1}$. These values are typical for a deterministic ADC like the one calculated in (3).

Unlike in a Hopfield network, the convergence of the energy function toward its minimum in the proposed network is globally attractive and unaffected by the transient behavior of the circuit elements. Moreover, the proposed network outperforms the Hopfield network in terms of scalability: the number of synapses is halved, and each weight value is reduced by 2^i. In the next section, we show that the network converges after training to the minimum energy level.

C. Training Algorithm

The learning capability of the asymmetric Hopfield network was thoroughly investigated in [39], [40]. A learning algorithm based on the least mean square (LMS) algorithm was introduced, and several specific examples were considered to demonstrate the learning ability, network flexibility, linear separability for conversion, and the effectiveness of LMS in training the asymmetric network as compared to the Hopfield and multi-layer neural networks. The recurrency of the Hopfield network complicates its feasibility for *in situ* training and adaptivity. Alternately, the Hopfield network could be cascaded by a deep neural network, trained using the backpropagation algorithm, to adaptively calibrate quantization errors and maintain the magnitude of digital output code within a manageable operating voltage range, as presented in [32]. This extension separates between the training (encoding) path and the conversion (inference) path, which could complicate the feasibility of the

scalable level-shifted architecture [32], consuming a large number of resources, in contrast to the proposed network.

Consider the following *supervised learning* task. Assume a learning system that operates on K discrete trials, indexed by $k = 1, 2, \ldots, K$. In each trial k, the system is given an empirical data set of $\{V_{in}, T_i\}, i = 0, \ldots, N-1$, where $V_{in}^{(k)} \in R$ is a sampled analog pattern, $T_i \in R^N$ is the desired digital label for $D_i^{(k)}$ corresponding to $V_{in}^{(k)}$, and $D_i^{(k)}$ is the actual i-th digital output, with all pairs sharing the same desired relation, $T_i^{(k)} = f(V_{in}^{(k)}, D_1, \ldots, D_{i-1})$. Note that two distinct patterns can have the same label (the same digital level in the ADC case depends on the quantization resolution). The goal of the system is to estimate (learn) the function $f(\cdot)$ using the empirical data. Suppose $\mathbf{W}$ is an asymmetric matrix as discussed, and consider each neuron estimator as

$$D_i^{(k)} = u\left(V_{in}^{(k)} - \sum_{j>i}^{N-1} W_{ij}^{(k)} D_j^{(k)} + c\right), \quad (7)$$

where $u(\cdot)$ is denoted as the signum neural activation function, and c is a constant that refers to a reference voltage, while each D_i behaves as a linear classifier with one output and forward propagates to approximate other outputs. Thus, there is no need for hidden layers, and the signum activation function is sufficient for estimating the function $f(\cdot)$. The proposed network could be seen as a concurrent single-layer or a pipelined feedforward multi-layer neural network where each layer determines an output bit. Each estimator $D_i^{(k)}$ should aim to predict the correct *teaching labels* $T_i^{(k)}$ for new unseen *patterns* V_{in}. To solve this problem, W is tuned to minimize some measure of error between the estimated and desired labels, over a K_0-long subset of the empirical data, or training set (for which $k = 1, \ldots, K_0$). Then, a common measure *error* metric is the least mean square error function defined as

$$E_{LMS} = \frac{1}{2}\sum_{k=1}^{K_0}\sum_{i=0}^{N-1}\left(D_i^{(k)} - T_i^{(k)}\right)^2, \quad (8)$$

where the 1/2 coefficient is for mathematical convenience. One can use different error measures as well. The performance of the resulting estimators is then tested over a different subset, called the test set ($k = K_0 + 1, \ldots, K$). A reasonable iterative algorithm for minimizing the error (that is, updating W where the initial choice of W is arbitrary) is the following instance of online stochastic gradient descent,

$$W^{(k+1)} = W^{(k)} - \frac{\eta}{2}\nabla_{W^{(k)}}\sum_{i=0}^{N-1}\left(D_i^{(k)} - T_i^{(k)}\right)^2, \quad (9)$$

where η is the *learning rate*, a (usually small) positive constant, and each iteration k, a single empirical sample $V_{in}^{(k)}$ is chosen randomly and presented at system input. The chain rules (7) and (8) are used to obtain the outer product [39]:

$$\Delta W_{ij(j>i)}^{(k)} = -\eta\left(T_i^{(k)} - D_i^{(k)}\right)T_j^{(k)}. \quad (10)$$

This update rule is known as the least mean square (LMS) algorithm [41], used in adaptive filters for signal processing and
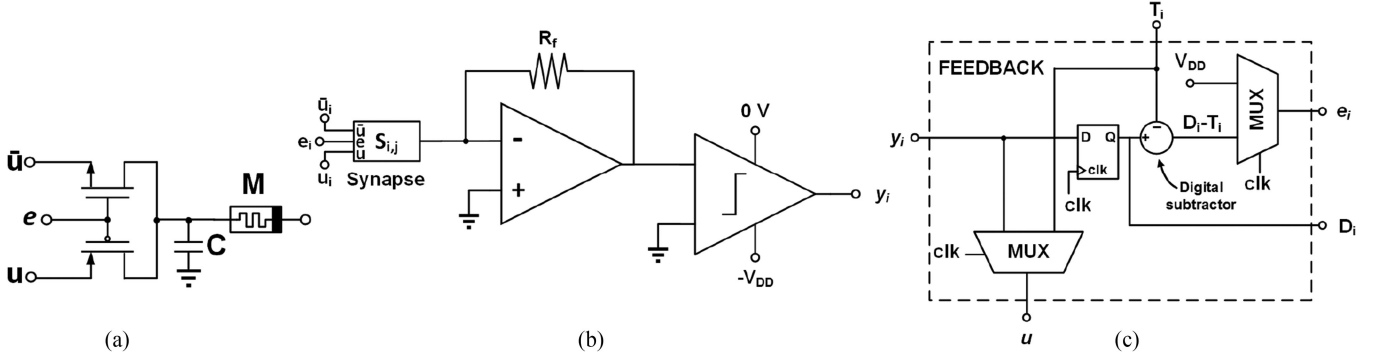
Fig. 5. Building blocks of the neural network 4-bit ADC. (a) Schematic of the memristive synapse $S_{i,j}$. Note that $W_{ij} = R_f/S_{ij}$. (b) Schematic of the neuron, which comprises an inverting OpAmp for integration and a latched-comparator for decision-making. (c) Digital feedback circuit for the gradient descent algorithm.

control [42], [43]. Note that the update rule (10) is local, i.e., the change in synaptic weight $W_{ij(j>i)}$ depends only on the related components $D_i^{(k)}, T_i^{(k)}, T_j^{(k)}$. This local update, widely used in ANN training and ML algorithms, allows massively parallel acceleration [10]. The training phase continues until the error is below $E_{\text{threshold}}$, a small predefined constant threshold that quantifies the learning accuracy. We show in the next section, for the first time, that the error function in (8) after training is proportional to the cost function in (5) and the network energy function in (4). The training algorithm is implemented by the feedback shown in Fig. 4(b), and its flow resembles the flow presented in our previous work [11].

