

B. TECH. PROJECT REPORT

On

LARGE-SCALE ARCHITECTURES OF NEUROMORPHIC DATA CONVERTERS

BY

KANISHKA SHARMA



**DISCIPLINE OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

December 2019

LARGE-SCALE ARCHITECTURES OF NEUROMORPHIC DATA CONVERTERS

A PROJECT REPORT

*Submitted in partial fulfillment of the
requirements for the award of the degrees*

of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

Submitted by:

KANISHKA SHARMA

Guided by:

Prof. Shahar Kvatinsky, PhD, MBA (off-campus)

&

Dr. Santosh Kumar Vishvakarma, PhD (on-campus)



INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2019

CANDIDATE'S DECLARATION

I hereby declare that the project entitled “**LARGE-SCALE ARCHITECTURES OF NEUROMORPHIC DATA CONVERTERS**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Electrical Engineering’ completed under the supervision of **Prof. Shahar Kvatinsky, Associate Professor, Electrical Engineering, Technion - Israel Institute of Technology** and **Dr. Santosh Kumar Vishvakarma, Associate Professor, Electrical Engineering, IIT Indore** is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student with date

CERTIFICATE by BTP Guide

It is certified that the above statement made by the students is correct to the best of my knowledge.

Supervisor

Dr. Santosh Kumar Vishvakarma

Associate Professor,

Indian Institute of Technology Indore

Date:

Preface

This report on “Large-Scale Architectures of Neuromorphic Data Converters” presents my research during the period from May 2019 to November 2019 at the ASIC² Technion Research Group, Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Israel.

Through this report, I have tried to give detailed designs of the novel architectures of neuromorphic data converters. I have tried to the best of my abilities and knowledge to explain the content in a coherent and lucid manner.

Kanishka Sharma

B.Tech. IV Year

Discipline of Electrical Engineering,

IIT Indore

Acknowledgements

This research project would not have been possible without the support of many people. They deserve my deepest gratitude and warmest thanks.

I am heartily thankful to my supervisors, Prof. Shahar Kvatinsky and Dr. Santosh Kumar Vishvakarma, for their guidance and the opportunity to work on this project. I would like to express my special gratitude to my mentor, Loai Danial, for the inspiration, support, and offering a new approach to problems. I also want to thank Eric Herbelin for always helping with the technical and non-technical aspects of being a visitor student.

I extend this opportunity to thank all the members of the ASIC² Technion Research Group and the VLSI Lab, IIT Indore. I would like to thank Gaurav Singh for always offering his knowledge and assistance. Many thanks to all my friends for enriching my out-of-work life during this project, especially Anmol Jain, Kapil Jain and Shivansh Dwivedi.

Above all, I am indebted to my parents and my family for their unconditional love, encouragement, support, and motivation at every stage of my personal and academic life.

Kanishka Sharma

B.Tech. IV Year

Discipline of Electrical Engineering,

IIT Indore

Abstract

With the advent of high-speed, high-precision, and low-power mixed-signal systems, there is an ever-growing demand for accurate, fast, and energy-efficient analog-to-digital (ADCs) and digital-to-analog converters (DACs). Unfortunately, with the downscaling of CMOS technology, modern ADCs trade-off speed, power and accuracy. Recently, memristive neuromorphic architectures of four-bit ADC/DAC have been proposed. Such converters can be trained in real-time using machine learning algorithms, to break through the speed-power-accuracy trade-off while optimizing the conversion performance for different applications. However, scaling such architectures above four bits is challenging.

In the first half of this thesis, I describe our proposed scalable and modular neural network ADC architecture based on a pipeline of four-bit converters, preserving their inherent advantages in application reconfiguration, mismatch self-calibration, noise tolerance, and power optimization, while approaching higher resolution and throughput in penalty of latency. SPICE evaluation shows that an 8-bit pipelined ADC achieves 0.18 LSB INL, 0.20 LSB DNL, 7.6 ENOB, and 0.97 fJ/conv FOM. This work presents a significant step towards the realization of large-scale neuromorphic data converters.

In the later half, I describe our proposed neuromorphic logarithmic ADC/DAC. Logarithmic ADC/DAC 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. The proposed architecture achieves 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.

List of Publications

Conference papers:

1. L. Danial, **K. Sharma**, and S. Kvatinsky, “A Pipelined Memristive Neural Network Analog-to-Digital Converter” IEEE International Symposium on Circuits and Systems (ISCAS), Spain, May 2020 (under review).
2. L. Danial, **K. Sharma**, S. Dwivedi, and S. Kvatinsky, “Logarithmic Neural Network Data Converters using Memristors for Biomedical Applications”, *Proceedings of the IEEE Biomedical Circuits and Systems (BioCAS)*, Japan, October 2019.

Table of Contents

Candidate's Declaration	5
Supervisors' Certificate	5
Preface	7
Acknowledgements	9
Abstract	11
List of Publications	13
Table of Contents	15
List of Figures	17
List of Tables	19
Abbreviations	21
1. Introduction	23
2. Background	25
2.1 Memristor	25
2.1.1 The VTEAM Memristor Model	25
2.1.2 HfOx based Memristor	25
2.2 Artificial Synapse	26
2.3 Neuromorphic ADC	26
2.4 Neuromorphic DAC	28
2.5 ADC Performance Metrics	29
3. Scaling Challenges	31
4. Memristive Pipelined Neuromorphic Analog-to-Digital Converter	33
4.1 Introduction to Pipelined ADC	33
4.2 Pipelined Neuromorphic ADC	34
4.2.1 Neural Network Architecture	34
4.2.2 Training Framework	37
4.2.3 Performance Evaluation	39

4.2.4	Performance Comparison	42
4.2.5	Scalability Evaluation	42
5.	Logarithmic Neuromorphic Data Converters	43
5.1	Introduction	43
5.1.1	Applications of Logarithmic Data Converters	43
5.1.2	Logarithmic ADC	44
5.1.3	Logarithmic DAC	45
5.2	Trainable Neural Network Logarithmic ADC	45
5.3	Trainable Neural Network Logarithmic DAC	46
5.4	Circuit Design of Neural Network ADC/DAC	46
5.5	Performance Evaluation	48
6.	Conclusion	51
	References	53

List of Figures

Fig. 2.1	Schematic of the Artificial Synapse consisting of two transistors and a memristor	26
Fig. 2.2	Neural network-based 4-bit ADC architecture trained online using SGD, including synapses $W_{i,j}$, neurons N_i , and feedback FB_i	27
Fig. 2.3	Schematic of a memristive synapse connected to an artificial neuron implemented as an inverting opAmp for integration and a comparator for decision making	28
Fig. 2.4	Neural network-based 4-bit binary-weighted DAC architecture, including synapse W_i , an artificial neuron implemented as an inverting opAmp, and a PWM-based feedback circuitry for the time-varying SGD	29
Fig. 4.1	General concept of pipelining	33
Fig. 4.2	Schematic of a three-stage conventional pipelined ADC	33
Fig. 4.3	Proposed architecture of a two-stage pipelined ADC trained online using SGD	34
Fig. 4.4	SPICE design of the memristive neural network pipelined ADC	36
Fig. 4.5	SPICE design of the four-bit neural network sub-ADC	36
Fig. 4.6	SPICE design of the four-bit neural network DAC	37
Fig. 4.7	Training dataset of the sub-ADCs	38
Fig. 4.8	Pipeline ADC training evaluation	40
Fig. 4.9	Variation of synaptic weights of the sub-ADC and the DAC during training showing self-reconfiguration when the full-scale voltage and sampling frequency are changed	41
Fig. 5.1	Characteristics of reconfigurable quantization: linear versus logarithmic	43
Fig. 5.2	Architecture of the proposed 3-bit logarithmic neural network ADC and DAC	47
Fig. 5.3	Logarithmic ADC training evaluation	49
Fig. 5.4	DNL and INL plots for the logarithmic ADC	50

