# DIDACTIC: A Data-Intelligent Digital-to-Analog Converter with a Trainable Integrated Circuit using Memristors

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

***Abstract*— In an increasingly data-diverse world, in which data are interactively transferred at high rates, there is an ever-growing demand for high-precision data converters. In this paper, we propose a novel digital-to-analog converter (DAC) configuration that is calibrated using an artificial intelligence neural network technique. The proposed technique is demonstrated on an adaptive and self-calibrated binary-weighted DAC that can be configured on-chip in real time. We design a reconfigurable 4-bit DAC with a memristor-based neural network. This circuit uses an online supervised machine learning algorithm called "binary-weighted time-varying gradient descent." This algorithm fits multiple full-scale voltage ranges and sampling frequencies by iterative synaptic adjustments, while inherently providing mismatch calibration and noise tolerance. Theoretical analysis, as well as simulation results, show the efficiency and robustness of the training algorithm in reconfiguration, self-calibration, and desensitization, leading to a significant improvement in DAC accuracy: 0.12 LSB in terms of integral non-linearity, 0.11 LSB in terms of differential non-linearity, and 3.63 bits in terms of effective number of bits. The findings constitute a promising milestone toward scalable data-driven converters using deep neural networks.**

***Index Terms*— Adaptive systems, calibration, converters, memristors, neuromorphic computing, reconfigurable architectures, supervised learning.**

## I. Introduction

THE digital-to-analog converter (DAC) is a ubiquitous component that exists in every data-driven acquisition system and mixed-signal circuit. DACs are the link between the digital domain of signal processing and the real-world of analog transducers [1]. In modern VLSI circuit design, power consumption awareness and reliable computation constraints have rigorously paved the way towards hybrid analog–digital design methodologies. A key role of an efficient hybrid framework is a fast, robust, and ultra-low-energy DAC. Achieving both high resolution and speed is, however, challenging due to the effect of timing errors, jitters, and parasitic capacitance. Furthermore, the real limiter of achieving accurate CMOS-based data converters is device mismatches due to manufacturing process variations along the continuous technology scaling. These imperfections are poorly handled by current techniques due to their tremendous overhead. This trade-off between performance and reliability is a major bottleneck in data converter design, leading to special purpose designs and sophisticated custom techniques for specific applications [2], [3].

This paper investigates innovative approaches for digital-to-analog conversion by artificial intelligence (AI) techniques. A novel approach to design a generic, high-precision, high-speed, and energy-efficient DAC using artificial neural networks (ANNs) and neuromorphic computing [4] is proposed. The immense computational power of neuromorphic chips will encourage the interpolation of emergent collective characteristics into DAC design. Parallelism, simplicity, fault tolerance, and energy-efficiency are just a few examples of properties that would enhance conventional DAC circuits. Notably, the trainable, adaptive, and self-repairing capabilities of machine learning algorithms are considered novel intelligent features for next-generation DACs. These features, based on online predictions and cognitive decisions, will enable DACs to be self-reconfiguring, self-calibrating, and noise tolerant, utilizing the massive amount of correlated data to adapt to real-time variations and the running application specifications.

In the proposed ANN circuit, the promising technology of memristors is used to design synapses for the realization of artificial neural systems [5]. The small footprint, analog storage properties, low energy consumption, and non-volatility characteristics of memristors potentially offer brain-like density in integrated technologies for ANNs [6]. We leverage the use of memristors as synapses in a high-speed and ultra-low-power DAC [7] to achieve high precision, and a cost-effective reconfigurable, versatile, single-channel architecture.

The proposed binary-weighted DAC, which we term DIDACTIC, consists of an adaptive single-layer neural network, based on hybrid CMOS-memristor mixed-signal circuit design. The conductance of the memristors can be adjusted on-chip by a gradient descent training algorithm [8], [9]. The algorithm is capable of using different full-scale analog teaching signals and sampling frequencies to flexibly adjust the memristors' conductance online for general purpose DACs. The proposed DAC is also capable of self-calibrating device

Manuscript received May 14, 2017; revised September 9, 2017; accepted November 9, 2017. Date of publication December 6, 2017; date of current version April 3, 2018. This work was supported in part by the Israeli Planning and Budgeting Committee Fellowship, in part by the Viterbi Fellowship at the Technion Computer Engineering Center, and in part by EU COST Action IC1401. This paper was recommended by Guest Editor J.-S. Seo. *(Corresponding author: Loai Danial.)*

L. Danial, N. Wainstein, and S. Kvatinsky are with the Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 3200003, Israel (e-mail: sloaidan@tx.technion.ac.il).

S. Kraus is with PLSense Ltd., Yokneam 2066724, Israel.

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

Digital Object Identifier 10.1109/JETCAS.2017.2780251

mismatches and process variations. As a result, it alleviates non-linear distortions, tolerates noise, improves the effective number of bits (ENOB), and achieves almost ideal static figures of merit, *i.e.*, integral and differential non-linearity (INL and DNL, respectively), with approximately zero least significant bits (LSB).

The rest of this paper is organized as follows. In Section II, background on memristors and online gradient descent training is given. In Section III, DIDACTIC architecture, training algorithms, circuit design, and mechanisms of a four-bit DAC are described. In Section IV, the circuit operation and learning capability are evaluated, demonstrating a reconfigurable and self-calibrated DAC with online binary-weighted time-varying gradient descent training. In Section V, comparison, design-trade-offs and large-scale architectures are discussed. The paper is concluded in Section VI.

## II. Background

In this section, we present basic background information on the building blocks of this paper: a binary-weighted DAC, memristors, and the online gradient descent learning algorithm. We emphasize their correlated characteristics to simplify the understanding of the proposed device.

### A. Binary-Weighted DAC

The simplest type of DAC uses a binary-weighted architecture [1], where $N$ (number of bits) binary-weighted distributed elements (*e.g.,* current sources, resistors, or capacitors) are combined to provide a discrete analog output with finite resolution. The binary-weighted DAC is based on a simple and intuitive concept that utilizes the fundamental rule-of-thumb binary to decimal basis transformation. The direct conversion feature can be exploited for high-speed applications [1] because it uses a minimal number of conventional components and small die area. This DAC topology relies on the working principle of the inverting summing operational amplifier circuit, as shown in Fig. 1. Hence, the output voltage is the inverted sum of the input voltages, weighted by the ratio between the feedback resistor and the series resistance for each input. Digital inputs follow full-scale voltages, which means that logical '1' is equivalent to $V_{DD}$, and similarly, logical '0' is equivalent to 0 V. The LSB input is connected to the highest resistance value, which equals the feedback resistance $R$. Accordingly, the MSB input is connected to the lowest resistance value $R/2^{N-1}$. The other bits are correspondingly determined in a binary-weighted fashion. The resulting discrete voltage of the amplifier output is

$$V_{out} = -\frac{1}{2^N}\sum_{i=0}^{N-1} 2^i V_i, \tag{1}$$

where the minus sign is a result of the inverting operational amplifier, and $V_i$ is the digital voltage input of a bit with index $i$, after it has been attenuated by $2^N$, which is a normalization factor that fits the full-scale voltage. The output voltage is proportional to the binary value of the word $V_{N-1} \ldots V_0$.

Despite the simplicity of the binary-weighted DAC concept, critical practical shortcomings have hindered its realization.

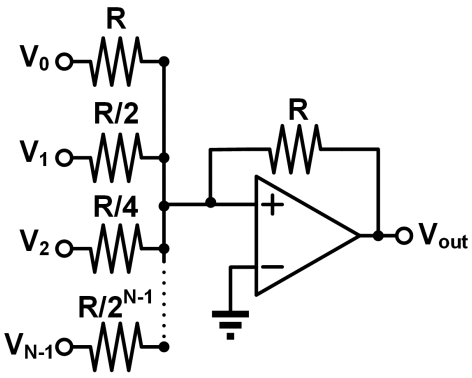


Fig. 1. Binary-weighted resistors based DAC.

The variability of the resistors, which defines the ratio between the MSB and LSB coefficients (dynamic range), is enormous and grows exponentially with the number of resolution bits, making accurate matching very difficult, and overwhelming a huge asymmetric area with power starved resistors, *e.g.*, for $N$ bits the ratio equals $2^{N-1}$. Furthermore, maintaining accurate resistance values over a wide range is problematic. In advanced submicron CMOS fabrication technologies, it is challenging to manufacture resistors over a wide resistance range and preserve an accurate ratio, especially in the presence of temperature variations. Process imperfections degrade the conversion precision and increase the vulnerability to mismatch errors, as listed in Table III in terms of INL, DNL, and ENOB.

Therefore, practical limitations, scalability drawbacks, and real-time variations are making the analog output $V_{out}$, determined by (1), non-deterministic and pushing binary-weighted DACs out of the band of interest of both high-speed and high-precision applications. Tremendous efforts have been invested in developing novel techniques using CMOS to eliminate mismatch errors; such techniques include self-calibration [10] or current steering [11]. Alternative architectures, *e.g.*, fully/partially segmented DACs, have emerged to achieve better accuracy and robustness [11].