IV. CIRCUIT DESIGN

In this section, we present the circuit design building blocks of the proposed ADC architecture, including its different components: neuron, synapse, and feedback circuit. The design methodologies, operational mechanism, and constraints of the building blocks are based on our previous work [11]. For simplicity, we provide the circuit design of the quantization stage and assume that the analog input is sampled separately by means of an external sample-and-hold circuit.

A. Artificial Synapse

We adopt our synapse circuit design from earlier work [10], [11]: a single voltage-controlled memristor, connected to a shared terminal of two MOSFET transistors (p-type and n-type), as shown in Fig. 5(a). The circuit utilizes the intrinsic dynamics of the memristive crossbar (2T1R), which inherently implements Ohm's and Kirchhoff's laws for ANN hardware realization [9]. The output of the synapse is the current flowing through the memristor. The synapse receives three voltage input signals: u is connected to the source of one transistor, $\bar{u} = -u$ is connected to the source of the other, and the *enable* signal e is connected to the gates of both. The enable signal e can have a zero value, meaning that both transistors are non-conducting, V_{DD}, meaning that only the NMOS is conducting, or $-V_{DD}$, meaning that only the PMOS is conducting.

The synaptic weight is modified in accordance with the value of e, which selects either input u or $\overline{u}$. Hence, the writing voltage, V_w (or $-V_w$), is applied via the source terminal of both transistors. Note that the right terminal of the memristor

is connected to the virtual ground of an OpAmp [11], whereas the left terminal is connected to a transistor that operates in the ohmic regime and a shock absorption capacitor [44]. The memristor value $M_{i,j}$ varies between low and high resistance states, R_{on} and R_{off}, respectively. The assumption of transistors in ohmic operation bounds the write and read voltages, and constrains the initial memristive state variable and other design parameters, as further described in our previous work [11].

For the design of the proposed ADC, we have used 0.18 μm CMOS process, and memristors fitted by the VTEAM model [45] to the Pt/HfO$_x$/Hf/TiN RRAM device with a buffer layer [46]. This device has a high-to-low resistance state (HRS/LRS) ratio of $\sim$50 and low forming, set, and reset voltages. The circuit parameters are listed in Table I.

B. Artificial Neuron

The neural activation is the *de facto* activity in neuromorphic computing that collectively integrates analog inputs and fires output by means of a non-linear activation function. The neural activity is a mathematical abstraction that simply aims to capture some features of real biological neurons. Several implementations of artificial neuron circuits have been suggested in the literature [6], [47]–[49]. The neural activation function in the originally proposed Hopfield neural network [25] has some constraints in linearity and monotonicity, as were carefully implemented in [31] using a complicated design to ensure disturbance-free and leakless transient neural activity. Fortunately, in asymmetric Hopfield networks, no such strict constraints are required, and simple digital comparators can be used [24], [35], while device mismatches, parasitics, and instability issues of the neuron circuit are adaptively compensated for by the synapse.

Thus, our neuron circuit is realized, as shown in Fig. 5(b), by a transimpedance amplifier implemented as an inverting operational amplifier (OpAmp), cascaded to a comparator with zero voltage reference, zero voltage V_{max}, and $-V_{\text{dd}}$ as V_{min} to generate negative signs for the inhibitory synapses of the LSBs. The comparator is latched using time-interleaved phased clock, and its decision result (0 V or $-V_{\text{dd}}$) is sampled at the end of the reading cycle T_r, after transient effects are mitigated and neurons synchronized, and their outputs are forward propagated in pipeline. It is latched for the entire writing cycle T_w, and handled by the feedback circuit. Note that the effective weights are

TABLE I
CIRCUIT PARAMETERS

Type	Parameter	Value	Type	Parameter	Value
Device parameters			**Design Parameters**		
Power supply	V_{DD}	$1.8V$	Shock capacitor	C_{shock}	$100\,fF$
NMOS	W/L	10	Writing voltage	V_W	$\pm 0.5\,V$
	V_{T_n}	$0.56\,V$	Reading voltage	V_r	$-0.1125\,V$
PMOS	W/L	20	Feedback resistor	R_f	$45\,k\Omega$
	V_{T_p}	$-0.57\,V$	Reading time	T_r	$5\,\mu s$
	$V_{on/off}$	$-0.3\,V, 0.4V$	Writing time	T_w	$5\,\mu s$
	$K_{on/off}$	$-4.8mm/s, 2.8mm/s$	Parasitic capacitance	C_{mem}	$1.145\,fF$
	$\alpha_{on/off}$	$3, 1$	Parasitic inductance	L_{mem}	$3.7\,pH$
Memristors	R_{ON}	$2\,k\Omega$	Input resistance	R_{in}	$45\,k\Omega$
	R_{OFF}	$100\,k\Omega$	Comparator bandwidth	BW	$4\,GHz$
	$f(s)$	$s \cdot (1-s)$	OpAmp gain	A	100
ADC parameters			**Learning parameters**		
Sampling frequency	f_s	$0.1MSPS$	Learning rate	η	0.01
Number of bits	N	4	Error threshold	$E_{threshold}$	$2 \cdot 10^{-3}$
Full-scale voltage	V_{FS}	$[\frac{V_{DD}}{2} - V_{DD}]$			

normalized via the OpAmp and equal to $W_{ij,j>i} = R_f/S_{ij,j>i}$, where R_{f} is the negative feedback resistor and S_{ij} is the effective resistance of M_{ij} and the serial transistor.

C. Feedback Circuit

The online gradient descent algorithm is executed by the feedback circuit, which precisely regulates the synaptic adaptation procedure. Our aim is to design (10) in hardware and execute basic subtraction and multiplication operations. The ADC system is more sophisticated than the DAC system [11] and has stronger applicative impact; however, its training circuit design is much simpler because $D_i^{(k)}$, $T_i^{(k)}$, $T_j^{(k)}$ as they appear in (10) are digital values that do not require modulation techniques. The subtraction product $(T_i^{(k)} - D_i^{(k)})$ is implemented by a digital subtractor, as shown in Fig. 5(c). The subtraction result of each neuron (other than the MSB) is backward propagated as an enable signal e simultaneously to all its synapses. The multiplication is invoked as an AND logic gate via the synapse transistors and controlled by e, whereas the attenuated desired digital output $T_j^{(k)}$ is connected via the source of the synapse. All circuits are controlled by interchangeable synchronous read and write clock cycles with the ADC sampling frequency f_s. After the training is complete ($E \leq E_{\mathrm{threshold}}$), the feedback is disconnected from the conversion path.

V. EVALUATION

In this section, our proposed four-bit ADC design is discussed and evaluated in a SPICE simulation (Cadence Virtuoso) using a $0.18\mu m$ CMOS process and the VTEAM memristor model [45]. The simulation methodology is based on our previous work [11]. First, the learning algorithm is evaluated in terms of least mean square error and training time. Next, the circuit is statically and dynamically evaluated, and finally power consumption is analyzed. The proposed ADC functionality and robustness were massively tested under extreme conditions using MATLAB. The design parameters and constraints are listed in Table I. Furthermore, circuit variations and noise sources are quantified and validated, as listed in Table II.