List of Tables

TABLE I.	Scaling Challenges in Neuromorphic ADC	31
TABLE II.	Pipelined ADC Circuit Parameters	38
TABLE III.	Performance Comparison	42
TABLE IV.	Scalability Evaluation	42
TABLE V.	Log ADC/DAC Circuit Parameters	48
TABLE VI.	Log ADC/DAC Performance Evaluation	50

Abbreviations

ADC	Analog-to-Digital Converter
ANN	Artificial Neural Network
CMOS	Complementary Metal Oxide Semiconductor
CPG	Coherent Power Gain
DAC	Digital-to-Analog Converter
DNL	Dynamic Non-Linearity
DR	Dynamic Range
ENBW	Equivalent Noise Bandwidth
ENOB	Effective Number of Bits
FFT	Fast Fourier Transform
FOM	Figure-of-Merit
HRS	High Resistance State
INL	Integral Non-Linearity
LMS	Least Mean Square
LRS	Low Resistance State
LSB	Least Significant Bit
ML	Machine Learning
MSB	Most Significant Bit
MSE	Mean Square Error
PWM	Pulses-Width-Modulation
RRAM	Resistive Random-Access Memory
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise Ratio
SNDR	Signal-to-Noise-plus-Distortion Ratio
VTEAM	Voltage Threshold Adaptive Memristor

Chapter 1

Introduction

Data Converters are ubiquitous in modern mixed-signal systems and emerging data-driven applications. These modern systems demand accurate and reliable conversion performance. However, the analog performance in advanced technology nodes is severely degraded due to reduced signal-to-noise ratio (SNR), low intrinsic gain, device leakage, and device mismatch [1]. These deep-submicron effects exacerbate the intrinsic speed-power-accuracy tradeoff in the analog-to-digital converters (ADCs), which has become a chronic bottleneck of modern system design [2]. Moreover, these effects are poorly handled with specific and time-consuming design techniques for special purpose applications, resulting in considerable overhead and severely degrading their performance [2].

The complexity of the construction of data converters in smaller feature size, combined with the demand for flexible architectures by modern systems is creating a vacuum for novel computing paradigms [3]. Neuromorphic computing suggests one such intriguing approach which can adaptively perform big amount of energy-efficient operations in parallel, such as pattern recognition [4]. Notably, analog-to-digital conversion can be seen as an example of simple pattern recognition, where the analog input can be classified into one of the 2^N different patterns for N bits, and thus can be readily solved using artificial neural networks (ANNs). Furthermore, the calibration process of these networks can be viewed as modification of neural parameters based on the measured error calculated during learning.

Recently, neuromorphic architectures of four-bit ADC and DAC have been proposed [2], [5]. Such reconfigurable converters can be trained *in-situ* in real-time using machine learning (ML) algorithms to autonomously calibrate for device mismatch, noise tolerance, and power optimization. Chapter 2 briefly describes the building-blocks and architectures of these data converters. Chapter 3 discusses the challenges in scaling the Neural Network ADC architecture. In Chapter 4, the proposed memristive pipelined neural network ADC [6] architecture is described along with its performance comparison and scalability evaluation. In Chapter 5, the proposed logarithmic neural network data converters [7] are described, and the report is finally concluded in Chapter 6.

Chapter 2

Background

2.1 Memristor

The idea of memristive devices was proposed by L. Chua in 1971 [8]. They are two-terminal passive circuit elements that are used in a several applications, including logic circuits, digital memory and Neuromorphic computing. Their resistance varies with the integral of current flowing through the device, or alternatively, the integral voltage across the device. The resistance is not affected when the electrical input is removed, essentially making them non-volatile in nature. The first stable prototype was recently developed in 2008 by HP Labs [9]. Since then, numerous memristor models have been proposed in literature [10], [11], [12].

2.1.1 The VTEAM Memristor Model

This work uses the Voltage Threshold Adaptive Memristor (VTEAM) model designed by Kvatinsky *et. al.* [12] to accurately model the memristor's behaviour in design and simulations. The model is given by the following equations,

$$\frac{dw(t)}{dt} = \begin{cases} k_{off} \cdot \left(\frac{v(t)}{v_{off}} - 1 \right)^{\alpha_{off}} \cdot f_{off}(w), & 0 < v_{off} < v, \\ 0, & v_{on} < v < v_{off}, \\ k_{on} \cdot \left(\frac{v(t)}{v_{on}} - 1 \right)^{\alpha_{on}} \cdot f_{on}(w), & 0 < v_{on} < v, \end{cases} \quad (1)$$

$$i(t) = G(w, v) \cdot v(t). \quad (2)$$

where w is an internal state variable, $v(t)$ is the voltage across the memristive device, $i(t)$ is the current passing through the memristive device, $G(w, v)$ is the device conductance, k_{off} , k_{on} , α_{off} , α_{on} are constants, v_{on} and v_{off} are threshold voltages.

2.1.2 HfO_x based Memristor

We use the multi-level linearized Pt/HfO_x/Hf/TiN RRAM device based on [13]. For this device, post fitting to the VTEAM model, the I-V relationship is given by,

$$i(t) = \left[R_{on} + \frac{R_{off} - R_{on}}{W_{off} - W_{on}} \cdot (w - w_{on}) \right]^{-1} \cdot v(t). \quad (3)$$

2.2 Artificial Synapse

Synapses are the building blocks of a neural network as they connect one neuron to the other. The strength of this connection is determined by the synaptic weight. A higher synaptic weight means strong dependency on the output of a neuron on its preceding neuron. When neuromorphic architecture is implemented on the conventional computing architecture, the synaptic weights are fetched from the memory unit to the processor unit where they are read and updated. The updated weights are stored back to the memory unit and the Von Neumann bottleneck remains a challenge [14].

This work implements artificial synapses using hybrid CMOS-memristor design from [2]. The resistance of memristors can be changed based on the history of applied electrical stimuli. This closely resembles to the biological synapses where the strength of connection increases or decreased based on the applied action potential [14]. The memristive synapse can not only store the weight but also naturally transmit information into post-neurons, overcoming the Von Neumann bottleneck. The design in [2] consists of a voltage-controlled memristor connected to the shared terminal of PMOS and NMOS, as shown in Fig. 2.1. The functionality of this design is described in the context of neuromorphic ADC and DAC in the following sections.

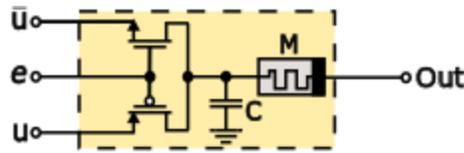


Figure 2.1: Schematic of the Artificial Synapse consisting of two transistors and a memristor [2].

2.3 Neuromorphic ADC

The deterministic four-bit neural network ADC in [2] converts an analog input voltage (V_{in}) to a digital output code ($D_3D_2D_1D_0$) according to the following iterative expressions,

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

where V_{ref} is the reference voltage equals to one full-scale voltage quantum (LSB), and $u(\cdot)$ is the signum neural activation function (neuron) having either zero or full-scale voltage output. The neural network shown in Fig. 2.2 implements (4) in hardware using reconfigurable synaptic weights ($W_{i,j}$ – conductance between a pre-synaptic neuron with index j and a post-synaptic neuron with index i) to address their non-deterministic distribution in real-time operation and post-silicon fabrication. As shown in Fig. 2.1, the synapses are realized using one NMOS, one PMOS and one memristor, with gates of the transistors connected to a common enable input e [15]. When $e = V_{DD}$ ($-V_{DD}$), the NMOS (PMOS) switches on and u ($-\bar{u}$) is passed to the output. When $e = 0$, both transistors are off and the output is zero. As shown in Fig. 2.3, the neurons comprise of an inverting op-amp for integration and a latched comparator for decision making.