### B. Memristors

The memristor was originally proposed in 1971 by Chua [12] as the missing fourth fundamental passive circuit element. Memristors are two-terminal analog passive devices with varying resistance, which changes according to a time integral of the current flowing through the device, or alternatively, the integrated voltage across the device. Memristors are non-volatile by definition, and in their physical realization, they are usually fabricated in the back-end-of-line (BEOL) of the CMOS process using only a few masks [13]. The relatively small size of physical memristors enables the integration of memory with computing elements, allowing compact, dense, and efficient parallel architecture for a variety of applications such as memory [14], analog circuits [15], logic design [16], machine learning algorithms [17] and ANNs [8]. The activation-dependent dynamics of memristors make them a promising feature for registering and updating synaptic weights. Thus, it is attractive to integrate memristors

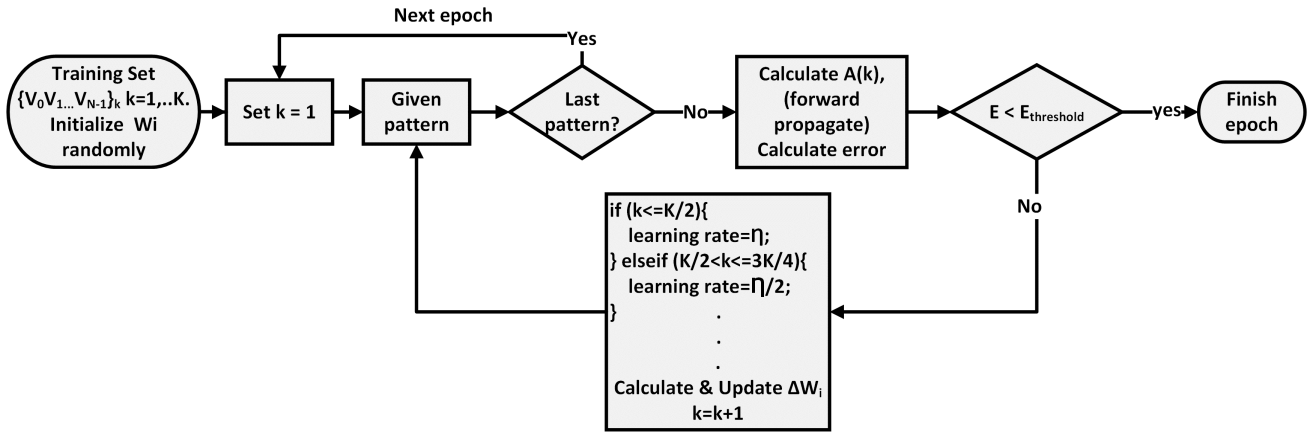


Fig. 2. Flow of the online binary-weighted time-varying gradient descent training algorithm, which updates the weights according to the error function.

as weighted elements in binary-weighted DACs [7] to achieve high energy-efficiency and scalability. In previous work [8], we proposed the following small signal analysis model of memristors, which perfectly describes the memory state derivation of synaptic weights:

$$i(t) = G\left(s(t)\right) v(t), \tag{2}$$

$$\frac{ds(t)}{dt} = f(v(t), s(t)), \tag{3}$$

where $s$ is a general state variable that evolves according to a function of voltage $f$. Eq. (2) is Ohm's law for a time-varying conductance $G$. Thus, the memristor enables an efficient design of trainable neural networks in hardware [8]. The VTEAM model [18] is used in this paper to accurately model the non-linear memristive behaviors in our design and evaluation.

### C. Online Gradient Descent Algorithm

The field of machine learning (ML) is dedicated to the study and implementation of systems that can learn from data, and evolve their cognitive ability to make crucial decisions based on a training phase. ANNs and neuromorphic computing are well-established infrastructures that apply brain-inspired learning rules to interpolate novel computational abilities [4], *e.g.*, adaptation and self-repair, beyond the conventional paradigm. In previous work [8], we analyzed a simple neural network topology (perceptron), comprising a single layer of binary inputs $V_i$, synapses $\boldsymbol{W_i}$ (decimal weights), and a single neuron. The neuron is considered the *de facto* neuroprocessing element that performs the following dot product of inputs and weights,

$$A = \sum_{i=0}^{N-1} W_i V_i, \tag{4}$$

where $A$ is an analog result of the digital inputs' weighted sum. From this stage, both deterministic and non-deterministic equivalence between (1) and (4) are derived. Thus, the discrete voltage of the DAC output as defined in (1) can be seen as a special case of a single-layer ANN, and (4) could be adjusted, using AI learning methods, to behave as a binary-weighted DAC with intrinsic variations. To the best of our knowledge, no neural based DAC has been developed in the past. We exploit the neural network's intelligent properties to achieve an adaptive DAC that is trained online by an ML algorithm.

Assume a learning system that operates on $K$ discrete trials, with $N$ digital inputs $V_i^{(k)}$, actual discrete output $A^{(k)}$ as in (4), and desired labeled output (*i.e.,* teaching signal) $t^{(k)}$. The weight $\boldsymbol{W_i}$ is tuned to minimize the following mean square error (MSE) of the DAC through the training phase

$$E = \frac{1}{2}\sum_{k=1}^{K}\left(A^{(k)} - t^{(k)}\right)^2. \tag{5}$$

A reasonable iterative update rule for minimizing (5) (*i.e.*, updating $\boldsymbol{W}$, where initially $\boldsymbol{W}$ is arbitrarily chosen) is the following online stochastic gradient descent iteration,

$$\begin{aligned}\Delta W_i^{(k)} &= -\eta\frac{\partial E}{\partial W_i^{(k)}} \Rightarrow \Delta W_i^{(k)} = -\eta\frac{\partial E}{\partial A^{(k)}}\cdot\frac{\partial A^{(k)}}{\partial W_i} \\ &\Rightarrow \Delta W_i^{(k)} = -\eta\left(A^{(k)} - t^{(k)}\right)V_i^{(k)},\end{aligned} \tag{6}$$

where $\eta$ is the *learning rate*, a small positive constant, and during each iteration $k$, a single empirical sample of the digital input voltage $V^{(k)}$, is chosen randomly. This learning algorithm, called *Adaline* or *LMS* [19], is widely used in adaptive signal processing and control systems [20]. Note that (6) is a local update rule, *i.e.*, the change in synaptic weight $\mathbf{W_i}$ depends only on the related components $A^{(k)}$, $t^{(k)}$, and $V_i^{(k)}$. This local update, which is ubiquitous in ANN training and ML algorithms, enables a massively parallel acceleration. The training phase continues until the MSE is less than $E_{threshold}$, a small predefined constant threshold that quantifies the learning accuracy. Due to the periodicity characteristic of the data used in digital-to-analog conversion, the convergence time scale is inversely proportional to the bit significance degree. For example, if the digital inputs are randomly generated, then the expectation of their corresponding teaching analog label will be in the middle of the analog dynamic range. The MSB, which divides the data range into two different sections, will begin its adjustment procedure towards the error gradient descent. In the same way, other

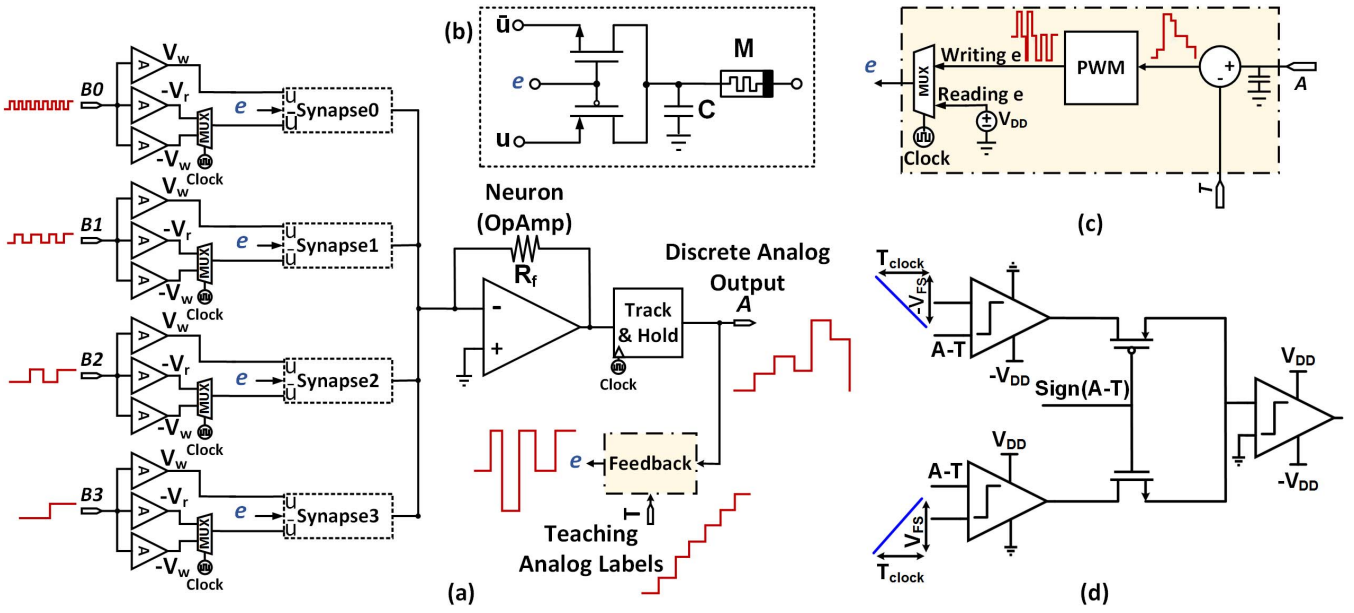


Fig. 3. (a) Schematic of a four-bit adaptive DAC based on a single layer ANN and binary-weighted synapses, trained online by a supervised learning algorithm executed by the feedback. (b) Schematic of the memristive synapse. (c) Feedback circuit for the gradient descent learning algorithm. (d) Schematic of the PWM circuit [9] that generates fixed amplitude pulses with a time width proportional to the subtraction product between the real and teaching signals.

bits will gradually begin their adjustment procedure later, after they converge to their relevant sections. Therefore, the training dynamics of less significant bits is complex and requires more time to be captured. Hence, the convergence time expectation is also binary-weighted distributed.