A. Reconfiguration

The basic deterministic functionality of the four-bit ADC is demonstrated during training by the online gradient descent algorithm. The learning rate is crucial to the adaptation performance: it depends on the circuit parameters, the write voltage, the pulse-time width, the feedback resistor, the present state, and the physical properties of the memristive device. The learning rate is

$$\eta(t) = \frac{\Delta R}{R} = \frac{(R_{\mathrm{OFF}} - R_{\mathrm{ON}})\Delta s(t)}{R_f}, \qquad (11)$$

where Δs is the change in the memristor's internal state defined as in the VTEAM model [45],

$$\Delta s = \int_0^{T_w} K_{\mathrm{on/off}} \left(\frac{V_W}{V_{\mathrm{on/off}}} - 1 \right)^{\alpha_{\mathrm{on/off}}} \cdot f(s)\, dt, \qquad (12)$$

where $K_{\mathrm{on/off}}$, and $\alpha_{\mathrm{on/off}}$ are constants that describe, respectively, the state evolution rate and its nonlinearity, $V_{\mathrm{on/off}}$ are voltage thresholds, and $f(s)$ is a window function that adds nonlinearity and state dependency during state evolution. These parameters are fitted to the Pt/HfO$_{\mathrm{x}}$/Hf/TiN RRAM device [46]. The fitted learning rate succeeded to converge to a global minimum with high accuracy [39], [50]. The learning rate as given in (11) is state and time dependent.

Fig. 6(a) shows the resistive value of the synapses when two sawtooth training datasets with different full-scale voltage ranges (V_{DD} and $V_{DD}/2$) and different sampling frequencies (f_{s} and $100 f_{\mathrm{s}}$) are applied successively in real time. After approximately 4000 training samples, which is equal to 40 ms training time for 0.1MSPS conversion rate, the error according to (8) is below $E_{\mathrm{threshold}}$ and the network converges from a random initial state to a steady state. $E_{\mathrm{threshold}}$ is determined to be 50% effectively misclassified digital output codes (8 codes in the case of 4-bits) out of the total number of training samples, as listed in Table I. Furthermore, when the full-scale voltage changes to $V_{DD}/2$ and the sampling frequency changes to $100 f_{\mathrm{s}}$, the system converges to a new steady state that quantizes 0.9 V full-scale at a 10 MSPS sampling rate. In each case, the network is

reconfigured to operate correctly under different specifications, as illustrated by the different synaptic weights in Fig. 6(a). The least mean square error (5) optimization toward its gradient descent during training is shown in Fig. 6(b). In the same context, neural activity adaptation that denotes digital output bits is shown, at three different time stamps, in Fig. 6(c) for the initial state before training (samples 0–15), coarse-grained training (i.e., where the error is slightly higher than $E_{\text{threshold}}$, samples 220–235), and fine-grained training (i.e., where the error is sufficiently low and the ADC response converges to the desired state, samples 3720–3735). The digital outputs are ideally converted to discrete analog via an ideal 4-bit DAC that is connected back-to-back and accurately reproduces the ADC's present state, as shown in Fig. 6(d) at the same three time stamps.

B. Self-Calibration

As introduced in Section II, the accuracy of an ADC depends on many critical factors including process variations, frequency-dependent variations, device mismatches, device wear out, parasitic effects, delays, poles, gain and offset errors. Table II lists the magnitude of variability for these effects. The process variation parameters for the memristor are pessimistically chosen [11], randomly generated with a normal distribution, and incorporated into the VTEAM model [45] with a variance of approximately 10% to cover wide reliability margins [10]. Transistor parameters such as V_W, W/L, and V_T in Table I are chosen to guarantee a globally optimal solution even under such extreme conditions. In Fig. 6, we show that the proposed training algorithm can tolerate such variations over time and compensate for them by using different synaptic weights.

We statically evaluated how the proposed ADC responds to the DC ramp signal at the three given time stamps, as illustrated in Fig. 7(a) and (b). The teaching staircase in Fig. 6(d) is a subset of DC ramp input that statically evaluates the ADC at the aforesaid time stamps. The differences between two adjacent digital output decimal codes within the actual ADC output are therefore the differential non-linearities (DNL). Likewise, the differences between the actual ADC output and the ideal staircase for each digital input code are the integral non-linearities (INL) [1]. The DNL of the last code is undefined. Results of the maximum DNL and INL are shown, respectively, in Fig. 7(a) and (b). Prior to training, the ADC is completely non-linear and non-monotonic, with several missing codes. Thus, INL $\approx$ 8 LSB, and DNL $\approx$ 5 LSB. Improved performance can be seen at the second time stamp (2 ms ∼ 200 samples), where the ADC appears monotonic; however, it is still not accurate (INL $\approx$ −2 LSB, DNL $\approx$ 2 LSB). After the training is complete (40 ms), the ADC is almost fully calibrated, monotonic, and accurate: INL $\approx$ 0.4 LSB, and DNL $\approx$ 0.5 LSB.

Furthermore, parasitic effects such as capacitance and inductance, as listed in Table II, which are the dominant factors in ADC accuracy at high frequencies, have been adaptively captured as simulated by 10 MSPS within longer training time.

C. Noise Tolerance

In contemporary ADCs, calibration mechanisms [17] can be used to compensate for device mismatch and process
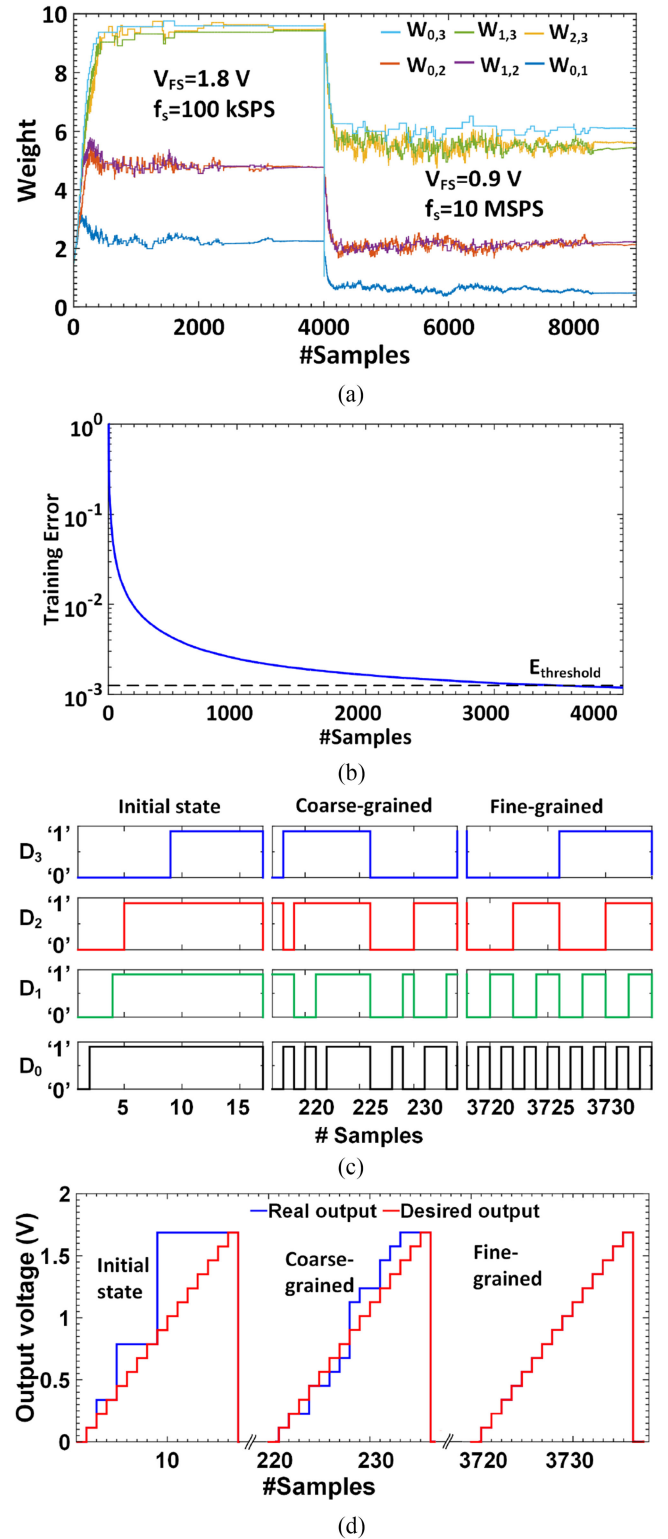