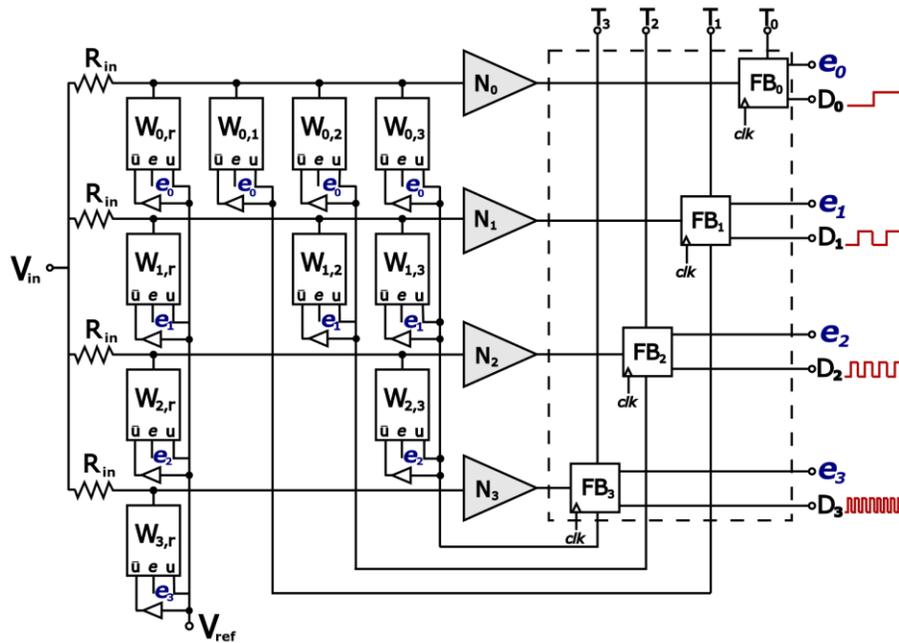


Figure 2.2: Neural network-based 4-bit ADC architecture trained online using SGD, including synapses $W_{i,j}$, neurons N_i , and feedback FB_i

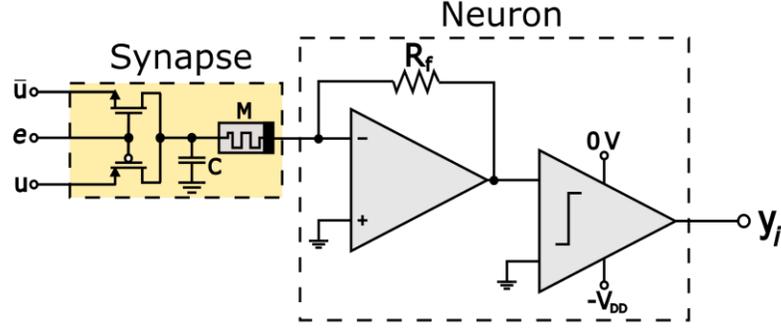


Figure 2.3: Schematic of a memristive synapse connected to an artificial neuron implemented as an inverting opAmp for integration and a comparator for decision making.

Synaptic weights are tuned to minimize the mean square error (MSE) by using the stochastic gradient descent (SGD) learning rule,

$$\Delta W_{ij(j>i)}^{(k)} = -\eta \left(T_i^{(k)} - D_i^{(k)} \right) T_j^{(k)}, \quad (5)$$

where η is the learning rate (a small positive constant), and in each iteration k , the output of the network $D_i^{(k)}$ is compared to the desired teaching label $T_i^{(k)}$ that corresponds to the input $V_{in}^{(k)}$. The training continues until the training error falls to $E_{threshold}$, a predefined constant that defines the learning accuracy.

2.4 Neuromorphic DAC

The neural network DAC in [5] converts the four-bit digital input code ($V_3V_2V_1V_0$) to an analog output (A) as,

$$A = \frac{1}{2^4} \sum_{i=0}^3 2^i V_i, \quad (6)$$

where binary weights (2^i) are implemented with reconfigurable synaptic weights W_i and having similar realization as in Fig. 2.1. As shown in Fig. 2.4, the four synapses collectively integrate the input through the neuron (op-amp) to produce the output. This output is compared to the analog teaching labels in the pulse width modulation (PWM)-based feedback circuit, which regulates the value of the weights in real-time according to the time-varying gradient descent learning rule,

$$\Delta W_i^{(k)} = -\eta(t) \left(V_{out}^{(k)} - t^{(k)} \right) D_i^{(k)}, \quad (7)$$

where $\eta(t)$ is the time-varying learning rate, and $t^{(k)}$ is the analog teaching label. The feedback is disconnected after the training is complete ($E < E_{threshold}$).

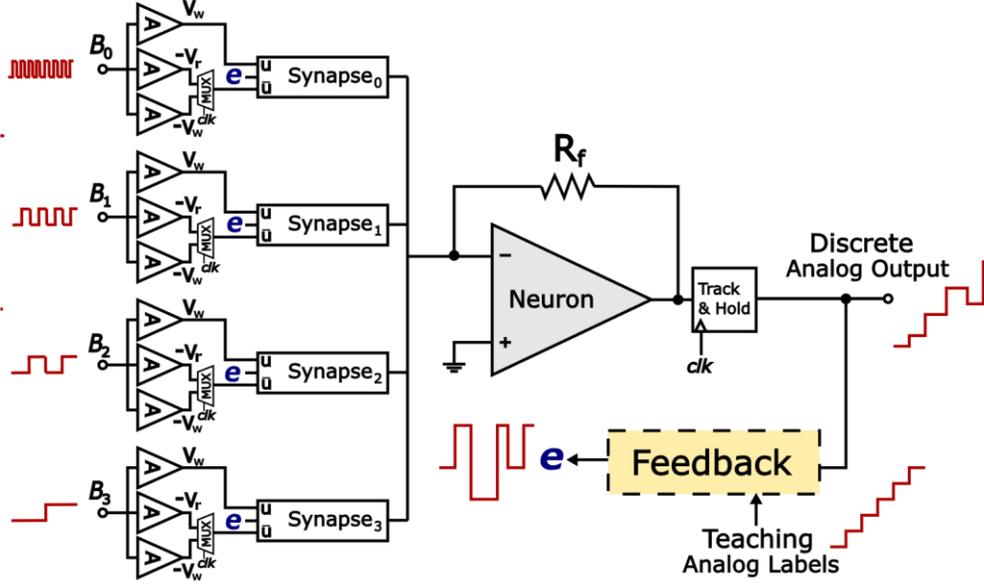


Figure 2.4: Neural network-based 4-bit binary-weighted DAC architecture, including synapse W_i , an artificial neuron implemented as an inverting opAmp, and a PWM-based feedback circuitry [5] for the time-varying SGD.

2.5 ADC Performance Metrics

The ADC is evaluated statistically for differential non-linearity (DNL) and integral non-linearity (INL). These are defined as,

$$DNL(j) = \frac{V_{j+1} - V_j}{LSB_{ideal}} \quad (7)$$

$$INL(j) = \sum_{i=1}^j DNL(i) \quad (8)$$

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

The *Signal to Noise and Distortion Ratio* (SNDR) is calculated from the FFT plot of ADC's output as,

$$SNDR = P_{signal} - P_{noise} \quad (9)$$

$$P_{signal} = P_{peak} + CPG + Scalloping_Loss \quad (10)$$

$$P_{noise} = P_{noise-floor} + P_G + CPG - ENBW \quad (11)$$

$$P_G = 10 \cdot \log_{10} \frac{N}{2} \quad (12)$$

where P_{peak} is the peak signal power from the FFT plot, $P_{noise-floor}$ is the average noise power, N is the total number of bits, and CPG , $Scalloping_Loss$, $ENBW$ are window-dependent parameters.