The LSB, which represents the most precise quantum, requires the longest resolution match and the lengthiest training time to converge. While the MSB can quickly achieve a stable value, the LSB may still present oscillations, thus continuously changing the collective error function in (5). Concurrently, the MSB will be disturbed and swing back and forth recursively in a deadlock around a fixed point. This problem is aggravated in the presence of noise and variations, and ameliorated by using smaller learning rates. Hence, we propose a slightly modified update rule to guarantee a global minimum of the error, and to fine-tune the weights proportionally to their significance degree. We call the modified rule the *binary-weighted time-varying gradient descent* learning rule, expressed as

$$\Delta W_i^{(k)} = -\eta(t)\left(A^{(k)} - t^{(k)}\right)\cdot V_i^{(k)}, \tag{7}$$

where $\eta(t)$ is a time-varying learning rate, decreasing in a binary-weighted manner along with the training time, as shown in Fig. 2. The expression for $\eta(t)$ is

$$\eta(t) = \begin{cases} \eta & if\ k \le K/2 \\ \eta/2 & if\ K/2 < k \le 3K/4 \\ \dots & \\ \eta/2^{N-1} & if\ (2^{N-1}-1)\cdot K/2^{N-1} < k \le (2^N-1) \\ & \quad \cdot K/2^N. \end{cases}$$

This learning rule utilizes the convergence time acceleration and the decaying learning rate to reduce bit fluctuations around a fixed point. In Section IV, we show that this learning rule is better than (6) in terms of training time duration, accuracy, and robustness to learning rate non-uniformity.

## III. Neural Network DAC Architecture

In this section, we present DIDACTIC, our proposed DAC architecture. We begin with a system overview, followed by the circuit design. Finally, the learning algorithm implementation is presented.

### A. System Overview

We leverage the conceptual simplicity, parallelism level, and minimum die size of the binary-weighted DAC architecture by implementing online gradient descent in hardware, thus achieving a reconfigurable, accurate, adaptive, and scalable DAC that can be used for high-speed, high-precision, and cost-effective applications. In Fig. 3(a), the proposed architecture for a four-bit neural network DAC is shown. As mentioned, the device is based on memristive synapses that collectively integrate through the operational amplifier, and a feedback circuit that regulates the value of the weights in real time according to (7). The architecture is composed of four synapse units, one neuron unit, and a synchronous training unit.

Depending on the characteristics and requirements of the application, a set of system parameters is determined. First, the sampling frequency $f_s$, which specifies the DAC speed, is determined, followed by the number of resolution bits $N$, which specifies the accuracy of the converter, and then the full-scale voltage $V_{FS}$, which specifies the DAC input dynamic range. Dynamic specifications concerning the learning rate are taken into consideration during the training phase, in order to

address further system requirements such as desired precision level, training time, and power consumption. All these specifications are defined for the peripheral circuitry according to the application requirements, and are certainly bounded by the technology. Voltage regulators for voltage scaling and phase locked loop (PLL) for frequency scaling are examples of such circuits [21].

The supervised learning algorithm is activated by interchangeable synchronous read and write cycles, utilizing the same execution path for both read and write operations *in situ*. Reading is the intended conversion phase; its final result is sampled at the end of the reading cycle $T_r$, after transient effects are mitigated, and it is latched by a negative-edge triggered latch for the entire writing cycle. The writing cycle $T_w$ activates the feedback circuit, which executes the learning algorithm, and compares the actual analog output of the DAC sampled at the end of the read cycle to the desired analog value, which is supplied by the peripheral circuit. During conversion, the training feedback is disconnected, leaving a simple low-power-consuming binary-weighted DAC [7]. It is preferable that the reading cycle be equal to the writing cycle, which will make it possible to capture the same non-deterministic behaviors, intrinsic noise, and environmental variations in the synaptic weights while training. Thus, the sampling frequency is

$$f_s = \frac{1}{T_r + T_w}. \tag{8}$$

### B. Artificial Synapse

For the synapse design, we adopt our previously proposed synapse circuit [8], which is composed of a single memristor, connected to a shared terminal of two MOSFET transistors (p-type and n-type), as shown in Fig. 3(b). The output of the synapse is the current flowing through the memristor. The synapse receives three voltage input signals: $u$ and $\bar{u} = -u$ are connected, respectively, to the source of both transistors, and the *enable* signal $e$ is connected to the gate of both transistors. The *enable* signal can have a zero value; then, both transistors are non-conducting, $V_{DD}$, when only the NMOS is conducting, or $-V_{DD}$ when only the PMOS is conducting. Thus, the magnitude of the input signal $u$ should be less than the minimum conductance threshold,

$$|u| < \min\left(V_{Tn}, \left|V_{Tp}\right|\right). \tag{9}$$

The synaptic weight is modified based on the value of $e$, which selects either input $u$ or $\bar{u}$. Thus, the writing voltage, $V_w$ (or $-V_w$), is applied via the source terminal of both transistors, and must be higher than the threshold voltage for memristive switching:

$$|V_{th,mem}| < |V_w| < \min\left(V_{Tn}, \left|V_{Tp}\right|\right). \tag{10}$$

In this paper, we use voltage controlled synapses, unlike the synapses in [8]. The read voltage $V_r$ must be sufficiently lower than the switching threshold of the memristor to prevent accumulative reads from disturbing the conductance of the memristor (*i.e.,* state drift) after several read operations. Hence,

$$|V_r| < \left|V_{th,mem}\right|. \tag{11}$$

A great advantage of these low read and write voltages is the resulting low-power consumption [7], and low subthreshold current leakage; high leakage would threaten the accuracy of the memristor. Voltages $V_w$ and $V_r$ are attenuated values of the digital DAC inputs that fit design constraints (10) and (11). Note that the right terminal of the memristor is connected to the virtual ground of the operational amplifier, whereas the left terminal of the memristor is connected to a transistor that operates in the ohmic regime as described in [8].

The assumption of ohmic operation is valid only if the conductance of the memristor is much smaller than the effective conductance of the transistor, as follows,

$$R_{mem}\left(s(t)\right) \gg \frac{1}{K\left(V_{DD} - 2\max(V_{T_n}, V_{T_p})\right)}, \tag{12}$$

where $K$ is a technology dependent constant that describes the transistor conduction strength, $V_{DD}$ is the maximum power supply, $s$ is the memristor internal state variable distributed between [0-1], and $R_{mem}$ refers to the memristor resistance as a function of the state variable $s$. The latter relationship is chosen to be linear [18]

$$R_{mem}(t) = s(t) \cdot (R_{OFF} - R_{ON}) + R_{ON}. \tag{13}$$

As a result, the memristor resistance level that could be achieved during training is lower bounded. Otherwise, the applied voltage over the memristor during the write cycle will not be sufficient to stimulate it. This constraint is achieved by the following condition:

$$|V_w| \frac{R_{mem,min}\left(s_{min}(t)\right)}{\frac{1}{K(V_{DD}-2V_T)} + R_{mem,min}\left(s_{min}(t)\right)} \geq \left|V_{th,mem}\right|. \tag{14}$$

The voltage division creates non-uniformity in the writing voltage of each cycle and will explicitly affect the learning rate. A shock absorption capacitor [9] was added to eliminate fluctuation spikes derived from either subthreshold leakage or high frequency switching. Its value is bounded by the sampling frequency of the DAC,

$$\frac{1}{K\left(V_{DD} - 2V_T\right)} C_{shock,max} \leq \frac{1}{f_s}. \tag{15}$$

### C. Artificial Neuron

The neuron is implemented by an operational amplifier with a negative feedback resistor $R$ [7]. It receives currents from $N$ memristors and sums them simultaneously, as follows:

$$A \approx -\sum_{i=0}^{N-1} \frac{R_f}{R_{mem_i}} V_i, \tag{16}$$

where $V_i$ is a read voltage via a memristor with index $i$, which represents the digital input value of the $i$-th bit. In the reading cycle, only the NMOS transistor is conducting since $e = V_{dd}$, with a negative read voltage to eliminate the inverting sign of the operational amplifier. The resolution of the DAC, which equals the minimal quantum, is defined by $r = V_{FS}/2^N$. The maximum analog output is achieved when the digital input '11...11' is inserted, and is equal to $A_{max} = \left(2^N - 1\right) V_{FS}/2^N$. Therefore, the read voltage equals