(a)

(b)

(c)

(d)

Fig. 6. Training evaluation. (a) Synaptic weight reconfiguration during the training phase for the $V_{FS} = 1.8$ V and $f_s = 100$ KSPS. Synapses are immediately trained for the $V_{FS} = 0.9$ V and $f_s = 10$ MSPS and shown in real time. The synaptic weight is equal to the ratio between R_f and the corresponding memristor, thus it has no units. (b) The LMS error function optimization during training until it achieves $E_{\text{threshold}}$. (c) The actual digital outputs D_i (logical value) at three different time stamps during the training; periodic digital outputs are achieved after the training is finished, corresponding to the analog input ramp. (d) Comparison between the corresponding discrete analog values of the teaching dataset and the actual output by connecting it to an ideal DAC, at three different time stamps during the training; an identical staircase is obtained after the training is complete.

Type	Nominal value	Variance
Device mismatch		
Resistor	$W = 2um$ $R_\square = 50\Omega/\square$	$\pm 0.5\%um$
Capacitor	$W = 0.15um$ $C_A = 0.68fF/um^2$	$\pm 1\%um$
NMOS/PMOS	W/L V_T	$\pm 10\%$ $\frac{A_{V_T}}{\sqrt{WL}} = 7.14mV$
Sampler	τ	$400ps$
OpAmp finite gain	$-\frac{\frac{R_f}{R_{on}}}{1+(1+\frac{R_f}{R_{on}})/A}$	81
Comparator	V_{offset}	$5mV$
Memristor	$V_{on/off}$ $K_{on/off}$ R_{ON} R_{OFF}	$\pm 10\%V$ $\pm 10\%mm/s$ $\pm 10\%\Omega$
Noise sources		
Thermal noise	$2kTg^{-1}, kT/C$	$10^{-16}V^2s$
IR drop	V_w	$\pm 10\%V$
Quantization noise	$\frac{V_{FS}}{2^{N+1}} = 56.25mV$	$\frac{V_{FS}}{2^{N+1}\sqrt{3}} = 32.5mV$
Frequency-dependent noise and variations / aging		
ADC cutoff frequency	$f_{T,max}$	$1.668Ghz$
Propagation time	$R_{OFF} \cdot C_{in}$	$30ps$
Input referred noise	$\log_2\left(\frac{V_{FS}^2}{6kTRf_s}\right)^{0.5} - 1$	$1.27mv$
OpAmp input noise	$1/f$ flicker noise	$10nv/\sqrt{Hz}$
Slew rate	$2\pi f V_{FS}$	$1.13V/ns$
Comparator ambiguity	$\frac{\pi BW}{6.93f_s} - 1.1$	$0.625mV$
Jitter	$\log_2(\frac{2}{\sqrt{3}\pi f_s \tau_{jitter}}) - 1$	$50ps$
Memristor stochastics	Poisson process (τ)	$\frac{2.85 \cdot 10^{-5}}{e^{\frac{V_w}{0.156}}} = 1.1\mu s$
Memristor OFF impedance	R_{OFF}	$\frac{R_{OFF}}{\sqrt{(1+(R_{OFF}C_{mem} \cdot 2\pi f)^2}}$
Endurance degradation	ΔR	$10\%/decade$

imperfections, but noise can irreparably degrade performance. Noise is also less straightforward to capture at design time. However, we believe that the effects of intrinsic noise on the performance of the analog circuit are relatively small: adaptive intelligent systems that employ machine learning techniques are inherently robust to noise, because noise is a key factor in the type of problems they are designed to solve.

Noise sources include intrinsic thermal noise from the feedback resistor, memristor, and transistor [10], in addition to quantization noise [51], jitter, comparator ambiguity [3], input referred noise, random offsets [19], non-linear distortions, training label sampling noise, memristor switching stochastics, and frequency-dependent noise [52]. These noise sources are listed in Table II.

The ADC non-linear functionality $V_{\text{out}} = f(v_i)$ in response to voltage input $v_i = A\cos(\omega t)$, where A is the amplitude and ω is the frequency, could be qualitatively described as

$$V_{\text{out}} = a_0 + a_1 A\cos(\omega t) + \frac{a_2 A^2}{2}[1 + \cos(2\omega t)] + \dots, \quad (13)$$

where a_o is the DC constant, a_1 is the small-signal gain constant, and a_2 is the distortion constant. Thus, as a result of non-linear effects, we get harmonic distortions, which appear as spectral spurs in sampling frequency multiples and degrade the SNDR and the ADC precision. We show that the proposed algorithm is able to adaptively alleviate non-linear distortions and tolerate noise by estimating the $f(\cdot)$ function given in Section III-C.

The ADC is dynamically evaluated and analyzed, at the three given time stamps, in response to a sinusoidal input signal with 44 kHz frequency, which meets the Nyquist condition, $f_{\text{input}} \leq f_s/2$, and applies for coherent fast Fourier transform (FFT) using a Hamming window and a prime number of cycles distributed over 5000 samples, which is sufficient for reliable FFT without collisions and data loss [53]. Fig. 7(c) shows the FFT for signal and distortion power as a function of frequency, where each time stamp is shown in a different color. The ADC cutoff frequency $f_{T,\max}$ is bounded by the high-to-low memristor impedance ratio [11]. Fig. 7(c) illustrates that the harmonic distortions are mitigated, the fundamental power increases, and the SNDR and ENOB improve as the training progresses.

