Logarithmic Neural Network Data Converters using Memristors for Biomedical Applications

Loai Danial, Kanishka Sharma, Shivansh Dwivedi, and Shahar Kvatinsky
Andrew and Erna Viterbi Faculty of Electrical Engineering,
Technion - Israel Institute of Technology, Haifa 3200003, ISRAEL, Email: sloaidan@campus.technion.ac.il

***Abstract*— Data converters are ubiquitous in electrical data driven systems, where they are heterogeneously distributed across the analog-digital interface. Unfortunately, conventional data converters trade off speed, power, and accuracy. Logarithmic analog-to-digital/digital-to-analog converters (ADC/DACs) are employed in biomedical applications where signals with high dynamic range are recorded. For the same input dynamic range of a linear ADC/DAC, a logarithmic one can efficiently quantize the sampled data by reducing the number of resolution bits, sampling rate, and power consumption, albeit with reduced accuracy for high amplitudes. Previously, we employed novel neural network architectures to design smart data converters that could be trained in real-time for general purpose applications, breaking through the speed-power-accuracy tradeoff, and using machine learning techniques and memristors for synaptic realization. In this paper, we report the results of SPICE simulations performed to train our converters to perform logarithmic quantization. The proposed architecture achieved a 77.19 pJ/conv FOM, 2.55 ENOB, 0.26 LSB INL, and 0.62 LSB DNL. These promising features will pave the way towards adaptive human-machine interfaces with continuous varying conditions for precision medicine applications.**

Keywords*— *Analog-to-digital/digital-to-analog conversion, biomedical applications, adaptive systems, memristors, machine learning, neuromorphic computing, logarithmic quantization.

I. Introduction

For several biomedical applications, such as cochlear implants [1], hearing aids [2], neural recording and stimulation [3-7], a nonlinear analog-to-digital converter (ADC) seems a more appealing choice for a signal processing system than a linear ADC. Audio signals, for example, are well-suited to log encoding because the human ear is less able to distinguish sound levels when the dynamic range of the signals is larger. The benefits of a nonlinear ADC include the ability to handle input signals with a large dynamic range [1-5], reduction of noise and data bit-rate [6], and compensation for nonlinear sensor characteristics [8]. A logarithmic ADC performs conversions with non-uniform quantization, where small analog amplitudes are quantized with fine resolution, while large amplitudes are quantized with coarse resolution.

Unfortunately, the intrinsic speed-power-accuracy tradeoff in linear ADCs is pushing them out of the application band of interest [9]. Furthermore, with the continuous downscaling of technology motivated by Moore's law, this tradeoff has become a chronic bottleneck of modern systems design due to deep sub-micron effects. Those effects are poorly handled by technology-dependent design techniques that overload data converters with enormous overhead, exacerbating the tradeoff and severely degrading their performance [9]. Moreover, conventional data converters lack design standards and are customized with sophisticated design flow. Thus, their architectures are optimized for special purpose applications, from high-speed to high-resolution to low-power [9]. These methods not only require exhaustive characterization and massive validation, but they are also expensive to develop, with a long time-to-market.

This paper takes a novel systematic approach to design data converters capable of reconfigurable quantization and logarithmic encoding. The converted data can be used, based on our previous work [9-11], to train the converter to autonomously adapt to the exact specifications, including quantization scale, of the running application and adjust to environmental variations, as shown in Fig. 1(a). This approach will reduce the time to market, efficiently scale with newer technologies, drastically reduce cost, standardize the design flow, and enable a generic architecture for general purpose applications. While practical applications require large-scale linear quantization, small-scale logarithmic quantization is sufficient. We believe that these promising features will substantially increase the number of adaptive converters in low-power wearable monitoring devices for precision medicine applications.

We propose a three-bit logarithmic ADC/DAC design. The proposed trainable data converters are based on memristive technology [9] and utilize machine learning (ML) algorithms to train an artificial neural network (ANN) architecture. With their synapse-like behavior, memristors are becoming increasingly prevalent in the design and realization of ANNs [10]. Their small footprint, analog storage properties, low energy consumption, and non-volatility allow them to mimic synapses, where the conductance of the memristor is considered as the synaptic weight [12].