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 | $T_r$ | $5\ \mu s$ |
| Memristors | $V_{on/off}$ | $-0.3\ V, 0.4V$ | 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 | $L_{mem}$ | $3.7\ pH$ |
| | $R_{ON}$ | $2\ k\Omega$ | | | |
| | $R_{OFF}$ | $100\ k\Omega$ | | | |
| | $f(s)$ | $s \cdot (1-s)$ | | | |
| **DAC parameters** | | | **Learning parameters** | | |
| Sampling frequency | $f_s$ | 0.1MSPS | Maximum learning rate | max ($\eta$) | 0.01 |
| Number of bits | $N$ | 4 | Error threshold | $E_{threshold}$ | $2 \cdot 10^{-3}$ |
| Full-scale voltage | $V_{FS}$ | $[\frac{V_{DD}}{2} - V_{DD}]$ | | | |

$V_r = r = V_{FS}/2^N$, and it should obey the constraints in (11). Based on this read voltage, bounds on the number of resolution bits that the DAC could hold were formalized. From (11), we extract the minimal number of resolution bits,

$$N_{min} \geq \left\lceil \log_2 \left( \frac{V_{FS}}{\min\left(V_{T_n}, V_{T_p}\right)} \right) \right\rceil, \tag{17}$$

where the maximal number of resolution bits is bounded by the binary-weighted levels within the dynamic range of the memristor, $N_{max} \leq \log_2\left(\frac{R_{OFF}}{R_{ON}}\right)$. Because of the serial transistor resistance, however, it is undesirable to use surrounding levels. Doing so decreases the number of bits by $\log_2\left\lceil \frac{1}{\mathrm{R_{ON}} K (V_{DD} - 2V_T)} \right\rceil$, which is approximated to be zero in our case because $R_{ON} \gg 1/K(V_{DD} - 2V_T)$. Additionally, in the case of smaller full-scale voltage, some levels should be reserved. For example, if the full-scale voltage is half of the maximum power supply $V_{FS} = V_{DD}/2$, then the highest binary-weighted level should be reserved. Doing so will decrease the effective number of bits by $\left\lceil \log_2\left(\frac{V_{DD}}{\mathrm{V_{Fs,min}}}\right)\right\rceil$. The maximum number of bits that the proposed DAC could convert is up to

$$N_{max} \leq \log_2\left(\frac{R_{OFF}}{R_{ON}}\right) - \log_2\left\lceil \frac{1}{\mathrm{R_{ON}} K (V_{DD} - 2V_T)} \right\rceil - \left\lceil \log_2\left(\frac{V_{DD}}{\mathrm{V_{Fs,min}}}\right)\right\rceil. \tag{18}$$

In this case, if the minimal full-scale voltage is $V_{FS} = V_{DD}/2$, then the number of bits that could be converted by a DAC with the device parameters listed in Table I is at most four.

In the same context, the feedback resistor is upper-bounded by the minimal full-scale voltage and the highest resistance of the memristor,

$$R_f \leq \frac{R_{OFF} V_{FS}}{V_{DD}}, \tag{19}$$

when considering bi-directional variations of the training above and below the fixed resistance level, respectively. These variations are evaluated as $\pm 10\%$ of the nominal value.

### *D. Feedback Training Unit*

The feedback circuit is the main component for the execution of the binary-weighted time-varying gradient descent algorithm, which precisely regulates the synaptic adaptation procedure. Our aim is to design (7) in hardware and implement basic subtraction and multiplication operations. The subtraction discrete voltage product (namely, the error) is pulse modulated by a pulse-width modulator (PWM) with time width linearly proportional to the error and $\pm V_{DD}$, 0 $V$ pulse levels. As illustrated in Fig. 3(c), the PWM product is applied, via the feedback loop, to the synapse as an enable signal. The PWM [9], as shown in Fig. 3(d), is controlled by a clock that determines the maximum width of the product pulse. If $sign\,(A - T) \geq 0$, then the NMOS is conducting and the subtraction amplitude is compared to a positive ramp with full-scale voltage and clock cycle time width. Otherwise, the PMOS is conducting and the subtraction amplitude is compared to a negative ramp with full-scale voltage and clock cycle time width. As implied in (7), the learning rate is time varying. This is achieved by controlling the PWM with clocks with binary-weighted frequency multiples.

Therefore, the multiplication is invoked as an AND logic gate and controlled by the modulated enable signal, whereas the attenuated digital input is connected via the source of the synapse. The input is attenuated to obey the constraint in (10), as specified in Table I. The learning rate is a key factor of the adaptation performance: it depends on the circuit parameters listed in Table I, and on the write voltage, pulse-time width, feedback resistor, present state, and memristor device physical properties. The learning rate is

$$\eta(t) = \frac{\Delta R}{R} = \frac{(R_{OFF} - R_{ON})\,\Delta s(t)}{R_f}, \tag{20}$$

where $\Delta s$ is the change in the memristor internal state, and is defined as in the VTEAM model,

$$\Delta s = \int_0^{T_w} K_{on/off} \left( \frac{V_W}{V_{on/off}} - 1 \right)^{\alpha_{on/off}} \cdot f(s) dt, \quad (21)$$

where $K_{on/off}$, and $\alpha_{on/off}$ are constants that describe the state evolution rate and its nonlinearity, respectively, $V_{on/off}$ 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/$HfO_x$/Hf/TiN RRAM device with a buffer layer [22], with a high-to-low resistance state ratio of ~50 and low forming, set and reset voltages.

## IV. Evaluation

In this section, the proposed four-bit DAC design is discussed and evaluated in a SPICE simulation using a 0.18$\mu$ m CMOS process and the VTEAM memristor model [18]. First, the learning algorithm is evaluated in terms of mean square error (MSE) and training time. Then, a static evaluation of the circuit is described, and finally the dynamic evaluation is presented. The proposed DAC has been tested in both ideal and non-ideal cases. The circuit parameters, architectural specifications, and design constraints are listed in Table I.

### A. Ideal Case

The basic deterministic functionality of the four-bit DAC is demonstrated while being trained by the online gradient descent algorithm. Figure 4(a) shows the synapse resistive value where two sawtooth training datasets with different full-scale voltage ranges ($V_{DD}$, and $V_{DD}/2$) are applied successively in real time. It can be observed that the network converges from a random initial state to a steady state once the error determined by (5) is lower than $E_{threshold}$, after ~2000 training samples. Furthermore, it can be observed that when the full-scale voltage changes to $V_{DD}/2$, the system converges to a new steady state that quantizes 0.9V full-scale. In each case, the network is successfully reconfigured to operate under different specifications, as shown by different binary-weighted synaptic values in Fig. 4(a).

The DAC is next evaluated in terms of accuracy and training time, as illustrated in Fig. 4(b–d). The static evaluation in response to the DC ramp signal at three different time stamps is shown in Fig. 4(b) for (I) the initial state before training, (II) coarse-grained training (*i.e.,* where the error is slightly higher than $E_{threshold}$), and (III) fine-grained training (*i.e.,* where the error is low enough and the DAC response converges to the final, desired state). The teaching staircase in Fig. 4(b) is the same DC ramp input that statically evaluates the DAC at the three given time stamps. Therefore, the differences between two adjacent digital input codes within the actual DAC output are the differential non-linearity (DNL), and similarly, the total voltage differences between the actual DAC output and the desired staircase for each digital input code are the integral non-linearity (INL). Results of the DNL and INL are shown, respectively, in Fig. 4(c) and 4(d).