The *Effective Number of Bits* (ENOB) is calculated from the SNDR as,

$$ENOB = \frac{SNDR(dB) - 1.76}{6.02} \quad (13)$$

The figure-of-merit (FOM) relates the ADC's sampling frequency, f_s , power consumption during conversion, P , and effective number of bits, ENOB. A lower value of FOM signifies better overall performance. FOM is defined as,

$$FOM = \frac{P}{2^{ENOB} \cdot f_s} [J/conv] \quad (14)$$

Chapter 3

Scaling Challenges

Increasing the scale of the neural network ADC described in the previous chapter, above four bits, is challenging. Table I highlights the effect of scaling on design and performance parameters of the ADC. The number of synapses in the network increases quadratically. Consequently, the area and power consumption rise significantly. Moreover, there is an exponential rise in the aspect ratio of synaptic weights, which is practically limited by the high-to-low resistive states ratio (HRS/LRS), number of resistive levels, endurance of the memristor, and time and power consumption of the training phase [13] – ultimately limiting the practical achievable resolution to four-bits. Additionally, higher number of neurons require longer conversion-time which limits the maximal Nyquist sampling frequency.

TABLE I. SCALING CHALLENGES IN NEUROMORPHIC ADC

Parameter	4-bit	8-bit	N -bit
# Neurons, feedbacks	4	8	N
# Synapses	10	36	$N(N+1)/2$
Total area (μm^2)	4850	9740	$N(1.1N+1250)$
Conversion rate (GSPS)	1.66	0.74	$1/(N \cdot t_p + (N-1)/\text{BW})$
Power (μW)	100	650	$P_{int} + P_{act} + P_{synapse}$
FOM (fJ/conv)	8.25	7.5	$P/(2^{N-0.3} \cdot f_s)$
HRS/LRS (memristor)	2^4	2^8	$2^{N-1+\log_2(V_{dd}/V_{fs})}$
# Levels (memristor)	64	2048	$N \cdot 2^N$

Chapter 4

Memristive Pipelined Neuromorphic Analog-to-Digital Converter

4.1 Introduction to Pipelined ADCs

Pipeline is a technique where multiple instructions are overlapped during execution. It is divided into stages which are connected with one another to form a pipe like structure, as shown in Fig. 4.1. When one stage finishes execution, its output is sent to the following stage, allowing it to execute the next instruction. Thus, we can execute multiple instructions simultaneously. Pipeline increases the overall throughput on the expense of latency. With increase in number of stages, latency increases. Throughput is limited by the execution speed of the slowest stage.

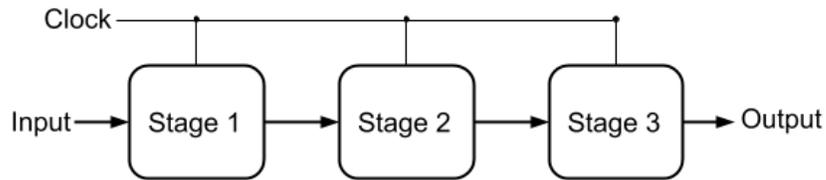


Figure 4.1: General concept of pipelining

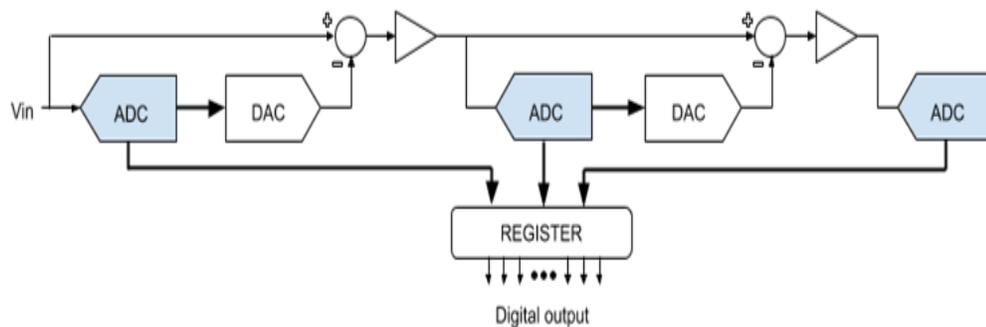


Figure 4.2: Schematic of a three-stage conventional pipelined ADC

Analog-to-digital conversion can be performed in a pipelined fashion. Fig. 4.2 shows the schematic of a conventional pipelined ADC. Quantization of the input analog signal is divided into stages, where each stage resolves a specific number of bits. After one stage performs conversion, the remaining information is present in the quantization error (analog input minus

digital output converted back to analog) which is amplified, tracked and held for the next stage. The digital output of each stage is time-aligned using digital logic.

4.2 Pipelined Neuromorphic ADC

In [6], we present the architecture, training mechanism, circuit topology, and evaluation results of our memristive pipelined neuromorphic ADC. In this chapter, I explain these findings in detail.

4.2.1 Neural Network Architecture

We propose using light-weight coarse-resolution neural network ADCs and DACs to build a fine-resolution pipelined network. An eight-bit two-stage pipelined ADC is shown in Fig. 4.3.

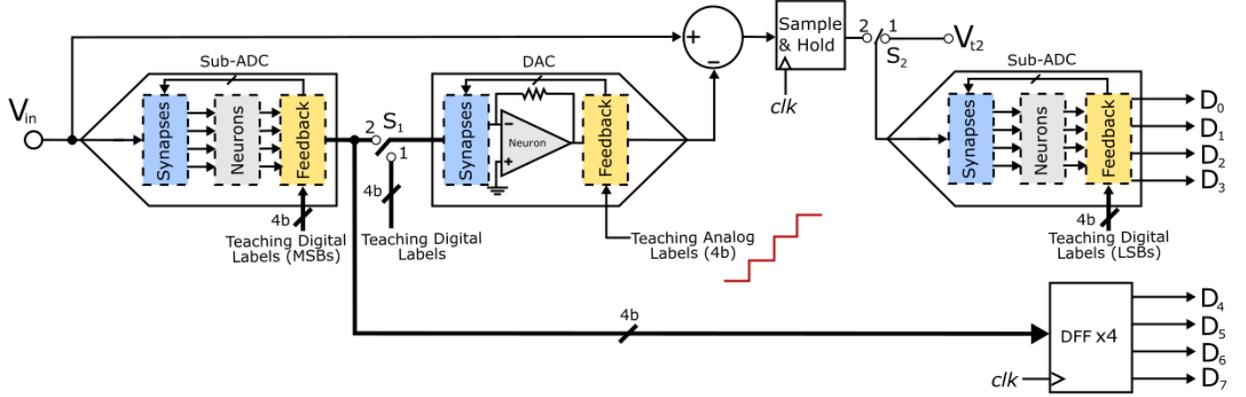


Figure 4.3: Proposed architecture of a two-stage pipelined ADC trained online using SGD.

In the first-stage sub-ADC, a synapse W_{ij} is present between a pre-synaptic neuron with index j and digital output D_j , and a post-synaptic neuron with index i , and digital output D_i . A neuron for each bit collectively integrates inputs from all synapses and produces an output by the signum neural activation function $u(\cdot)$. The sub-ADC coarsely quantizes (MSBs) the sampled input V_{in} to the digital code $D_7D_6D_5D_4$ (MSB to LSB) as,

$$\begin{cases} D_7 = u(V_{in} - 8V_{ref}), \\ D_6 = u(V_{in} - 4V_{ref} - W_{6,7}D_7), \\ D_5 = u(V_{in} - 2V_{ref} - W_{5,6}D_6 - W_{5,7}D_7), \\ D_4 = u(V_{in} - V_{ref} - W_{4,5}D_5 - W_{4,6}D_6 - W_{4,7}D_7). \end{cases} \quad (15)$$

The output of the sub-ADC is converted back to an analog signal A by the DAC as,

$$A = \frac{1}{2^4} \sum_{i=4}^7 W_i D_i, \quad (16)$$

where W_i are the synaptic weights. Next, this output is subtracted from the held input to produce a residue Q as,

$$Q = V_{in} - A. \quad (17)$$

This residue is sent to the next stage of the pipeline, where it is first sampled and held. The second stage sub-ADC is designed similar to that of the first stage, except that the resistive weights of the input are modified from $R_{in} = R_f$ (feedback resistance of neuron) to $R_f/16$. This is made in order to scale the input from $V_{FS}/16$ to the full-scale voltage V_{FS} . The LSBs of the digital output are obtained from this stage as

$$\begin{cases} D_3 = u(16Q - 8V_{ref}), \\ D_2 = u(16Q - 4V_{ref} - W_{2,3}D_3), \\ D_1 = u(16Q - 2V_{ref} - W_{1,2}D_2 - W_{1,3}D_3), \\ D_0 = u(16Q - V_{ref} - W_{0,1}D_1 - W_{0,2}D_2 - W_{0,3}D_3). \end{cases} \quad (18)$$

The sample-and-hold circuit enables concurrent operation of the two stages, achieving a high throughput rate, but introduces latency of two clock cycles. Thus D-flipflop registers are used to time-align the MSBs and the LSBs.