The remainder of this paper is organized as follows. Section II provides background on the logarithmic data conversion theory, evaluation terminology and methodology. Section III presents the proposed ANN logarithmic data conversion and circuit design. In Section IV, the proposed trainable logarithmic data converters are evaluated. The paper is summarized in Section V.

II. Logarithmic Data Converters

A. Logarithmic ADC

An *N*-bit logarithmic ADC converts an analog input voltage (V_{in}) to an N-bit digital output code ($D_{out}=D_{N-1},\ldots,D_0$) according to a logarithmic mapping described by

$$\sum_{i=0}^{N-1} D_i 2^i = \frac{2^N}{c} log_B \left(\frac{V_{in}}{V_{FS}} B^c\right), \quad (1)$$

where *N* is the number of bits, *B* is the base of the logarithmic function (*e.g.*, 10), *C* is defined as the code efficiency factor [7], and V_{FS} is the full-scale analog input voltage range. Larger values of *C* result in more logarithmic conversion, capturing smaller signals and a higher dynamic range. Eq. (1) implies that the logarithmic ADC achieves good resolution for small input signals, but still allows coarsely quantized large input signals. Quantization noise is thus lower when the signal amplitude is small, and it grows with the signal amplitude. In contrast to an *N*-bit linear ADC, which has a fixed LSB size of $V_{FS}/2^N$, the LSB size of an N-bit logarithmic ADC varies with the input amplitudes, as shown

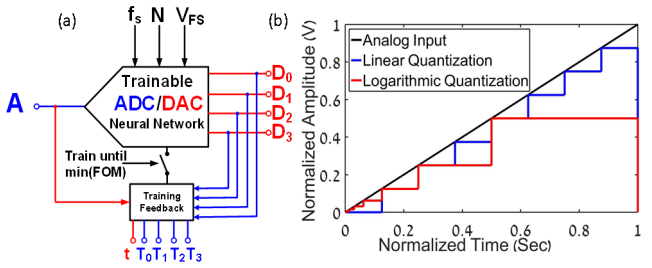

Fig. 1. (a) Scheme of the trainable four-bit ADC/DAC (blue/red path) neural network. The network receives f_s, V_{FS}, N, and is provided a specific teaching dataset T_i for real-time training. The training continues until the converter achieves the optimal FOM. (b) Characteristics of reconfigurable quantization: linear versus logarithmic.

in Fig. 1(b). For small input amplitudes, the LSB size is small and has a minimum value of

$$LSB_{min} = V_{FS}B^{-C}\left(B^{\frac{C}{2^N}} - 1\right), \quad (2)$$

when D_{out} changes from 0 to 1. For large input amplitudes, the LSB size is larger and has a maximum value of

$$LSB_{max} = V_{FS}\left(1 - B^{-\frac{C}{2^N}}\right), \quad (3)$$

when D_{out} changes from 2^N- 2 to 2^N-1. The dynamic range (DR) of an ADC is defined by the ratio of the maximum input amplitude to the minimum resolvable input amplitude:

$$DR(dB) = 20\log_{10}\left(\frac{V_{FS}}{LSB_{min}}\right) = 20\log_{10}\left(\frac{B^C}{B^{\frac{C}{2^N}} - 1}\right). \quad (4)$$

The DNL and INL for logarithmic ADC are defined similarly to the linear ADC except that in a logarithmic ADC the ideal step size varies with each step

$$DNL(j) = \frac{V_{j+1} - V_j}{LSB_{ideal}}, \quad (5)$$

$$INL(j) = \sum_{i=1}^{j} DNL(i), \quad (6)$$

where V_j and V_{j+1} are adjacent code transition voltages, and $j \in \{x | 1 <= x <= 2^N\text{-}2\}$.