Synaptic fluctuations arising due to noise and variation sources are mitigated by the switching non-linearity and threshold of the memristor [11]. Nonetheless, the gradient descent algorithm continues capturing and averaging stochastic dynamics and timing uncertainties (jitter) of the sampled input. The comparison to noisy labels will strengthen the immunity of the network against overfitting [54] and achieve reliable generalization performance. In the same context, the memristor switching stochasticity is characterized by a Poisson process [55], as listed in Table II, and incorporated into the VTEAM model [45] as a probabilistic shift in the threshold [55]. Along with the quantization noise or dither, this helps the network converge to a global minimum, and improve the ENOB, breaking through the thermal noise limit in some cases. This well-known phenomenon is called stochastic resonance. It was reported in the past in ANNs [56] and memristors [57]. Note that smaller learning rates will better overcome real-time variations; however, this will come at the cost of a training time penalty. The effective number of stable resistive levels, as a function of noise margin (*NM*) due to statistically correlated variation sources, was massively analyzed using Monte-Carlo simulations in our previous work [11]. Furthermore, its impact on the ENOB was determined, with typical results (in 38% of the cases) of 64 resistive levels, $\sim$3% NM, and $\sim$3.7 ENOB.

D. Power Optimization

Section III-B shows the equivalence between the Hopfield-like energy function of the network given by (4) and the cost function that solves the conversion optimization given by (6). The cost function achieves its minimum, lower bounded by quantization error power, when the synaptic weights are configured so as to guarantee that each analog input is correctly mapped to its corresponding digital output. In Fig. 6(b), we show that the error function given by (8) achieves a global minimum when the network is successfully trained to configure the ADC. Consequently, the power consumption is optimized dur-

ing training until it achieves its minimal value when the training is finished. The best energetic state of the proposed network is achieved when it is configured in an ADC manner. Consequently, the power dissipation of the entire network is analyzed, and is attributed to three sources:

1. Neural integration power: the power dissipated on the feedback resistor of the OpAmp is

$$P_{int_i} = \left(V_{in} - 2^i V_{ref} - \sum_{j=i+1}^{N-1} \frac{R_f}{R_{ij}} V_j \right)^2 / R_f. \quad (14)$$

This function solves the ADC quantization after training for each neuron, as described in (3). The total neural integration power dissipated on all neurons is $P_{int} = \sum_{i=0}^{N-1} P_{int_i}$.

2. Neural activation power: the power dissipated on the comparators and OpAmps at the sampling frequency. This power source is constant and negligible: $P_{act_i} = 3\,\mu W$ in 0.18 μm CMOS process in f_T. The total activation power dissipated on all neurons is $P_{act} = \sum_{i=0}^{N-1} P_{act_i}$.

3. Synapse power: the power dissipated on synapses, including reconfigurable and fixed synapses for each neuron, is

$$P_{synapse_i} = \frac{V_{in}^2}{R_f} + \frac{2^i V_{ref}^2}{R_f} + \sum_{j=i+1}^{N-1} \frac{V_i^2}{R_{ij}}. \quad (15)$$

The total synaptic power dissipation is $P_{synapse} = \sum_{i=0}^{N-1} P_{synapse_i}$. Note that an effective transistor resistance in series to the memristor is also taken into account.

Thus, the total power consumption is the sum of the three power sources averaged on a full-scale ramp with 2^N samples (epoch), as shown in Fig. 7(d) during training time. Each point in the horizontal axis represents a full-scale ramp, and its corresponding value in the vertical axis represents the average of the total dissipated power. After the training is finished and the network configured as an ADC, the average of the synapse power on a full-scale ramp is half of the maximum power dissipated, and the neural integration power is minimal. This balance results in optimal power dissipation.

Note that the dynamic power consumption as a result of updating memristors during the training phase is not determined and is not considered as conversion power dissipation by FOM definition in (1). We neglec the power dissipation of the feedback because, after the training is finished, the feedback is disconnected and the network maintains the minimal achieved power dissipation level during conversion. We assume that this power source is relatively low because of the small area of training feedback, short training time, and the small ratio between training to conversion cycles during the lifetime of the converter [11], even at a high rate of application configurations.

VI. DISCUSSION

In the broader scope of the results shown in Section V, we discuss the potential to break through the speed-power-accuracy tradeoff. Furthermore, we discuss the scaling challenges of the proposed architecture.
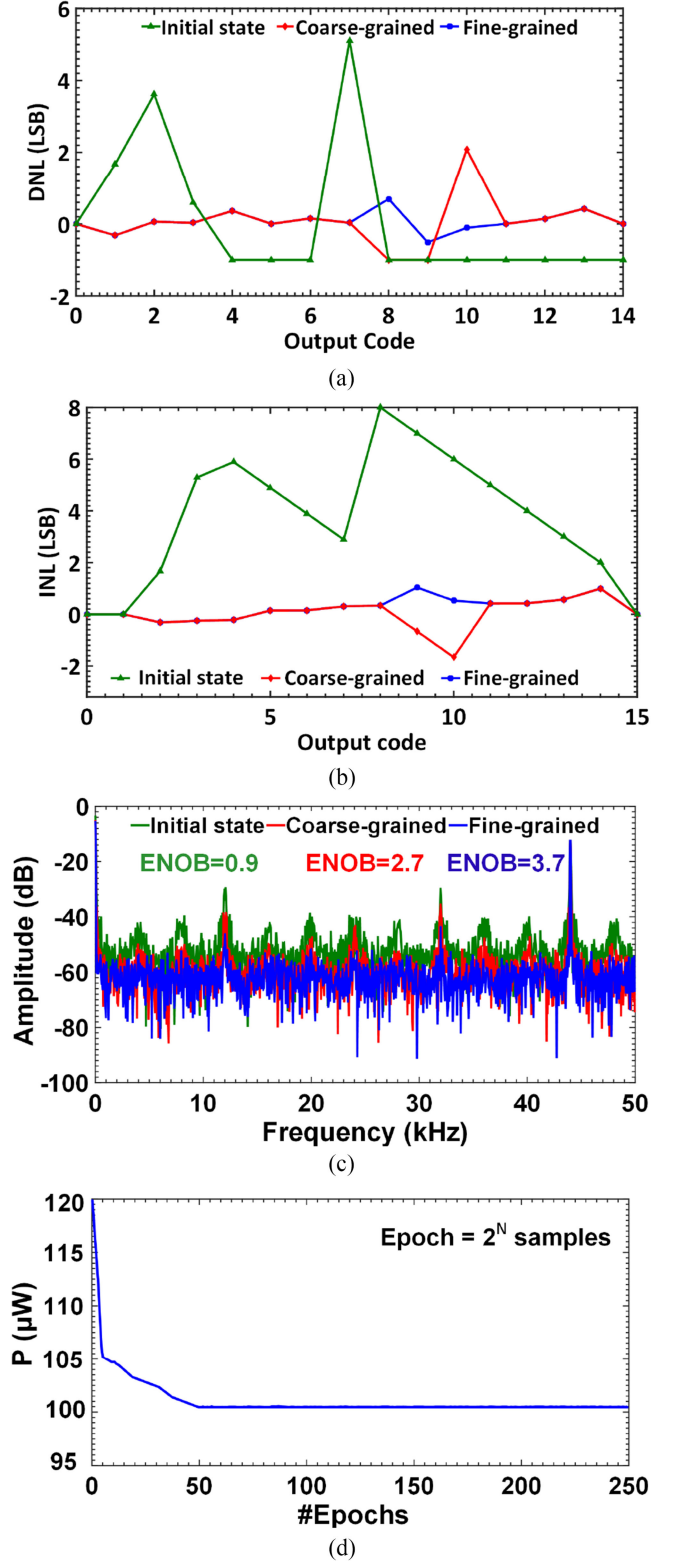