Trainable neural network ADC/DAC cores in this design have minimalistic design with mismatch self-calibration, noise tolerance, and power consumption optimization. This eliminates the need for an exclusive inter-stage gain unit and calibration mechanism, because the residue is amplified by the input resistive weight of the second sub-ADC. Although resistors are highly prone to manufacturing variations, they can be effectively used as the input weights because their mismatches will be calibrated for by other memristive weights in the second stage [5]. Furthermore, the training algorithm ensures that the quantization error remains within tolerable limits without using digital calibration techniques.

Fig. 4.4 show the circuit design of this two-stage pipelined ADC in 180 nm technology using SPICE (Cadence Virtuoso). Fig. 4.5 and Fig. 4.6 shows the schematic of sub-ADC and the DAC.

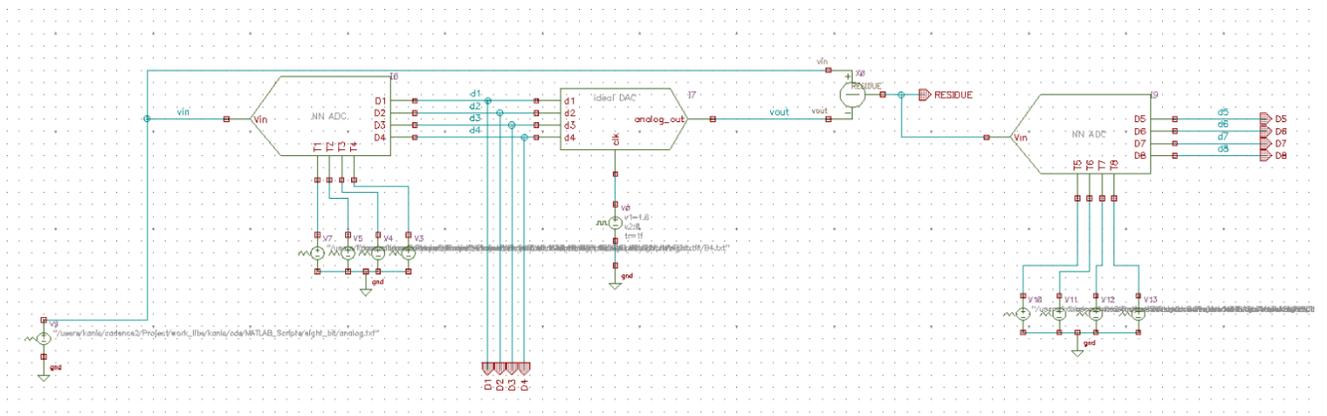


Figure 4.4: SPICE design of the memristive neural network pipelined ADC

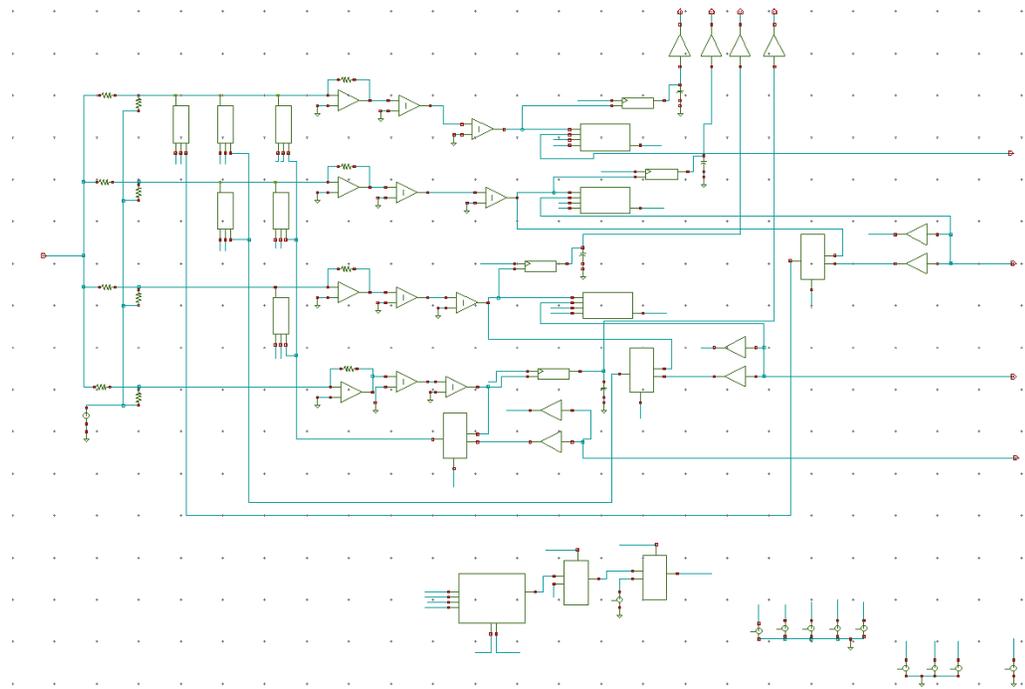


Figure 4.5: SPICE design of the four-bit neural network sub-ADC

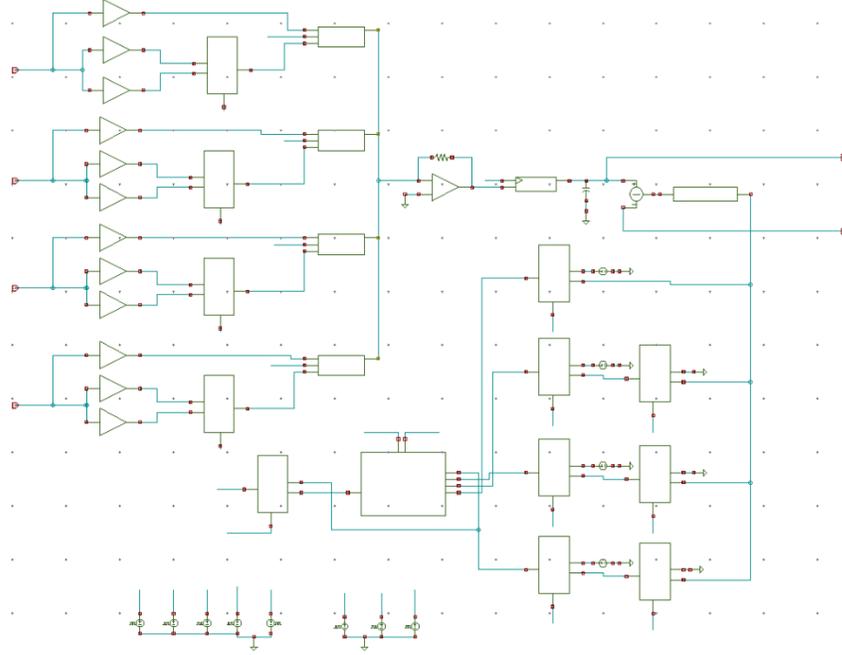


Figure 4.6: SPICE design of the four-bit neural network DAC

4.2.2 Training Framework

The aim of the training is to configure the network from a random initial state (random synaptic weights) to an accurate eight-bit ADC. It is achieved by minimizing the mean-square-error (MSE) of each sub-ADC and the DAC by using specific teaching labels for desired quantization. During the training phase, switches S_1 and S_2 are in position 1.