The collective performance of the ADC could be quantified by the *FOM* as

$$FOM = \frac{P}{2^{ENOB} \cdot f_s}\left[\frac{J}{conv}\right], \quad (7)$$

which relates the ADC power dissipation during conversion, P, to its conversion rate in terms of sampling frequency, f_s, and effective number of resolution bits *(ENOB)*. The *ENOB is* calculated from the SNDR as

$$ENOB = \frac{SNDR(dB) - 1.76}{6.02}. \quad (8)$$

Originally, logarithmic ADCs were implemented by exploiting the inherent exponential I-V characteristics of a non-linear device in combination with a linear ADC [13]. Such devices include automatic gain control but cannot respond to rapidly fluctuating signals. Logarithmic amplifiers can solve this problem but require a look-up table to precisely describe the device-derived nonlinear characteristics [1]. Combining a back-end digital compander with a high-resolution ADC is also possible, but this method is power-hungry [1]. Alternatively, a logarithmic ADC can be realized by performing the required signal operations in the logarithmic domain with conventional ADC architectures such as pipeline [3][14], successive-approximation-register type [7], cyclic [8], or lookup table [15], and with potentially much lower power consumption. Reference scaling [14] is implemented in pipeline ADCs to convert complex math operations such as multiplication/division and exponents to simple addition/subtraction and scalar multiplication in the log domain.

B. Logarithmic DAC

An N-bit logarithmic DAC converts an N-bit digital input code (D_{in}) to an analog output voltage (V_{out}) according to a logarithmic (exponential) mapping described by

$$V_{out} = \frac{V_{FS}}{B^{2^N-1}} B^{\sum_{i=0}^{N-1} D_i 2^i}. \quad (9)$$

Exponential DAC, cascaded to a logarithmic ADC, is required to reproduce the linear analog input of the ADC. The INL, DNL, and ENOB for logarithmic DAC are defined as for the linear DAC, after activating a logarithmic transformation on V_{out}.

III. Logarithmic Neuromorphic Data Converters

In this section, we describe our proposed ANN logarithmic ADC/DAC. We show the architecture, theory, and an ML algorithm to train the networks.

A. Trainable Neural Network Logarithmic ADC

In previous work [9], we transformed the temporal binary search algorithm of a four-bit SAR to an optimized spatial single-layer neural network with binary-weighted synapses and pipelined forward-propagated neurons (MSB to LSB). We utilize the learning capabilities of ANNs, applying linear vector-matrix-multiplication and non-linear decision-making operations to train them to perform logarithmic quantization. Therefore, we formulate the logarithmic ADC equations in an ANN-like manner as follows, using three bits as an example,

$$\begin{cases} D_2 = u\left(V_{in} - 2^4 V_{ref}\right) \\ D_1 = u\left(V_{in} - 2^2 V_{ref}\overline{D_2} - 2^6 D_2\right) \\ D_0 = u\left(V_{in} - 2V_{ref}\overline{D_1 D_2} - 2^3 D_1\overline{D_2} - 2^5\overline{D_1}D_2 - 2^7 D_1 D_2\right) \end{cases}, \quad (10)$$

where V_{in} is the analog input and $D_2D_1D_0$ is the corresponding digital form (i=2 is the MSB), while each $\overline{D_i}$ is the complement of each digital bit, and each bit (neuron product) has either zero or full-scale voltage. $u(\cdot)$ is denoted as the signum neural activation function, and V_{ref} is a reference voltage equal to LSB_{min}. 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 between different neurons and their complement.

In a real-time operation, where non-ideal, stochastic, and varying conditions affect the conversion accuracy, the correct weights are not distributed deterministically in binary-weighted style as in (10). Rather, the weights should be updated in real-time *in situ* by a training mechanism. Four interconnected weights are needed to implement a three-bit logarithmic ADC. The interconnected synaptic weights of the network are described by an asymmetric matrix W, and each element W_{ij} represents the synaptic weight of the connection from pre-synaptic neuron j to post-synaptic neuron i. In the linear ADC case, i and j were bounded by the network dimensions, which are equal to N. However, in this case, where we have additional synaptic connections due to the AND product between neurons and their complements, the matrix dimensions approach $(2^{N-1} + 2)$.