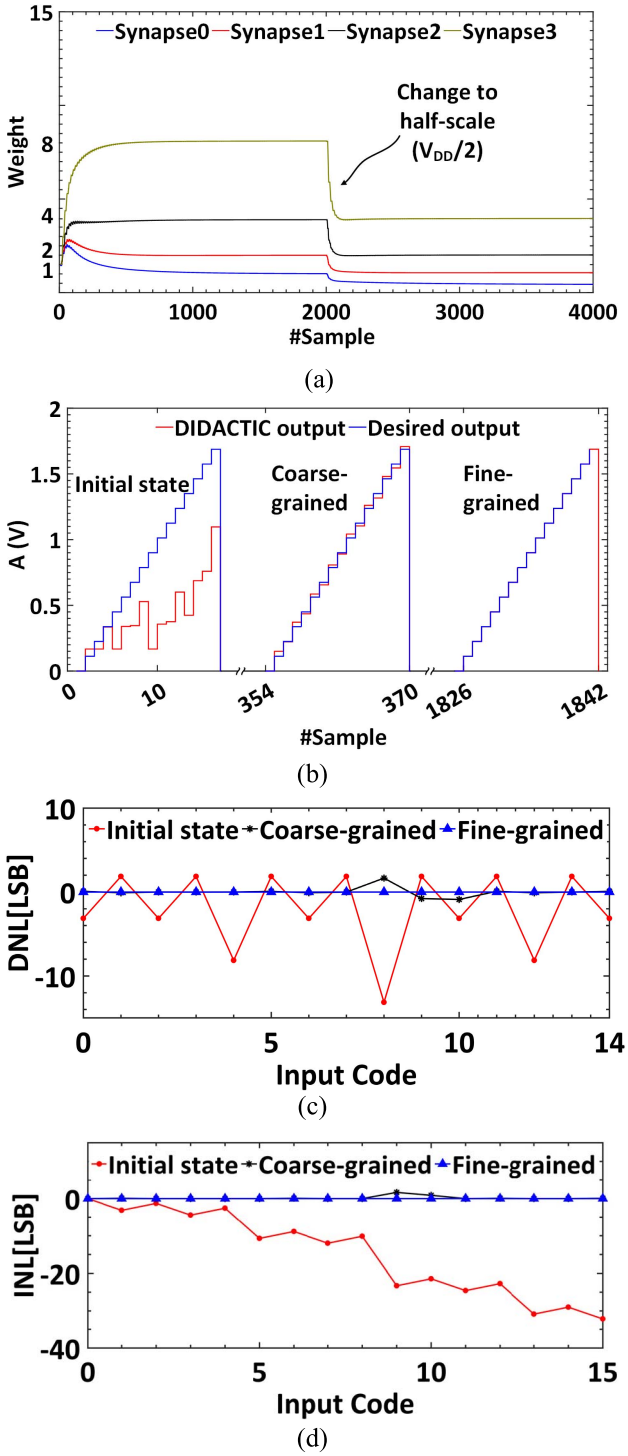


Fig. 4. (a) Binary-weighted synaptic adaptation during the training phase for the 1.8V full-scale output voltage range. Immediately, synapses are trained for the 0.9V full-scale output voltage range and shown in real time. (b) Comparison between the teaching dataset and the actual neural discrete analog DAC output at three different time stamps during the training; an identical staircase is achieved after the training is complete. (c) Differential and (d) integral non-linearities of the DAC at three different time stamps in response to the DC input voltage ramp.

As shown in Fig. 4(c–d), before the training state the DAC is completely non-linear and non-monotonic, with several missing codes. Thus, the maximum difference between the discrete

analog output and the ideal staircase, and the maximum difference between two adjacent analog levels, are considerably high: INL $\approx -32$ LSB, and DNL $\approx -13$ LSB. At the second time stamp (2ms $\sim$ 200 samples), however, the DAC performs better and appears monotonic but not sufficiently accurate (INL $\approx -1$ LSB, DNL $\approx -1$ LSB). After the training is complete (20ms), the DAC is fully calibrated: INL $\approx 0$ LSB, and DNL $\approx 0$ LSB. The fact that the DNL and INL are almost ideal proves that the training algorithm achieves maximum performance. The DAC also showed robustness when it was simulated with a randomly generated training dataset.

The improvements in static figures of merit significantly affect the dynamic figures of merit. The ENOB is a function of signal-to-noise and distortion ratio, whereas the distortions are a result of the DAC's non-linearity. If the DAC is non-linear (*e.g.*, INL, DNL$\neq$0), then harmonic distortion spurs will appear in the dynamic response, degrading the ENOB [1]. Therefore, improving the INL and DNL by learning and calibration techniques alleviates distortions and improves the ENOB, improving the precision of the DAC. To evaluate the ENOB, the DAC is dynamically evaluated and analyzed in response to a sine input with 40 $kHz$ frequency, which meets the Nyquist condition: $f_{input} \leq f_s/2$. The achieved ENOB in the ideal case is 3.71, which is almost ideal considering the intrinsic quantization error. All the extracted performance metrics are summarized in Table III.

TABLE II
CIRCUIT VARIATIONS & NOISE

| Type | Nominal value | Variance |
|---|---|---|
| **Device mismatch** | | |
| Resistor | $W = 2um$<br>$R = 50\Omega/$ | $\pm 0.5\% um$ |
| Capacitor | $W = 0.15um$<br>$C_A = 0.68fF/um^2$ | $\pm 1\% um$ |
| NMOS/PMOS | $W/L$<br>$V_T$ | $\pm$ 10%<br>$\pm 7\% V$ |
| Comparator | $V_{offset}$ | $\pm 5mV$ |
| Memristor | $V_{on/off}$<br>$K_{on/off}$<br>$R_{ON}$<br>$R_{OFF}$ | $\pm 10\% V$<br>$\pm 10\% mm/s$<br>$\pm 10\% \Omega$ |
| **Noise sources** | | |
| Thermal noise | $2kTg_1^{-1}$ | $10^{-16}V^2s$ |
| IR drop | $V_w$ | $\pm 10\% V$ |
| Pulse-width modulation noise | White noise | 50 $ps$ |
| Labels 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** | | |
| Input switching noise | $LdI/dt$ | $\pm 10\% V/\sqrt{Hz}$ |
| Opamp input noise | $1/f$ flicker noise | $10nv/\sqrt{Hz}$ |
| Slew rate | $2\pi f V_{FS}$ | $1.13V/ns$ |
| Memristor OFF impedance | $R_{OFF}$ | $\frac{R_{OFF}}{\sqrt{(1+(R_{OFF}C_{mem}\cdot 2\pi f)^2}}$ |
| Endurance degradation | $\Delta R$ | $10\%/decade$ |

### *B. Non-Ideal Case*

Usually, analog domains suffer from reduced robustness and vulnerability to noise and variations in comparison to their digital counterparts. DACs are being continuously pushed towards their performance limits as technology scales down and system specifications become more challenging. While device mismatch and process imperfections in modern DACs can be compensated for by calibration mechanisms [10], noise can irreparably degrade performance and is less straightforward to capture at design time. Several analysis methods have been established to estimate noise sources and their impact on the performance [23], [24]. All these mechanisms are specific and technology dependent, requiring exhaustive characterization, massive validation, and relatively long development time-to-market. Adaptive intelligent systems motivated by machine learning algorithms are, however, inherently robust to noise, which is a key element in the set of problems they are designed to solve. This suggests that the effects of intrinsic noise on the performance of the analog circuit are relatively small. Therefore, online training algorithms are not exclusive to reconfiguration, but can also be used for self-calibration, adaptation, and noise tolerance with generic standard methodology [8].

For this reason, a crude estimation of the magnitude of noise and variability has been extracted from [8], [23]–[30] and characterized as listed in Table II:

1. The process variation parameters for the memristor are pessimistically chosen, with a coefficient of variation (CV = standard deviation/mean $\sim$10%) to cover wide reliability margins [8]. The variability in the parameters of the memristors is equivalent either to corresponding changes in the synaptic weights or to the learning rate $\eta$. In Fig. 5, we show that the proposed binary-weighted time-varying gradient descent training algorithm is able to tolerate such process variations over time. The variability in the transistor parameters can also dramatically affect the learning performance; thus, transistor parameters such as $V_W$, $W/L$, *and* $V_T$ in Table I are chosen to guarantee a global optimal solution even in such extreme scenarios.
2. Noise sources include intrinsic thermal noise coming from the feedback resistor, memristor, and transistor [8], as well as pulse-width modulation noise, input referred noise [23], training label fluctuations as a result of supplier converter quantization noise [27], and frequency-dependent noise sources, which are quantified and estimated [28].
3. Frequency-dependent variations capture the parasitic capacitance and inductance of the memristor [29] and model it by a varying impedance as a function of the frequency. In addition, $\Delta$R degradation [30] along switching cycles as a result of oxide defects and device aging is considered.

While process variations determine the convergence time and accuracy, noise can cause the network to deviate from the optimum weights with destructive oscillations. In Fig. 5(a), the training processes for both gradient descent and the binary-weighted time-varying gradient descent with decaying learning rate are shown. Observe that the regular gradient descent, which succeeded in stabilizing the synapses without the presence of noise, now fails to stabilize the synapses.