The DAC is supplied with four-bit digital teaching labels corresponding to an analog ramp input, as shown in Fig. 4.3. We use the binary-weighted time-varying gradient descent rule in (7) to minimize the MSE between the estimated and desired label. Learning parameters are listed in Table II. The DAC is connected to the sub-ADC by switch S_1 when the error falls below $E_{threshold}$.

The accuracy requirements of each stage decrease through the pipeline and the first stage should be accurate to the overall resolution [18]. Moreover, the two-stages operate on different inputs for different quantization. Thus, their teaching dataset must be different to execute the online SGD algorithm as,

$$\Delta W_{ij(j>i)}^{(k)} = -\eta_{ADC} (T_i^{(k)} - D_i^{(k)}) T_j^{(k)}, 0 \leq i, j \leq 3, \quad (19)$$

$$\Delta W_{ij(j>i)}^{(k)} = -\eta_{ADC} (T_i^{(k)} - D_i^{(k)}) T_j^{(k)}, 4 \leq i, j \leq 7. \quad (20)$$

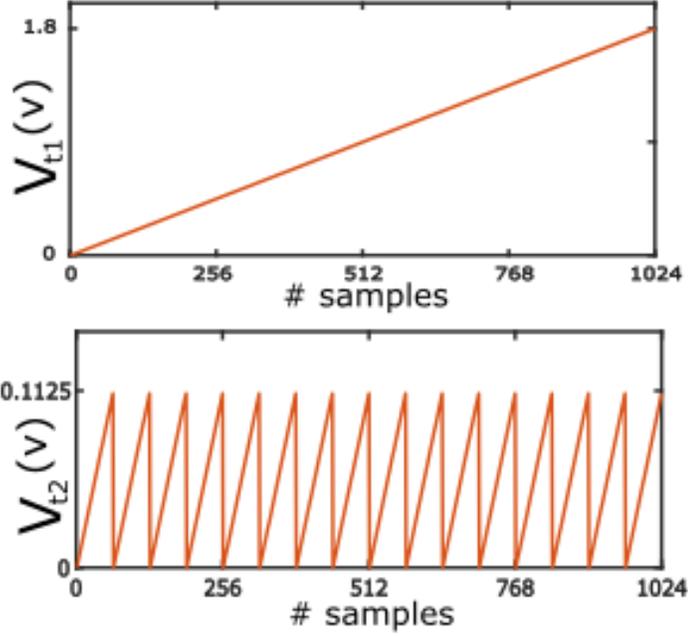


Figure 4.7: Training dataset of the sub-ADCs. V_{t1} and V_{t2} are supplied to the 1st and 2nd stage respectively.

TABLE II. PIPELINED ADC CIRCUIT PARAMETERS

Parameter	Value	Parameter	Value
Power supply		Feedback resistor	
V_{DD}	1.8 V	R_f	45 k Ω
NMOS		PMOS	
W/L	10	W/L	20
V_{TN}	0.56 V	V_{TP}	-0.57 V
Memristor			
$V_{on/off}$	-0.3 V, 0.4 V	$R_{on/off}$	2 k Ω , 100 k Ω
$K_{on/off}$	-4.8 $\mu\text{m/s}$, 2.8 $\mu\text{m/s}$	$\alpha_{on/off}$	3, 1
Reading voltage and time		Writing voltage and time	
V_r	-0.1125 V	V_w	± 0.5 V
T_r	5 μs	T_w	5 μs
Learning parameters		Sub-ADC/DAC parameters	
$\eta_{ADC/DAC}$	1, 1	f_s	0.1 MSPS
$E_{threshold}$ <small>ADC/DAC</small>	$4.5 \cdot 10^{-2}$, $9 \cdot 10^{-3}$	V_{FS}	V_{DD}

Interestingly, (19) and (20) can be implemented using different teaching inputs, as shown in Fig. 4.7. Furthermore, the two stages can be trained independently and in parallel as their teaching datasets are supplied separately.

For the training dataset, an analog ramp signal is sampled at $4 \cdot 2^8$ ($=1024$). Four adjacent samples are given the same digital labels, providing an eight-bit training dataset, shown as V_{t1} in Fig. 4.7. The more we train the ADCs with extra labels, the higher conversion accuracy we achieve. This is because of the nonlinear nature of the ADC task. The analog ramp input with the corresponding four MSBs is used to train the first stage ADC. A sawtooth version of this input (V_{t2} in Fig. 4.7) with the remaining LSBs is used for the training of second stage. The switch S_2 is turned to position 2, when the overall mean-square-error falls below $E_{threshold}$.

4.2.3 Performance Evaluation

Our proposed pipelined ADC is simulated and comprehensively evaluated in SPICE (Cadence Virtuoso) using a 180 nm CMOS process and memristors fitted by the VTEAM memristor model [12] to a Pt/HfO_x/Hf/TiN RRAM device [13]. The device has a HRS/LRS of 50. First, we evaluate the learning algorithm in terms of training error and learning time. Next, the circuit is statistically and dynamically evaluated, and finally, power consumption is analyzed. The circuit parameters are listed in Table II. To test the robustness of the design, we incorporate device non-idealities and noise, as listed in Table II in [5].

The basic deterministic functionality of the pipeline ADC is demonstrated during training by the online SGD algorithm. Fig. 4.8(a) shows the variation of the MSE of the first-stage DAC. After approximately 5,000 training samples (312 epochs), which equals 50 ms training time for a 0.1 MSPS conversion rate, the MSE error falls below $E_{threshold}$. Fig. 4.8(b) shows the total MSE of the two sub-ADCs. After approximately 40,000 training samples (39 epochs), which equals 400 ms training time, the total MSE falls below $E_{threshold}$. The analog output is converted through an ideal 8-bit DAC and probed at three different timestamps during training, as shown in Fig. 4.8(e). The output is identical to the input staircase after the training is completed.

Linearity plots (Fig. 4.8(c)), measured for 1.8 V ramp signal sampled by 18k points at 0.1 MSPS, show that dynamic nonlinearity (DNL) is within ± 0.20 LSB and integral nonlinearity (INL) is lower than ± 0.18 LSB. Fig. 4.8(d) shows the output spectrum at 0.1 MSPS sampling rate. The input is a 44 kHz 1.8 V_{pp} sine wave. The converter achieves 47.5 dB SNDR at the end of training. Next, we analyzed the power consumption of the network by considering neural integration power, neural activation power, and synapse power [2]. Remarkably, the total power consumption is optimized similar to [2] during training. The ADC consumes 272 μ W of power, averaged over a full-scale ramp with $4 \cdot 2^8$ samples.