Fig. 7. Static conversion evaluation that shows the efficiency of the training algorithm in mismatch calibration (a) differential and (b) integral non-linearities of the ADC at three different time stamps in response to the DC input voltage ramp. (c) Dynamic conversion evaluation that shows the efficiency of the training algorithm in noise tolerance and distortion mitigation by coherent fast Fourier transform of the ADC output in response to a sinusoidal input signal with 44 kHz frequency at three different time stamps during the training with ENOB calculation. (d) Power evaluation of the network that shows power optimization during training.
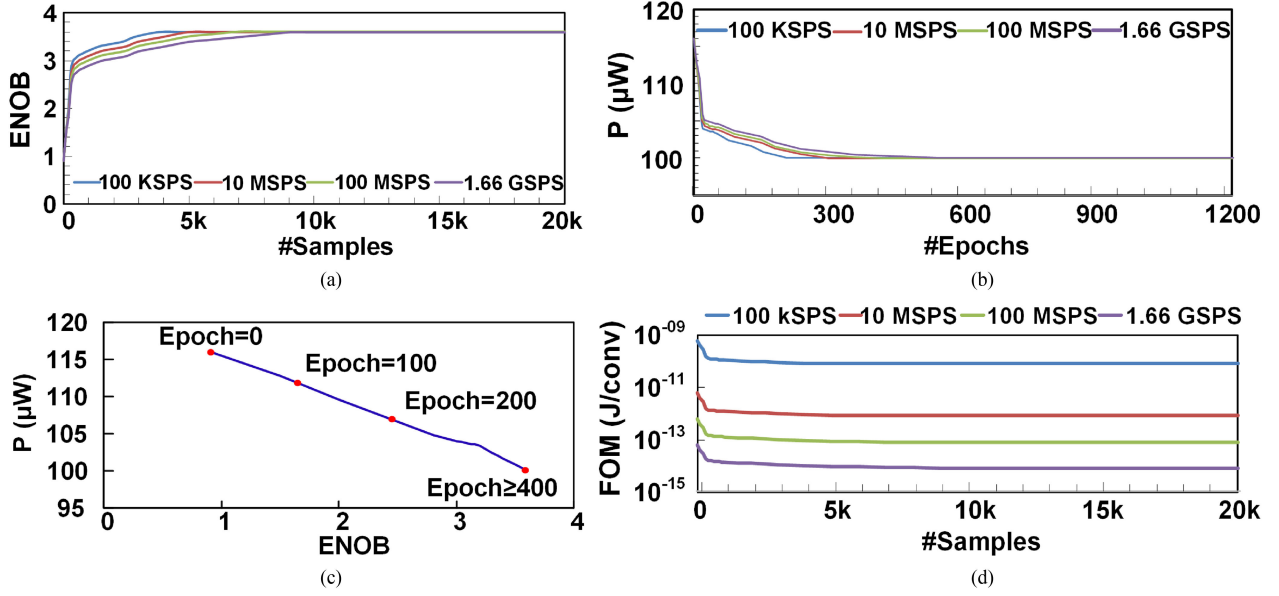
Fig. 8. Breaking through the speed-power-accuracy tradeoff. (a) Speed-accuracy tradeoff by achieving maximal *ENOB* regardless of f_s after training is complete. (b) Speed-power tradeoff by achieving minimal *P* regardless of f_s after the training is complete. The frequency-dependent power dissipation is negligible. (c) Accuracy-power tradeoff by achieving maximal *ENOB* and minimal *P* after the training is complete. (d) FOM dynamic optimization with training.

A. Breaking Through the Speed-Power-Accuracy Tradeoff

Having demonstrated the dynamic mechanism of the trainable ADC proposed in Section II, we investigate the real-time training of the ADC for general purpose applications. For every selected f_s within the f_T bandwidth, the ADC is trained correspondingly by a training data-set with the same specifications and achieves optimal ENOB as shown in Fig. 8(a). The maximal ENOB ($\sim$3.7) is asymptotically bounded by the intrinsic quantization noise, which is not exceeded. Analogously, the power consumption is dynamically optimized for every f_s to achieve the minimal power dissipation of the network, as shown in Fig. 8(b). The power dissipated on resistors has a greater effect on overall power dissipation than the frequency-dependent dissipation (e.g., capacitors). Simultaneously, and as we show that the equivalence between the quantization cost function (6) and the energy function (4) after the error function (8) is optimized, co-optimization in terms of both ENOB and power dissipation along the training samples is achieved, as shown in Fig. 8(c).

Interestingly, the collective optimization of the proposed architecture breaks through the speed-power-accuracy tradeoff, and dynamically scales the FOM to achieve a cutting-edge value of 8.25 fJ/conv.step, as shown in Figs. 8(d) and 2. The versatility of the proposed architecture with regard to reconfiguration, mismatch self-calibration, noise-tolerance, and power optimization is attained using a simple and minimalistic design with a reconfigurable single-channel. The proposed architecture moreover utilizes the resistive parallel computing of memristors to achieve high speed, in addition to its analog non-volatility, enabling a standard digital ML algorithm to intelligently adjust its conductance precisely and *in situ* to achieve high-accuracy. The minimalistic design results in low-power consumption, thus achieving a cost-effective ADC. All these features, when combined with the SAR architecture, the pipelined architecture, and the online trainable mechanism, will enable a general-purpose application architecture.

B. Challenges

However, scaling the proposed architecture is challenging. When increasing the scale of the network, the number of neurons, synapses, and feedbacks are quadratically higher. Consequently, this will increase the area and power consumption substantially, as calculated in Table III based on [11], [46], [58], [59]. Due to the successive nature of the proposed ADC architecture, higher numbers of neurons require longer conversion time as a result of the propagation time, settling time, and decision-making time of each. Therefore, to eliminate signal aliasing, the maximal Nyquist sampling frequency will unfortunately be limited, as determined in Table III.

Additional challenges in scaling are the required high-to-low resistance states ratio of the synaptic weights, the number of resistive levels, cutoff frequency, and endurance. We calculated the maximal number of bits in our previous work [11], which is four bits for the memristive device under test, but devices with higher HRS/LRS are achievable [60]. Moreover, we show in this paper that device-dependent properties are compensated for by longer training time to achieve maximal ENOB, which is equal to *(N-3)* bits regardless of the conversion speed. Overall, the FOM still improves as the number of bits increases, because of the optimal achieved ENOB, as calculated in Table III. Furthermore, in advanced CMOS technology nodes the FOM will improve due to lower power consumption and higher sampling rates. These findings prove that the proposed architecture is conceptually and practically scalable, even in the presence of the mentioned scaling challenges. These shortcomings still need to be investigated by leveraging mixed-signal architectures and deep neural network concepts. Our future work will consider training deep neural networks for large-scale architectures that contain multi-cores of the proposed 4-bit ADC and its complementary 4-bit DAC, proposed in our previous work [11], in a pipelined, time-interleaved, or oversampling style.