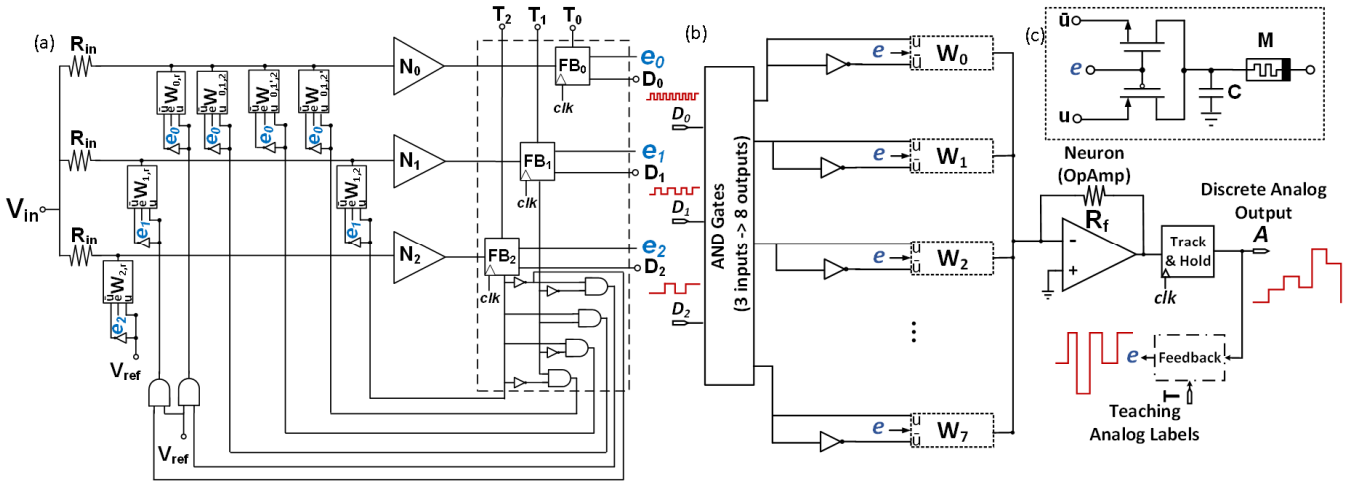

Fig. 2. Single-layer ANN three-bit logarithmic architectures, trained online using SGD, including synapses $W_{i,j}$, neurons N_i, and feedbacks FB_i of in addition to AND gates between neurons and their complements: (a) ADC with seven synapses, (b) DAC with eight synapses. (c) Schematic of the memristive synapse. Neuron and training feedback circuits in ADC and DAC, using digital circuits and PWM, respectively, are introduced in our previous work [9-10].

To train this network, W is tuned to minimize some measure of error (*e.g.*, MSE) between the estimated and desired labels, over a training set [9]. We use the online stochastic gradient descent (SGD) algorithm to minimize the error,

$$\Delta W_{ij(j>i)}^{(k)} = -\eta\left(T_i^{(k)} - D_i^{(k)}\right)T_j^{(k)}, \qquad (11)$$

where η is the *learning rate*, a small positive constant, and in each iteration k, a single empirical sample $V_{in}^{(k)}$ is chosen randomly and compared to a desired teaching label $T^{(k)}$. This update rule, common in ANN training and ML algorithms, allows massively parallel acceleration [12]. The training phase continues until the error is below $E_{threshold}$, a small predefined threshold that quantifies the learning accuracy [9].

B. Trainable Neural Network Logarithmic DAC

In previous work [10], we proposed a single-layer ANN with a single neuron to perform a four-bit linear DAC. Similarly, we formulate the logarithmic DAC equations in an ANN-like manner as follows, using three bits as an example,

$$\begin{aligned} V_{out} = {} & 2^0\overline{D_0}D_1\overline{D_2} + 2^1D_0\overline{D_1D_2} + 2^2\overline{D_0}D_1\overline{D_2} \\ & + 2^3D_0D_1\overline{D_2} + 2^4\overline{D_0}\overline{D_1}D_2 + 2^5D_0\overline{D_1}D_2 \\ & + 2^6\overline{D_0}D_1D_2 + 2^7D_0D_1D_2. \end{aligned} \qquad (12)$$

Thus, the logarithmic DAC is realized by a single-layer ANN with a linear neural activation output function and 2^N synapses. The DAC is trained using online SGD [10], with a time-varying learning rate and a teaching analog signal $t^{(k)}$

$$\Delta W_i^{(k)} = -\eta(t)\left(V_{out}^{(k)} - t^{(k)}\right)D_i^{(k)}. \qquad (13)$$