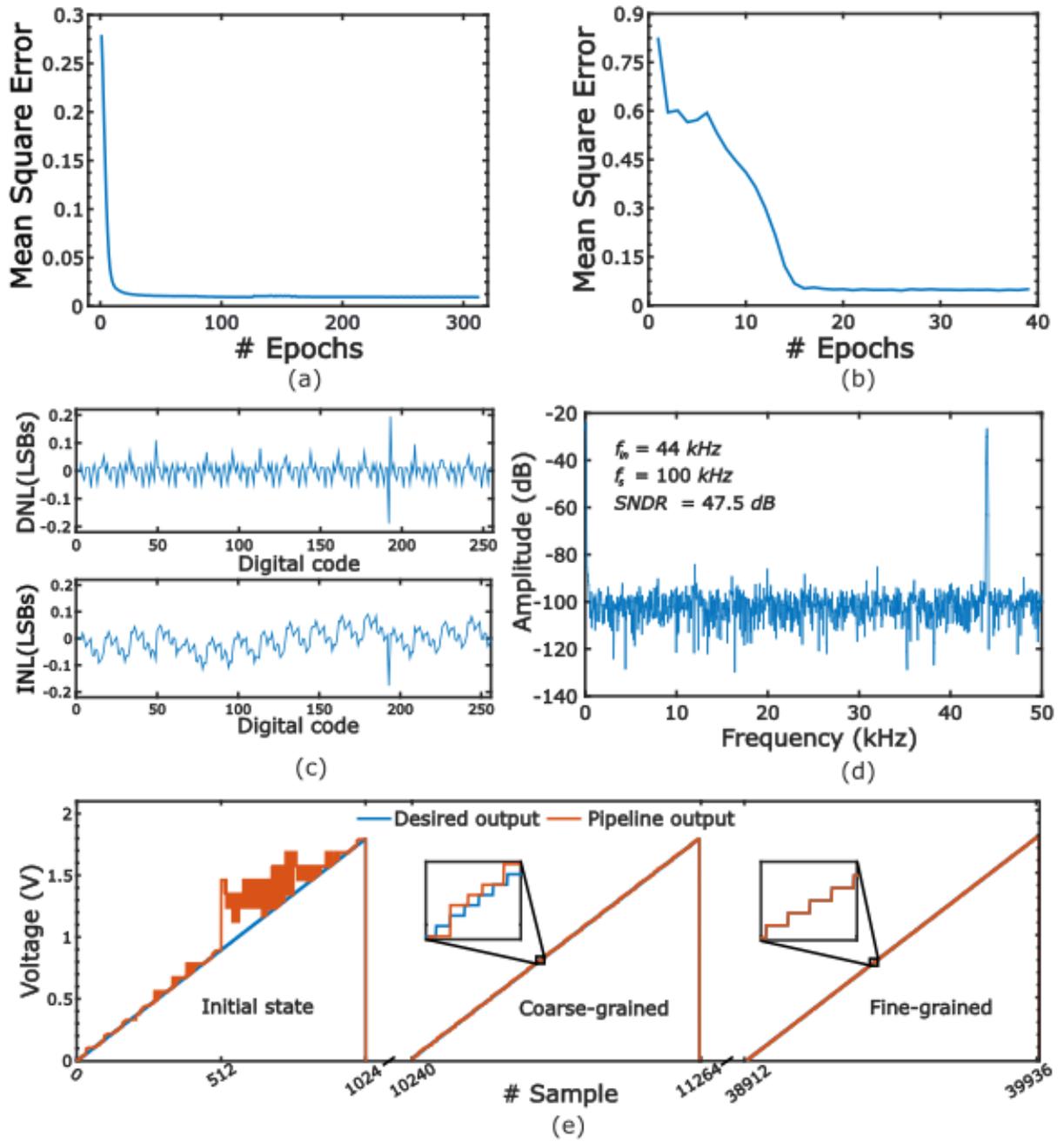


Figure 4.8: Pipeline ADC training evaluation. (a) Mean square error of the first-stage DAC minimization during its training. (b) Total mean square error of the two stages during training of sub-ADCs. (c) DNL and INL at the end of training. (d) 2048-point FFT for a 44 kHz sinusoidal input. (e) Comparison between the teaching dataset and the actual output of the ADC by connecting it to an ideal DAC, at three different timestamps during the training; an identical staircase (time-aligned for latency) is obtained when training is complete.

The pipelined ADC is tested for reconfigurability by changing the full-scale voltage from 1.8 V to 0.9 V and sampling frequency from 0.1 MS/s to 10 MS/s. The synaptic weights of the sub-ADCs and the DAC converges to new steady state to operate correctly under different specifications, as shown in Fig. 4.9. From the values of power consumption, maximum conversion speed and ENOB, the pipelined ADC achieves a FOM of 0.97 fJ/conv at the full-scale voltage.

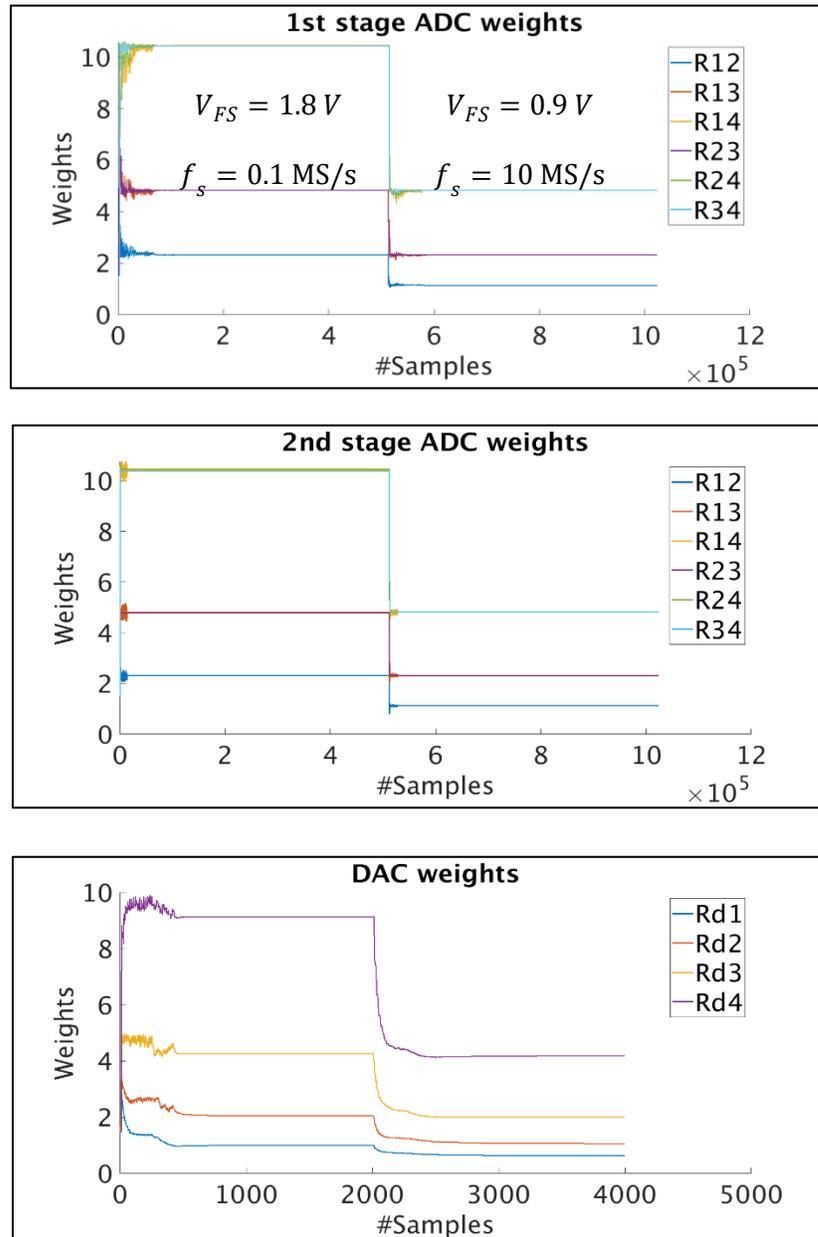


Figure 4.9: Variation of synaptic weights of the sub-ADC and the DAC during training showing self-reconfiguration when the full-scale voltage and sampling frequency are changed.

4.2.4 Performance Comparison

This 8-bit pipelined architecture is compared to the scaled version of neural network ADC [2]. As shown in Table III, the pipelined ADC consumes less power, achieves high conversion rate, and better FOM with lesser HRS/LRS device ratio.

TABLE III. PERFORMANCE COMPARISON

Parameter	NN ADC [2] ^a	This work
# Bits	8	8
# Synapse	36	24
Memristor HRS/LRS	2^8	2^4
Max conversion rate (GSPS)	0.74	1.66
Power (μ W)	650	272
FOM (fJ/conv)	7.5	0.97 ^b
Training time (ms)	1060	400

^a Based on scalability evaluation of the 8b neuromorphic ADC.

^b Extrapolated FOM at the maximum conversion rate.

4.2.5 Scalability Evaluation

To test the scalability of our architecture, we performed behavioral simulations in MATLAB. Our results for 12-bit design with ideal device parameters are summarized in Table IV.