TABLE III
SCALABILITY EVALUATION

#Bits	Area			Time			Power (μW)	FOM (fJ/conv)	Memristor	
	#Neurons, #feedbacks	#Synapses	Total (μm²)	Conversion rate (GSPS)	Training (KSamples)	Wearout (trainings/day for 10 yrs)			$\frac{HRS}{LRS}$	#levels
4	4	10	4850	1.66	4	150	100	8.25	2^4	64
8	8	36	9740	0.74	6	100	650	7.5	2^8	2048
N	N	$\frac{N(N+1)}{2}$	$\approx N(1.1N + 1208)$	$\dfrac{1}{N \cdot t_p + \dfrac{N-1}{BW}}$	$(2 - 2^{1-\frac{N}{4}}) \cdot 4$	$\dfrac{150}{2 - 2^{1-\frac{N}{4}}}$	$P_{int} + P_{act} + P_{synapse}$	$\dfrac{P}{2^{N-0.3}f_s}$	$2^{N-1+log_2 \frac{V_{DD}}{V_{FS}}}$	$N \cdot 2^N$

VII. CONCLUSION

This paper proposes a proof-of-concept of a real-time trainable ADC architecture for general purpose applications, which breaks through the speed-power-accuracy tradeoff. Motivated by the analogies between mixed-signal circuits and the neuromorphic paradigm, we exploit the intelligent properties of an ANN, and suggest a pipelined SAR-like neural network architecture ADC that is trained online by a supervised ML algorithm. The proposed network shares the Hopfield energy model, and we show the equivalence between the energy function to the conversion cost function and the training error function after the training is complete.

The neural network is realized by means of a hybrid CMOS–memristor circuit design. The trainable mechanism successfully proves collective properties of the network in reconfiguration to multiple full-scale voltages and frequencies, mismatch self-calibration, noise-tolerance, stochastic resonance, power optimization, and FOM dynamic scaling. We believe that the proposed ADC constitutes a milestone with promising results for large-scale architectures of data converters and emerging real-time adaptive applications with varying conditions, such as wearable devices and automotive applications.

ACKNOWLEDGMENT

REFERENCES

[1] R. J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*. New York, NY, USA: Springer, 2013.

[2] M. Steyaert and K. Uyttenhove, "Speed-power-accuracy trade-off in high-speed analog-to-digital converters: now and in the future . . . " in *Analog Circuit Design*. New York, NY, USA: Springer, Apr. 2000, pp. 3–24.

[3] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[4] P. Kinget and M. S. J. Steyaert, "Impact of transistor mismatch on the speed-accuracy-power trade-off of analog CMOS circuits," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1996, pp. 333–336.

[5] B. E. Jonsson, "A survey of A/D-converter performance evolution," in *Proc. IEEE Int. Conf. Electron., Circuits Syst.*, Dec. 2010, pp. 766–769.

[6] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[7] S. H. Jo *et al.*, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, Apr. 2010.

[8] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology.*, vol. 24, no. 38, Sep. 2013, Art. no. 384010.

[9] M. Prezioso *et al.*, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, May 2015.

[10] D. Soudry *et al.*, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2408–2421, Oct. 2015.

[11] L. Danial, N. Wainstein, S. Kraus, and S. Kvatinsky, "DIDACTIC: A data-intelligent digital-to-analog converter with a trainable integrated circuit using memristors," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 146–158, Mar. 2018.

[12] B. J. Hosticka, "Performance comparison of analog and digital circuits," *Proc. IEEE*, vol. 73, no. 1, pp. 25–29, Jan. 1985.

[13] B. Murmann, "A/D Converter trends: power dissipation, scaling and digitally assisted architectures," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2008, pp. 105–112.

[14] B. Murmann, "ADC Performance Survey 1997-2017," [Online]. Available: http://web.stanford.edu/~murmann/adcsurvey.html

[15] K. Uyttenhove and M. S. J. Steyaert, "Speed-power-accuracy tradeoff in high-speed CMOS ADCs," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 4, pp. 280–287, Apr. 2002.

[16] Y. Chiu, B. Nikoli, and P. R. Gray, "Scaling of analog-to-digital converters into ultra-deep-submicron CMOS," in *Proc. IEEE Custom Integr. Circuits Conf.*, pp. 368–375, Sep. 2005.

[17] J. Li, S. Member, U. Moon, and S. Member, "Background calibration techniques for multistage pipelined ADCs with digital redundancy," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 9, pp. 531–538, Sep. 2003.

[18] Bin Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, "Analog-to-digital converters," *IEEE Signal Proc. Mag.*, vol. 22, no. 6, pp. 69–77, Nov. 2005.

[19] P. Nuzzo, F. De Bernardinis, P. Terreni, and G. Van der Plas, "Noise analysis of regenerative comparators for reconfigurable ADC architectures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1441–1454, Jul. 2008.

[20] M. Yip and A. P. Chandrakasan, "A Resolution-reconfigurable 5-to-10-bit 0.4-to-1 V power scalable SAR ADC for sensor applications," *IEEE J. Solid-State Circuits*, vol. 48, no. 6, pp. 1453–1464, Apr. 2013.

[21] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Comput.*, vol. 10, no. 7, pp. 1601–1638, Oct. 1998.

[22] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *IEEE Comput.*, vol. 29, no. 3, pp. 31–44, Mar. 1996.

[23] T. Kugelstadt, "The operation of the SAR-ADC based on charge redistribution," *Texas Instrum. Analog Appl. J.*, pp. 10–12, Feb. 2000.

[24] C. Po-Rong, W. Bor-Chin, and H. M. Gong, "A Triangular connection hopfield neural network approach to analog-to-digital conversion," *IEEE Trans. Instrum. Meas.*, vol. 43, no. 6, pp. 882–888, Dec. 1994.

[25] D. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. 33, no. 5, pp. 533–541, May 1986.

[26] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, no. 10, pp. 3088–3092, May 1984.

[27] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[28] B. W. Lee and B. J. Sheu, "Design of a neural-based A/D converter using modified hopfield network," *IEEE J. Solid-State Circuits*, vol. 24, no. 4, pp. 1129–1135, Aug. 1989.

[29] B. W. Lee and B. J. Sheu, "Modified hopfield neural networks for retrieving the optimal solution," *IEEE Trans. Neural Netw.*, vol. 2, no. 1, pp. 137–142, Jan. 1991.

[30] L. Gao *et al.*, "Digital-to-analog and analog-to-digital conversion with metal oxide memristors for ultra-low power computing," *Proc. IEEE/ACM Int. Symp. Nanoscale Archit.*, Jul. 2013, pp. 19–22.

[31] X. Guo *et al.*, "Modeling and experimental demonstration of a hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits," *Frontiers Neurosci.*, vol. 9, pp. 1–8, Dec. 2015.

[32] A. Tankimanova, A. K. Maan, and A. P. James, "Level-shifted neural encoded analog-to-digital converter," in *Proc. IEEE Int. Conf. Electron., Circuits Syst.*, Dec. 2017, pp. 377–380.

[33] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: a model," *Science*, vol. 233, no. 4764, pp. 625–633, Aug. 1986.