C. Circuit Design

The neural network ADC/DAC architectures and their building blocks, including neurons, synapses, and training feedbacks, are illustrated in Fig. 2. The synapse circuit design is adopted from [12]: a single memristor (M), and PMOS and NMOS transistors, with gates connected to a common enable input e. When $e = V_{DD}$ $(-V_{DD})$, the NMOS (PMOS) turns on and $\bar{u}$ (u) is passed to the output. When $e = 0$, neither input is passed, and the output is zero. The memristive crossbar (2T1R) inherently implements Ohm's and Kirchhoff's laws for ANN hardware realization. Our ADC/DAC was designed using a 0.18 µm CMOS process and memristors fitted by the VTEAM model [16] to a Pt/HfO$_x$/Hf/TiN RRAM device [17]. This device has a high-to-low resistance state (HRS/LRS) ratio of 50 to 1000. The aspect weight ratio of the ADC/DAC is equal to 2^{2^N-1} (for V_{FS}=V_{DD}/2). The HRS/LRS ratio sets an upper bound on the number of conversion bits.

TABLE I: CIRCUIT PARAMETERS

Parameter	Value	Parameter	Value
Power Supply		**Feedback resistor**	
V_{DD}	1.8 V	R_f	400 $k\Omega$
NMOS		**PMOS**	
W/L	10	W/L	20
V_{TN}	0.56 V	V_{TP}	-0.57 V
Memristor			
$V_{on/off}$	-0.3V, 0.4V	$R_{on/off}$	2 $k\Omega$, 1.5 MΩ
$K_{on/off}$	-4.8 mm/s, 2.8 mm/s	$\alpha_{on/off}$	3, 1
Reading voltage & time		**Writing voltage & time**	
V_r	$-0.1125\ V$	V_W	$\pm 0.5\ V$
T_r	5 μs	T_w	5 μs
Learning parameters		**3-bit ADC/DAC parameters**	
η	0.01	f_s	0.1 MSPS
$E_{threshold}$	$2 \cdot 10^{-3}$	V_{FS}	V_{DD}

For example, four-bit logarithmic ADC/DAC is infeasible using this device. Thus, we demonstrate a three-bit logarithmic ADC/DAC, which has better DR than a four-bit linear ADC/DAC [7]. Table I lists the circuit parameters.

Our neuron circuits are realized by an inverting OpAmp cascaded to a comparator (for ADC) [9-10]. Neuron values are multiplied using AND gates, added to the DAC and ADC in the frontend and backend, respectively. The online SGD algorithm is executed by the feedback circuit, which precisely regulates the synaptic reconfiguration. Our aim is to implement (11) and (13) and execute basic subtraction and multiplication operations. We used the same training circuits from [9-10]. While the feedback of the ADC is simple and realized by digital circuits, the feedback of the DAC is implemented by a pulse width modulator (PWM) with time proportional to the error and $\pm V_{DD}, 0\ V$ pulse levels [10]. After the training is complete ($E \leq E_{threshold}$), the feedback is disconnected from the conversion path.

IV. EVALUATION

Our proposed three-bit logarithmic ANN ADC/DAC design is simulated and evaluated using Cadence Virtuoso. First, the MSE and training time of the learning algorithm are evaluated. Next, the circuit is statically and dynamically evaluated, and finally power consumption is analyzed. Functionality and robustness were massively tested under extreme conditions using MATLAB. The design parameters are listed in Table I. Furthermore, circuit variations and noise sources are quantified and validated, as listed in [9].