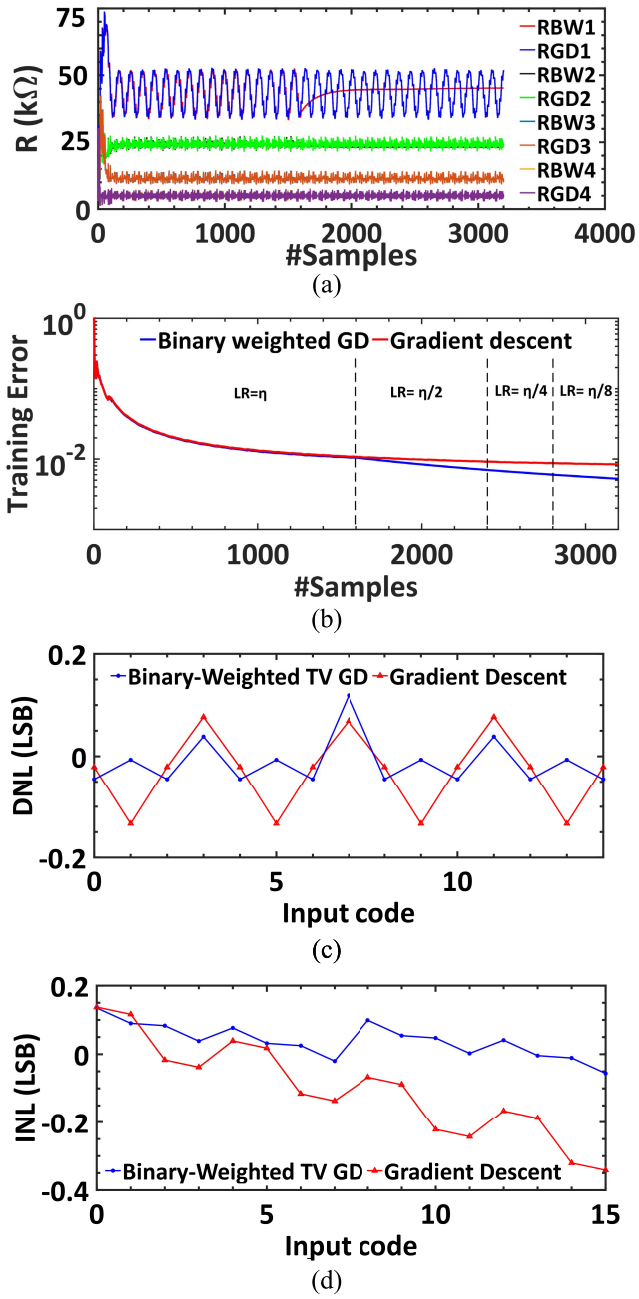


Fig. 5. Comparison between regular gradient descent (GD) and the proposed binary-weighted time-varying gradient descent (BW TV GD) algorithms in the presence of noise and process variations. (a) The GD failed to converge the synapses, whereas the BW TV GD succeeded and outperformed the GD with (b) smaller MSE, better (c) DNL, and (d) INL.

Conversely, the binary-weighted time-varying gradient descent with decaying learning rate successfully overcame noise and variations with stable synapses. The comparison is made, accordingly, in terms of MSE, DNL, and INL, as shown in Fig. 5(b–d), respectively. The switching non-linearity and threshold of the memristor device mitigate synaptic fluctuations derived from noise and variation sources. Nevertheless, the gradient descent algorithm fails to converge to a global optimum and keeps excessively capturing stochastic dynamics whereas the time-varying learning rate of the proposed algorithm enhances the network immunity against overfitting [31] and achieves reliable predictive performance on unseen data.

TABLE III
ACCURACY COMPARISON

| Type | Value |
|---|---|
| **Ideal case – Gradient descent** | |
| Maximum DNL | $\approx 0$ |
| Maximum INL | $\approx 0$ |
| ENOB | 3.71 |
| Training time | 20ms |
| MSE | $2 \cdot 10^{-3}$ |
| **Non-ideal case – Gradient descent** | |
| Maximum DNL | 0.15 LSB |
| Maximum INL | 0.38 LSB |
| ENOB | 3.18 |
| Training time | 30ms |
| MSE | $5 \cdot 10^{-3}$ |
| **Non-ideal case – Binary-weighted time-varying gradient descent** | |
| Maximum DNL | 0.11 LSB |
| Maximum INL | 0.12 LSB |
| ENOB | 3.63 |
| Training time | 30ms |
| MSE | $2 \cdot 10^{-3}$ |
| **Non-ideal case – Resistor-based DAC** | |
| Maximum DNL | 1.28 LSB |
| Maximum INL | 0.81 LSB |
| ENOB | 2.66 |

For robust validation of the DAC functionality in the presence of correlated variations and noise sources in Table II, we statistically analyzed the DAC performance for large numbers of randomly generated scenarios. We show the distribution of the achieved effective number of resistive levels in Fig. 6(a). The number of resistive levels, however, is finite and is a function of variations, data retention, noise margin, and amplifier sensitivity [32]. Fig. 6(a) shows that extreme cases where the write variation is $\pm 10\%$ and the comparator offset of the PWM is $\pm 5mV$ are less likely. Therefore, the effective number of resistive levels in the typical case (approximately 38% of the cases) is $\sim$64. The number of resistive levels has a key role in achieving such adaptive, self-calibrated, noise-tolerant, and highly accurate DACs. Due to its self-calibration capability, the DAC can tolerate variations and compensate for them by imposing a penalty of more training samples, as shown in Fig. 6(b). Alternately, fewer training samples or stable resistive levels are sufficient for lower accuracy, as shown in Fig. 6(c), in terms of ENOB, lower-bounded by five uniformly distributed binary-weighted levels covering a half- to full-scale voltage range. Results of the dynamic evaluation in terms of ENOB and training time in the typical case are listed in Table III, and compared to a resistor based binary-weighted DAC.

Endurance is an essential performance criterion of memristive devices for memory applications. Therefore, qualitative and pessimistically approximative analysis is done to evaluate the DAC's lifetime versus the increasing training time as a result of the memristor's endurance degradation. Endurance failure behavior is observed in Hf-based RRAM [30] and can be explained by different physical mechanisms that degrade its switching characteristics and high-to-low resistance ratio.

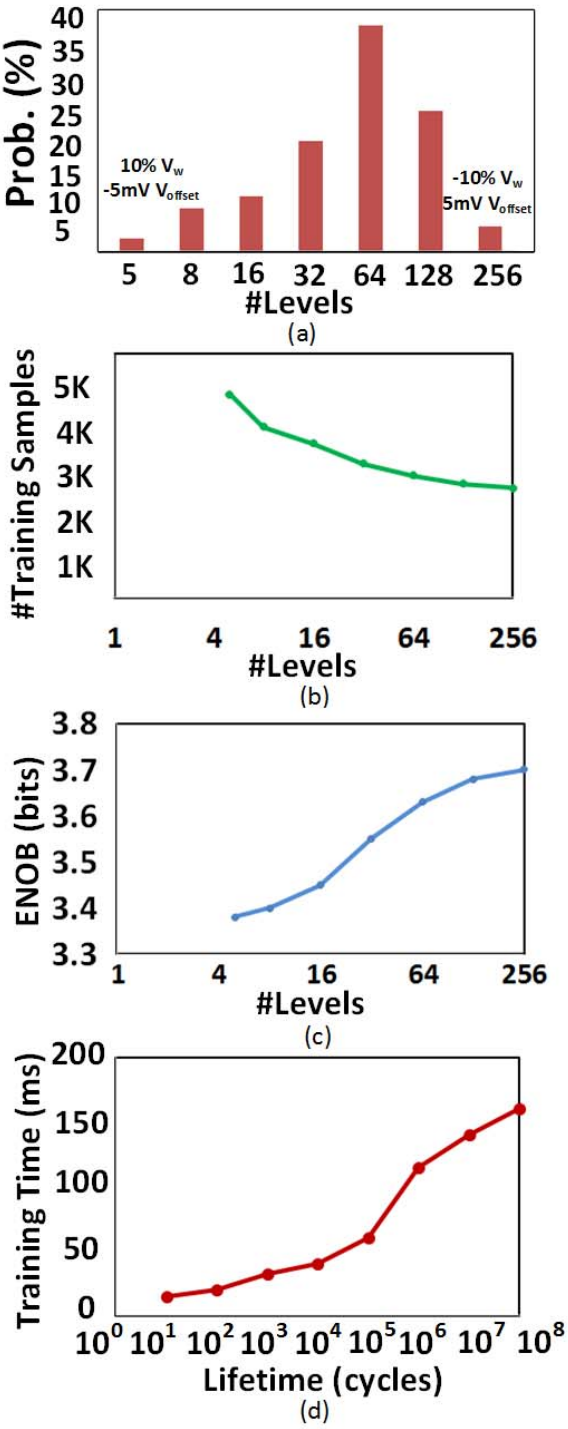


Fig. 6. (a) Statistical simulations of randomly generated variations and noise sources show the probability distribution of typical and extreme cases in terms of the effective number of resistive levels. (b) The impact of variations in the number of effective levels on the number of training samples in each case. (c) ENOB as a function of the number of stable resistive levels, where the minimum is five uniformly distributed binary-weighted levels. (d) Endurance degradation along device lifetime, in terms of full switching cycles, logarithmically affect $\Delta R$ in each training sample and are compensated for by the increasing training time for the whole epoch.

Among these mechanisms is the oxidation induced interface reaction, a result of high voltage/current during SET. The endurance of the fitted Pt/$HfO_x$/Hf/TiN is $\sim 8K$ cycles with 1.15 $V$ for SET and $-2.25$ $V$ for RESET, as observed in [22]. Decreasing operational voltages considerably improves the endurance while increasing the switching time of the device. According to the fitted parameters in Table I, the simulated switching time with $\pm V_w$is 75 $\mu s$ instead of the reported 400 $ns$ with 1.15 $V$ for SET, and 1 $ms$ instead of the reported 10 $\mu s$ with $-2.25$ $V$ for RESET [22]. The trade-off between write latency and endurance has been well-studied [33], and the relationship between them is formalized [34] as

$$Endurance \approx \left(\frac{t_{WP}}{t_0}\right)^{Expo\_factor}, \tag{22}$$

where $t_{WP}$ is write latency, $t_0$ is a device related constant, and Expo_factor is an empirical constant with a typical value of 2. Accordingly, the endurance of the device will increase to $8 \cdot 10^7$ cycles with the proposed writing voltage.