[34] N. Dame, "Analysis and synthesis of neural networks with lower block triangular," *IEEE Trans. Circuits Syst.*, vol. 37, no. 10, pp. 1267–1283, Oct. 1990.

[35] G. Avitabile, M. Forti, S. Manetti, and M. Marini, "On a Class of nonsymmetrical neural networks with application to ADC," *IEEE Trans. Circuits Syst.*, vol. 38, no. 2, pp. 202–209, Feb. 1991.

[36] C. L. Sun, T. Zheng, O. Ishizuka, and H. Matsumoto, "Synthesis and implementation of T-model neural-based A/D converter," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, May 1992, pp. 1573–1576.

[37] M. J. S. Smith and C. L. Portmann, "Practical design and analysis of a simple 'neural' optimization circuit," *IEEE Trans. Circuits Syst.*, vol. 36, no. 1, pp. 42–50, Jan. 1989.

[38] V. Chande and P. G. Poonacha, "On neural networks for analog to digital conversion," *IEEE Trans. Neural Netw.*, vol. 6, no. 5, pp. 1269–1274, Sep. 1995.

[39] O. Ishizuka *et al.*, "A T-model neural network with learning ability," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Nov. 1991, vol. 3, pp. 2288–2293.

[40] Z. Tang, O. Ishizuka, and H. Matsumoto, "Backpropagation learning in analog T-model neural network hardware," in *Proc. Int. Conf. Neural Netw.*, Oct. 1993, vol. 1, pp. 899–902.

[41] B. Widrow and M. A. Lehr, "30 Years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.

[42] B. Widrow and S. D. Stearns, "Adaptive signal processing," Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[43] E. V. K. Madisetti, D. B. Williams, and S. C. Douglas, "Introduction to adaptive filters," in *Digital Signal Processing Handbook*. Boca Raton, FL, USA: CRC Press, 1999, pp. 7–12.

[44] S. Greshnikov, E. Rosenthal, D. Soudry, and S. Kvatinsky, "A Fully analog memristor-based multilayer neural network with online backpropagation training," in *Proc. IEEE Int. Conf. Circuits Syst.*, May 2016, pp. 1394–1397.

[45] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Trans Circuits Syst. II, Express Briefs*, vol. 62, no. 8, pp. 786–790, Aug. 2015.

[46] J. Sandrini *et al.*, "Effect of metal buffer layer and thermal annealing on HfOx-based ReRAMs," in *Proc. IEEE Int. Conf. Sci. Elect. Eng.*, Nov. 2016, pp. 1–5.

[47] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue VLSI," *Ann. Rev. Neurosci.*, vol. 18, pp. 255–281, Mar. 1995.

[48] M. A. C. Maher, S. P. Deweerth, M. A. Mahowald, and C. A. Mead, "Implementing neural architectures using analog VLSI circuits," *IEEE Trans. Circuits Syst.*, vol. 36, no. 5, pp. 643–652, May 1989.

[49] A. G. Andreou *et al.*, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 205–213, Mar. 1991.

[50] W. Wang *et al.*, "An adaptive neural network A/D converter based on CMOS/memristor hybrid design," *IEICE Electron. Express*, vol. 11, no. 24, Nov. 2014, Art. no. 20141012.

[51] R. M. Gray, "Quantization noise spectra," *IEEE Trans. Informat. Theory*, vol. 36, no. 6, pp. 1220–1244, Nov. 1990.

[52] Y. Nemirovsky *et al.*, "1/f Noise in advanced CMOS transistors," *IEEE Instrum. Meas. Mag.*, vol. 14, no. 1, pp. 14–22, Feb. 2011.

[53] O. M. Solomon, "The Use of DFT windows in signal-to-noise ratio and harmonic distortion computations," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, May 1993, pp. 103–108.

[54] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surveys*, vol. 27, no. 3, pp. 326–327, Sep. 1995.

[55] R. Naous, M. Al-Shedivat, and K. N. Salama, "Stochasticity modeling in memristors," *IEEE Trans. Nanotechnol.*, vol. 15, no. 1, pp. 15–28, Jan. 2016.

[56] R. Benzi, A. Sutera, and A. Vulpiani, "The mechanism of stochastic resonance," *J. Phys. A, Math. General*, vol. 14, no. 11, pp. L453–L457, Nov. 1981.

[57] A. Stotland and M. Di Ventra, "Stochastic memory: Memory enhancement due to noise," *Phys. Rev. E*, vol. 85, no. 1, Jan. 2012, Art. no. 011116.

[58] P. Kakoty, "Design of a high frequency low voltage CMOS operational amplifier," *Int. J. VLSI Design Commun. Syst.*, vol. 2, no. 1, pp. 73–85, Mar. 2011.

[59] S. B. Mashhadi and R. Lotfi, "Analysis and design of a low-voltage low-power double-tail comparator," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 2, pp. 343–352, Feb. 2014.

[60] H.-S. P. Wong *et al.*, "Metal-oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012.

Loai Danial received the B.Sc. degree in electrical engineering from the Technion–Israel Institute of Technology, Haifa, Israel, in 2014. He is currently working toward the Ph.D. degree in the Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology. From 2013 to 2016, he was with IBM Labs as a hardware research student. His current research combines interdisciplinary interests of data converters, machine learning, biology, and mixed-signal systems using emerging memory devices. He received the 2017 Hershel Rich Technion Innovation Award, and the Israeli Planning and Budgeting Committee Fellowship.

Nicolás Wainstein received the B.Sc. degree in electrical engineering from the Universidad de la República, Uruguay, in 2014. He is currently working toward the Ph.D. degree in the Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology. From 2014 to 2015, he was a Research and Teaching Assistant with the Institute of Physics and the Institute of Electrical Engineering, Universidad de la República. His current research interests include RF and analog circuits and systems using emerging memory technologies.

Shraga Kraus received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Technion – Israel Institute of Technology (IIT), Haifa, Israel, in 2001, 2006 and 2011, respectively. Since 1999, he has worked at several companies in the Israeli hi-tech industry, mainly on analogue and mixed signal integrated circuits. Today he is with PLSense Ltd, Yokneam, Israel.

Shahar Kvatinsky received the B.Sc. degree in computer engineering and applied physics and the M.B.A. degree from the Hebrew University of Jerusalem, Jerusalem, Israel, in 2009 and 2010, respectively, and the Ph.D. degree in electrical engineering from the Technion – Israel Institute of Technology, Haifa, Israel, in 2014. From 2006 to 2009, he was with Intel as a Circuit Designer and a Post-Doctoral Research Fellow with Stanford University from 2014 to 2015. He is currently an Assistant Professor with the Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology. His current research interests include circuits and architectures with emerging memory technologies and design of energy efficient architectures. He received the 2015 IEEE Guillemin-Cauer Best Paper Award, the 2015 Best Paper of Computer Architecture Letters, the Viterbi Fellowship, the Jacobs Fellowship, the ERC Starting Grant, the 2017 Pazy Memorial Award, the 2014 and 2017 Hershel Rich Technion Innovation Awards, the 2013 Sanford Kaplan Prize for Creative Management in High Tech, the 2010 Benin Prize, and six Technion Excellence in Teaching Awards. He is an Editor for the *Microelectronics Journal*.