TABLE II: PERFORMANCE EVALUATION

Metric	Logarithmic ADC	Linear ADC [9]
N	3 bits	4 bits
INL	0.26 LSB	0.4 LSB
DNL	0.62 LSB	0.5 LSB
DR	42.114 dB	24.08 dB
SNDR	17.1 dB	24.034 dB
ENOB	2.55	3.7
P	45.18 µW	100 µW
FOM	77.19 pJ/conv	0.136 nJ/conv
Training time	20ms	40ms
Metric	**Logarithmic DAC**	**Linear DAC [10]**
N	3 bits	4 bits
INL	0.163 LSB	0.12 LSB
DNL	0.122 LSB	0.11 LSB
Training time	80ms	30ms

The basic deterministic functionality of the three-bit logarithmic ADC/DAC is demonstrated during training by the online SGD algorithm. Figure 3(a) shows the resistive value of the synapses when a logarithmic ramp training dataset with full-scale voltage V_{DD} and sampling frequency f_s are applied in real time. After approximately 2000 training samples, which equals 20 ms training time for a 0.1 MSPS conversion rate, the MSE is below $E_{threshold}$ and the network converges from a random initial state to a steady state. In the same context, the convergence of digital output bits (neurons) converged to logarithmic codes is shown, at three time stamps, in Fig. 3(b-c).

We show that the proposed training algorithm compensates for variations by reconfiguring the synaptic weights. We statically evaluated how the proposed ADC responds to the DC logarithmic ramp signal. After training, the ADC is almost fully calibrated, monotonic, and accurate: INL≈0.26 LSB, and DNL≈0.62 LSB. It is then dynamically evaluated and analyzed, in response to an exponential sinusoidal input signal with $44\,kHz$ frequency where the harmonic distortions are mitigated, and the SNDR and ENOB improve as the training progresses. We also analyzed the power consumption, as specified in [9], during training until it reaches its minimum when the training is finished. The best energetic state of the network is achieved when it is configured in a logarithmic ADC manner.

The DAC is evaluated using similar methodologies as in [10]. We show that the proposed networks can also be trained to perform linear ADC/DAC using linearly quantized teaching data-sets. Table II lists the full performance metrics.

V. CONCLUSIONS

We proposed a novel logarithmic quantization of an ANN ADC/DAC that is trained online using the SGD algorithm, enabling reconfigurable quantization. A hybrid CMOS–memristor circuit design was proposed for the realization of a three-bit neural network ADC/DAC. The learning algorithm successfully adjusted the memristors and reconfigured the ADC/DAC along with the full-scale voltage range, quantization distribution, and sampling frequency.

Our simulations achieved a 77.19 pJ/conv FOM, exceeding the performance of a linear ADC. We believe that the proposed logarithmic quantization constitutes a milestone with promising results for emerging applications with varying conditions, such as low-cost wearable devices for biomedical applications. The design can also dramatically reduce data dimensions in ML and neuromorphic computing.

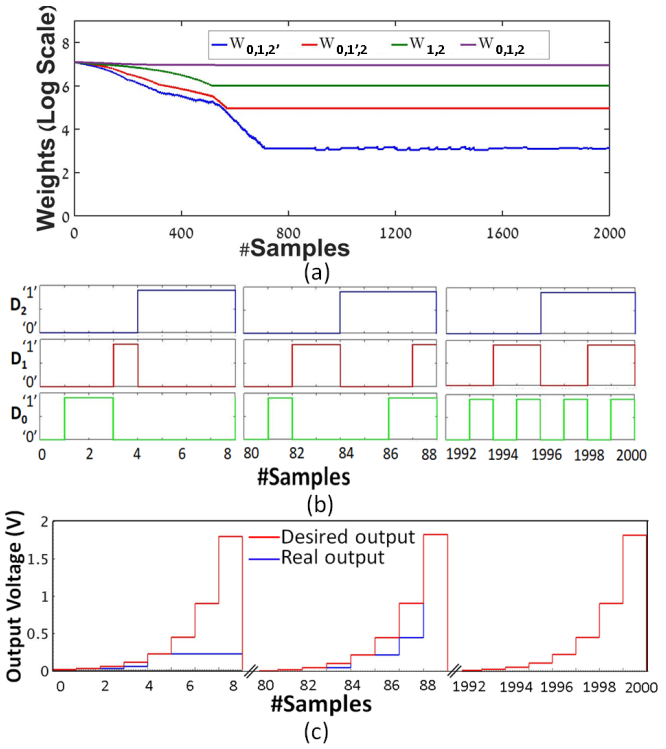

Fig. 3. Logarithmic ADC training evaluation. (a) Synapse reconfiguration (in log scale) during training for $N=3$, V_{FS}=1.8V and f_s=100KSPS. The weight is equal to the ratio between R_f and the corresponding memristor; thus, it has no units. (b) The actual digital outputs D_i (logical value) at three different time stamps during training; periodic outputs are obtained, corresponding to the logarithmic analog input ramp. (c) Comparison between the corresponding discrete analog values of the teaching dataset and the actual output; an identical logarithmic staircase is obtained after the training is complete.