Due to the nature of the proposed DAC, it will continue training until it equals $E_{threshold}$ and achieves a high ENOB. Thus, the high-to-low resistance ratio degradation is not discernible, as it is compensated for by longer training times. A rough approximation, using logarithmic endurance degradation in time, is modeled by a 10% drop of $\Delta R$ per decade, as listed in Table II. The training time as a function of the number of switching cycles is shown in Fig. 6(d). To prove that the endurance is not a limitation for the proposed DAC, we estimate the number of training epochs until wear-out. As a pessimistic evaluation, we assume that every 1 $ms$ of training time equals a full RESET. This assumption is more aggressive for degradation than a total of 200 intermediate switches in 1 $ms$ [30]. Therefore, the maximum training time is 160 $ms$ and the corresponding minimal number of training epochs until wear- out is $\approx \frac{8 \cdot 10^7}{160} = 500K$. This finding implies that, in the worst case, the DAC could be reconfigured $\sim$150 times per day for $\sim$10 years either for new configuration or for calibration-only, depending on the running application [35].

The proposed DAC was simulated with different sampling frequencies $f_s$ to show its versatility and flexibility to adapt to different conditions that represent different specifications for different applications. At high frequency the memristor is modeled as a resistor in parallel to a capacitor and is connected in series with an inductance on each side [36]. The parasitic capacitance between electrodes of the memristor is dominant at high frequencies. As a result, the equivalent impedance of the memristor decays along the frequency. The values of the parasitic capacitance and inductance are listed in Table I. The maximum frequency at which the DAC can operate, $f_{max}$, is defined as the frequency at which the high-to-low-impedance ratio will not allow binary-weighted distribution of $N$-bits that covers the half- to full-scale voltage range:

$$\left|\frac{Z_{OFF}}{Z_{ON}}\right| \leq 2^{N+1}, \tag{23}$$

where $Z_{OFF}$ and $Z_{ON}$ are high and low impedance states, respectively. At the frequency-band of interest, $Z_{ON} \approx R_{ON}$, $Z_{OFF} \approx R_{OFF} || \frac{1}{2\pi j f_s C_{mem}} = \frac{R_{OFF}}{1+2\pi j f_s C_{mem} R_{OFF}}$, and the series inductance is negligible. By solving (22), we find

$$f_{max} = \frac{1}{2\pi R_{OFF} C_{mem}} \cdot \sqrt{\left(\frac{R_{OFF}}{R_{ON} \cdot 2^{N+1}}\right)^2 - 1}. \tag{24}$$

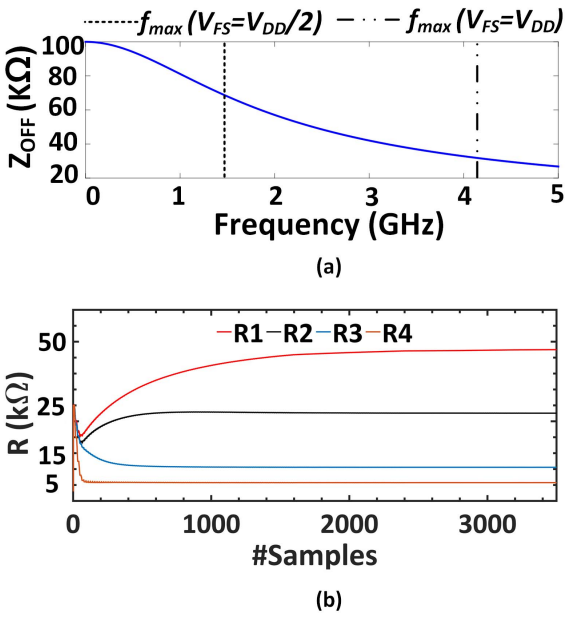


Fig. 7. (a) A high impedance state $Z_{OFF}$ as a function of sampling frequency; dashed lines indicate the maximum possible frequency bandwidth for a half- to full-scale voltage range with a high-to-low-impedance ratio of 32 and 16, respectively. (b) DAC reconfiguration for a 10MSPS sampling frequency, by continuous synaptic update. The frequency-dependent variations were captured by the synaptic weights.

The decay of $Z_{OFF}$ as a function of frequency is shown in Fig. 7(a), along with the maximum frequency bandwidth for different-scale voltages. In our case, for a four-bit DAC and full- to half-scale voltage range, $f_{max} = 1.668GHz$, which is below the transit frequency $f_T$ of 0.18 $\mu m$ CMOS transistors, the cutoff frequency of memristors [37], and the OpAmp slew rate.

The training dynamics are different in this case because the learning rate is a function of the pulse-width duration, which is a function of the sampling frequency. The higher the sampling frequency, the smaller the learning rate and the higher the number of training samples. Additionally, taking the frequency dependent variations into consideration, the synaptic weights are different and are able to absorb and compensate for these variations, as shown in Fig. 7(b) in response to the 10 MSPS sampling frequency. The frequency is 100× higher than 100 KSPS; as a result, the time interval for a single sample is 100× smaller, as is the learning rate. However, the total number of training samples until the error equals $E_{threshold}$is ∼1.5× greater, with ∼66× smaller training time (∼0.45ms). The ratios are not linear because the convergence time is different among the bits and not linear. This property proves that the DAC is a general-purpose device with a generic standard methodology.

## V. Expanding the DAC Design

As explained in the previous sections, a DAC is determined by its sampling frequencies and the number of resolution bits. These two specifications are challenging to achieve together in conventional DACs, and they are considered two major bottlenecks. We show an efficient mechanism that achieves optimal possible accuracy from the number of real allocated bits $N$ for each sampling frequency $f_s$. In Section III, we formalized the constraints on the number of bits in (17) and (18). Using these constraints and the design parameters listed in Table I, the maximum number of bits was at most four. This section discusses large-scale DACs by using the proposed four-bit DAC as a prototype that can be duplicated or cascaded to create a larger architecture. Interestingly, AI techniques that involve deep neural networks and backpropagation algorithms [8], [19] can be exploited and interpolated into the design of large-scale DACs that are based on the four-bit DAC.

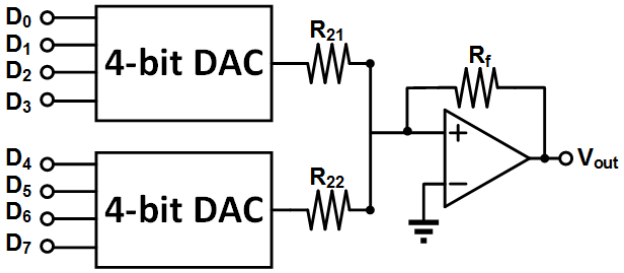


Fig. 8. An eight-bit reconfigurable DAC composed from two four-bit DACs by using a two-layer neural network.

For example, in Fig. 8, an eight-bit DAC that is based on the four-bit DAC is shown. The analog output of such a DAC is

$$\begin{cases} A_1 \approx -\sum_{i=0}^{3} \frac{R_f}{R_{mem_i}} V_i \\ A_2 \approx -\sum_{i=4}^{7} \frac{R_f}{R_{mem_i}} V_i \\ A_{tot} = W_{21} A_1 + W_{22} A_2. \end{cases} \tag{25}$$

where $W_{21}, W_{22}$ are the second-layer weights ($W_{2j} = R_f / R_{2j}^{j=1,2}$). Similarly to (5), the error function of the eight-bit deep neural network DAC is

$$E = \frac{1}{2} \sum_{k=1}^{K} \left( A_{tot}^{(k)} - t^{(k)} \right)^2. \tag{26}$$

The learning rules of the first layer synapses $W_{1i(0 \le i \le 7)}$ are extracted by using the error gradient descent and backpropagation algorithms

$$\begin{aligned} \Delta W_{1i(0 \le i \le 3)}^{(k)} &= -\eta \frac{\partial E}{\partial W_{1i}^{(k)}} = -\eta \frac{\partial E}{\partial A_{tot}^{(k)}} \cdot \frac{\partial A_{tot}^{(k)}}{\partial A_1^{(k)}} \cdot \frac{\partial A_1^{(k)}}{\partial W_{1i}^{(k)}} \\ &= -\eta W_{21} \left( A_{tot}^{(k)} - t^{(k)} \right) V_i^{(k)}, \end{aligned} \tag{27}$$

$$\begin{aligned} \Delta W_{1i(4 \le i \le 7)}^{(k)} &= -\eta \frac{\partial E}{\partial W_{1i}^{(k)}} = -\eta \frac{\partial E}{\partial A_{tot}^{(k)}} \cdot \frac{\partial A_{tot}^{(k)}}{\partial A_2^{(k)}} \cdot \frac{\partial A_2^{(k)}}{\partial W_{1i}^{(k)}} \\ &= -\eta W_{22} \left( A_{tot}^{(k)} - t^{(k)} \right) V_i^{(k)}. \end{aligned} \tag{28}$$

Using the same design methodology as for the four-bit DAC, this network defines a high precision eight-bit DAC with adaptive abilities to self-calibrate mismatches and tolerate variations. The weights in the second layer are fixed and predefined during design time; they do not need to be adjustable, and they do not obey the learning rule. Thus,

learning rules (27) and (28) depend on predefined parameters and do not vary during training as in multi-layer neural networks with a backpropagation algorithm [8]. The training data-set is given through and compared to the DAC output, which is the second layer output, and then the error product is back-propagated directly to the first layer synapses for both four-bit DACs simultaneously. Different learning rates are used for each four-bit DAC. Although resistors are highly prone to manufacturing variations, they can be used effectively for the second layer since the mismatches in that layer will be calibrated and compensated for by the weights of the first layer. Thus, the proposed large-scale concept will actually take advantage of the defects and handle them robustly. Furthermore, considering adjustable weights in the second layer will necessarily increase the design complexity of the training mechanism: its implementation will involve specific circuitry with higher area and power consumption, which may lead to undesired oscillations and wasteful training time.