ACKNOWLEDGMENTS

This research was partially supported by the Israeli PBC Fellowship, and by the Israeli Ministry of Science and Technology. The authors would like to thank R. Daniel and Y. Medan for the helpful consultancy.

REFERENCES

[1] J. J. Sit and R. Sarpeshkar, "A Micropower Logarithmic A/D with Offset and Temperature Compensation," *JSSC*, Vol. 39, No.2, pp. 308–319, 2004.

[2] J. Mahattanakul, "Logarithmic Data Converter Suitable for Hearing Aid Applications," *IET*, Vol. 41, No. 7, pp. 394 – 396, Mar. 2005.

[3] J. Lee *et al.*, "A 2.5 mW 80 dB DR 36 dB SNDR 22 MS/s Logarithmic Pipeline ADC," *JSSC*, Vol.44, No.10, pp. 2755–2765, 2009.

[4] J. Lee *et al.*, "A 64 Channel Programmable Closed-Loop Neurostimulator with 8 Channel Neural Amplifier and Logarithmic ADC," *JSSC*, Vol. 45, No.9, 2010.

[5] H. Rhew *et al.*, "A Fully Self-Contained Logarithmic Closed Loop Deep Brain Stimulation SoC with Wireless Telemetry and Wireless Power Management," *JSSC*, Vol. 45, No.10, 2014.

[6] M. Judy *et al.*, "Nonlinear Signal-Specific ADC for Efficient Neural Recording in Brain-Machine Interfaces," *TBME*, Vol. 8, No. 3, pp. 371-381, June 2014.

[7] Y. Sundarasaradula *et al.*, "A 6-bit, Two-Step, Successive Approximation Logarithmic ADC for Biomedical Applications," *ICECS*, pp. 25-28, 2016.

[8] A. Thanachayanont, "A 1-V, 330-nW, 6-Bit Current-Mode Logarithmic Cyclic ADC for ISFET-Based pH Digital Readout System," *CSSP*, pp.1405-1429, 2015.

[9] L. Danial *et al.*, "Breaking Through the Speed-Power-Accuracy in ADCs using a Memristive Neuromorphic Architecture," *TETCI,* Vol. 2, No.5, pp. 396-409, Oct. 2018.

[10] L. Danial *et al.,* "DIDACTIC: A Data-Intelligent Digital-to-Analog Converter with a Trainable Integrated Circuit using Memristors," *JETCAS*, Vol. 8, No. 1, pp. 146-158, March 2018.

[11] L. Danial and S. Kvatinsky, "Real-Time Trainable Data Converters for General Purpose Applications", NANOARCH, July 2018.

[12] D. Soudry *et al.*, "Memristor-Based Multilayer Neural Networks with Online Gradient Descent Training," *TNNLS,* Vol. 26, No. 10, pp. 2408-2421, Oct. 2015.

[13] S. Cantarano and G. V. Pallottino, "Logarithmic Analog-to-Digital Converters: A Survey," *IMS*, Vol. 22, No. 3, pp. 201-213, Sept. 1973.

[14] J. Guilherme and J. E. Franca, "New CMOS logarithmic A/D Converters Employing Pipeline and Algorithmic Architectures," *ISCAS*, pp. 529-532, 1995.

[15] Shih-Fang Chen *et al.*, "Logarithmic CMOS Image Sensor Through Multi-Resolution Analog-to-Digital Conversion," *VLSI-TSA*, pp. 227-230, 2003.

[16] S. Kvatinsky *et al.*, "VTEAM: A General Model for Voltage-Controlled Memristors," *TCASII,* Vol. 62, No. 8, pp. 786–790, August 2015.

[17] J. Sandrini *et al.*, "Effect of Metal Buffer Layer and Thermal Annealing on HfOx-based ReRAMs," *ICSEE*, pp. 1–5, November 2016.