A major challenge that directly relates to large-scale trainable DACs is how to generate the data-set for teaching. We assume that peripheral circuitry is provided and able to generate real-time data-sets with different specifications that fit the required DAC. Larger numbers of bits, smaller full-scale voltages, and higher frequencies, however, will be challenging for these circuits, which are not only technology dependent but also special purpose. For example, pulse-width modulators are bounded by the frequency with they can work. Therefore, the proposed binary-weighted time-varying gradient descent complicates the design but improves accuracy, compared to the regular gradient descent that uses a uniform learning rate. In future work, we will investigate general purpose peripheral circuitry that generates data-sets in real time.

## VI. Conclusions

We proposed a novel, reconfigurable and self-calibrating binary-weighted DAC that exploits the intelligent properties of an artificial neural network, and we demonstrated the equivalence between a single-layer neural network and a binary-weighted DAC. A supervised learning algorithm termed "binary-weighted time-varying gradient descent" was developed to train the network efficiently on-chip in real time to configure an adaptive high-precision four-bit DAC. A hybrid CMOS–memristor circuit design was proposed for the realization of the neural network. The learning algorithm successfully adjusted the memristors and reconfigured the DAC along with the full-scale voltage range and sampling frequency. It also successfully calibrated the DAC, improving its linearity and tolerating noise.

With an output range of 1.8 V, the training process improves the DNL and INL to 0.11 and 0.12 LSB, respectively, and improves the ENOB to 3.63 in the presence of noise and variations. To the best of our knowledge, this is the first neural-network-based DAC. We discussed how the proposed DAC might be used as a prototype for large-scale architectures, along with the corresponding challenges. Encouraged by the simplicity of the binary-weighted DAC architecture and its direct parallel conversion feature, we believe that this proof-of-concept will be a milestone with valuable results for large-scale data-driven converters, achieving high-precision, high-speed, cost-effective, and low-power consumption for general purpose applications.

## Acknowledgment

The authors would like to thank J. Sandrini and Y. Leblebici for their collaboration, and D. Soudry for his helpful comments.

## References

[1] R. J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*. Springer, 2013.
[2] 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.
[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] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.
[5] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.
[6] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, pp. 61–64, May 2015.
[7] L. Gao *et al.*, "Digital-to-analog and analog-to-digital conversion with metal oxide memristors for ultra-low power computing," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH)*, Jul. 2013, pp. 19–22.
[8] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, and S. Kvatinsky, "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.
[9] S. Greshnikov, E. Rosenthal, D. Soudry, and S. Kvatinsky, "A fully analog memristor-based neural network with online gradient training," in *Proc. IEEE Int. Conf. Circuits Syst.*, May 2016, pp. 1394–1397.
[10] D.-L. Shen, Y.-C. Lai, and T.-C. Lee, "A 10-bit binary-weighted DAC with digital background LMS calibration," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2007, pp. 352–355.
[11] S. Sarkar and S. Banerjee, "An 8-Bit 1.8 V 500 MSPS CMOS Segmented Current Steering DAC," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, May 2009, pp. 268–273.
[12] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.
[13] K.-H. Kim *et al.*, "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Lett.*, vol. 12, no. 1, pp. 389–395, Dec. 2012.
[14] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: Device characteristics and design implications," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2009, pp. 485–490.
[15] Y. V. Pershin and M. Di Ventra, "Practical Approach to Programmable Analog Circuits With Memristors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010.
[16] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman, "Memristor-based IMPLY logic design procedure," in *Proc. IEEE 29th Int. Conf. Comput. Design (ICCD)*, Oct. 2011, pp. 142–147.
[17] S. P. Adhikari, C. Yang, H. Kim, and L. O. Chua, "Memristor bridge synapse-based neural network and its learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 9, pp. 1426–1435, Sep. 2012.
[18] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 8, pp. 786–790, Aug. 2015.
[19] 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.
[20] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.
[21] S. C. Gupta, "Phase-locked loops," *Proc. IEEE*, vol. 63, no. 2, pp. 291–306, Feb. 1975.
[22] J. Sandrini, B. Attarimashalkoubeh, E. Shahrabi, I. Krawczuk, and Y. Leblebici, "Effect of metal buffer layer and thermal annealing on $HfO_x$-based ReRAMs," in *Proc. IEEE Int. Conf. Sci. Elect. Eng. (ICSEE)*, Nov. 2016, pp. 1–5.

[23] 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.
[24] T. Sepke, P. Holloway, C. G. Sodini, and H.-S. Lee, "Noise analysis for comparator-based circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 3, pp. 541–553, Mar. 2009.
[25] D. Niu, Y. Chen, C. Xu, and Y. Xie, "Impact of process variations on emerging memristor," in *Proc. 47th Design Autom. Conf. (DAC)*, vol. 10. Jun. 2010, pp. 877–882.
[26] M. Hu, H. Li, Y. Chen, X. Wang, and R. E. Pino, "Geometry variations analysis of $TiO_2$ thin-film and spintronic memristors," in *Proc. IEEE 16th Asia South Pacific Design Autom. Conf.*, Jan. 2011, pp. 25–30.
[27] R. M. Gray, "Quantization noise spectra," *IEEE Trans. Inf. Theory*, vol. 36, no. 6, pp. 1220–1244, Nov. 1990.
[28] Y. Nemirovsky, D. Corcos, I. Brouk, A. Nemirovsky, and S. Chaudhry, "1/f noise in advanced CMOS transistors," *IEEE Instrum. Meas. Mag.*, vol. 14, no. 1, pp. 14–22, Feb. 2011.
[29] A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, and R. S. Williams, "Sub-nanosecond switching of a tantalum oxide memristor," *Nanotechnology*, vol. 22, no. 48, p. 485203, Nov. 2011.
[30] B. Chen *et al.*, "Physical mechanisms of endurance degradation in TMO-RRAM," in *IEDM Tech. Dig.*, Dec. 2011, pp. 12.3.1–12.3.4.
[31] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surv.*, vol. 27, no. 3, pp. 326–327, Sep. 1995.
[32] H. Kim, M. P. Sah, C. Yang, and L. O. Chua, "Memristor-based multilevel memory," in *Proc. 12th Int. Workshop Cellular Nanosc. Netw. Appl. (CNNA)*, Feb. 2010, pp. 1–6.
[33] D. B. Strukov, "Endurance-write-speed tradeoffs in nonvolatile memories," *Appl. Phys. A, Solids Surf.*, vol. 122, no. 4, p. 302, Apr. 2016.
[34] L. Zhang, B. Neely, D. Franklin, D. Strukov, Y. Xie, and F. T. Chong, "Mellow writes: Extending lifetime in resistive memories through selective slow write backs," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2016, pp. 519–531.
[35] A. J. Gines, E. J. Peralias, and A. Rueda, "A survey on digital background calibration of ADCs," in *Proc. Eur. Conf. Circuit Theory Design*, Aug. 2009, pp. 101–104.
[36] N. Wainstein and S. Kvatinsky, "An RF memristor model and memristive single-pole double-throw switches," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2017, pp. 1–4.
[37] S. Pi, M. Ghadiri-Sadrabadi, J. C. Bardin, and Q. Xia, "Nanoscale memristive radiofrequency switches," *Nature Commun.*, vol. 6, Jun. 2015, Art. no. 7519.

**Loai Danial** received the B.Sc. degree in electrical engineering from the Technion–Israel Institute of Technology in 2014, where he is currently pursuing the Ph.D. degree with the Andrew and Erna Viterbi Faculty of Electrical Engineering. From 2013 to 2016, he was with IBM Laboratories as a Hardware Research Student. His current research combines the 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 pursuing the Ph.D. degree with 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 memristive devices.

**Shraga Kraus** received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Technion–Israel Institute of Technology, Haifa, Israel, in 2001, 2006, and 2011, respectively. Since 1999 he has been with several companies in the Israeli hi-tech industry, mainly on analogue and mixed signal integrated circuits. He is currently 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 in 2009 and 2010, respectively, and the Ph.D. degree in electrical engineering from the Technion–Israel Institute of Technology 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 the design of energy efficient architectures. He was a recipient of the 2015 IEEE Guillemin- Cauer Best Paper Award, the 2015 Best Paper of Computer Architecture Letters, a Viterbi Fellowship, a Jacobs Fellowship, an 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 of the *Microelectronics Journal*.