TABLE IV. SCALABILITY EVALUATION

# Bits	12
# Synapses	38
# Samples per epoch	$1 \cdot 2^{12}$
Max DNL	0.61 LSB
Max INL	0.60 LSB
Training time (ms)	2000

Chapter 5

Logarithmic Neuromorphic Data Converters

5.1 Introduction

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. Fig. 5.1 shows the characteristics of linear and logarithmic quantization.

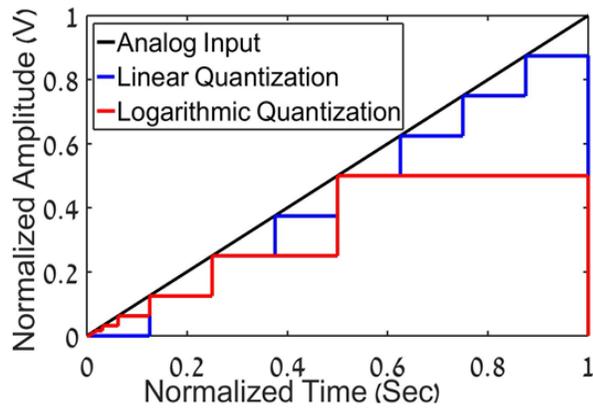


Figure 5.1: Characteristics of reconfigurable quantization: linear versus logarithmic.

5.1.1 Applications of Logarithmic Data Converters

For several biomedical applications, such as cochlear implants [16], hearing aids [17], neural recording and stimulation [18-22], 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 [16-20], reduction of noise and data bit-rate [21], and compensation for nonlinear sensor characteristics [23].

5.1.2 Logarithmic ADC

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

$$\sum_{i=0}^{N-1} D_i 2^i = \frac{2^N}{c} \log_B \left(\frac{V_{in}}{V_{FS}} B^c \right), \quad (21)$$

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 [22], 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. (21) 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.

For small input amplitudes, the LSB size is small and has a minimum value of,

$$LSB_{min} = V_{FS} B^{-c} \left(B^{\frac{c}{2^N}} - 1 \right), \quad (22)$$

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

$$LSB_{max} = V_{FS} \left(1 - B^{-\frac{c}{2^N}} \right), \quad (23)$$

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,

$$DR(dB) = 20 \log_{10} \left(\frac{V_{FS}}{LSB_{min}} \right) = 20 \log_{10} \left(\frac{B^c}{B^{\frac{c}{2^N}} - 1} \right). \quad (24)$$

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,

$$DNL(j) = \frac{V_{j+1} - V_j}{LSB_{ideal}}, \quad (25)$$

$$INL(j) = \sum_{i=1}^j DNL(i), \quad (26)$$

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

5.1.3 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

$$V_{out} = \frac{V_{FS}}{2^{N-1}} B^{\sum_{i=0}^{N-1} D_i 2^i}. \quad (27)$$

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} .

5.2 Trainable Neural Network Logarithmic ADC

In [7], we present the architecture, training mechanism, circuit topology, and evaluation results of our logarithmic neural network ADC/DAC. In this chapter, I explain these findings in detail.

The work utilizes 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,

$$\begin{cases} D_2 = u(V_{in} - 2^4 V_{ref}) \\ D_1 = u(V_{in} - 2^2 V_{ref} \bar{D}_2 - 2^6 D_2) \\ D_0 = u(V_{in} - 2 V_{ref} \bar{D}_1 \bar{D}_2 - 2^3 D_1 \bar{D}_2 - 2^5 \bar{D}_1 D_2 - 2^7 D_1 D_2) \end{cases}, \quad (28)$$

where V_{in} is the analog input and $D_2 D_1 D_0$ is the corresponding digital form ($i=2$ is the MSB), while each \bar{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 (28). 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 $(2^{N-1} + 2)$.

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,

$$\Delta W_{ij(j>i)}^{(k)} = -\eta(T_i^{(k)} - D_i^{(k)})T_j^{(k)}, \quad (29)$$

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 $T^{(k)}$. The training phase continues until the error is below $E_{threshold}$.

5.3 Trainable Neural Network Logarithmic DAC

We formulate the logarithmic DAC equations in an ANN-like manner as follows, using three bits as an example,

$$V_{out} = 2^0 \overline{D_0} \overline{D_1} \overline{D_2} + 2^1 D_0 \overline{D_1} \overline{D_2} + 2^2 \overline{D_0} D_1 \overline{D_2} + 2^3 D_0 D_1 \overline{D_2} + 2^4 \overline{D_0} \overline{D_1} D_2 + 2^5 D_0 \overline{D_1} D_2 + 2^6 \overline{D_0} D_1 D_2 + 2^7 D_0 D_1 D_2. \quad (30)$$

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, with a time-varying learning rate and a teaching analog signal $t^{(k)}$,

$$\Delta W_i^{(k)} = -\eta(t)(V_{out}^{(k)} - t^{(k)})D_i^{(k)}. \quad (31)$$

5.4 Circuit Design of Neural Network Logarithmic ADC/DAC

The neural network ADC/DAC architectures and their building blocks, including neurons, synapses, and training feedbacks, are illustrated in Fig. 5.2. The synapse and Neuron circuit

designs are explained in Chapter 1. The memristive crossbar (2T1R) inherently implements Ohm's and Kirchoff'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 [12] to a Pt/HfO_x/Hf/TiN RRAM device [13]. 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. 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 [22]. Table V lists the circuit parameters.

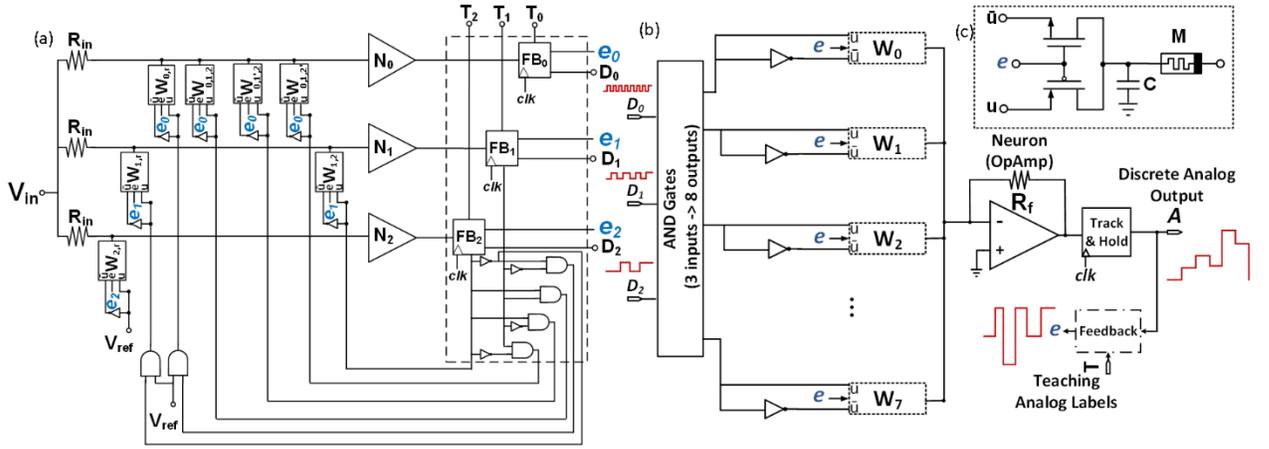


Figure 5.2: (a) Architecture of the proposed 3-bit logarithmic neural network ADC [7]; (b) Architecture of proposed 3-bit logarithmic neural network DAC [7]; (c) Schematic of artificial synapse [2]

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 (29) and (31) and execute basic subtraction and multiplication operations. We used the same training circuits from [2], [5]. 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 [5]. After the training is complete ($E \leq E_{threshold}$), the feedback is disconnected from the conversion path.

TABLE V. LOG ADC/DAC CIRCUIT PARAMETERS

Parameter	Value	Parameter	Value
Power Supply		Feedback resistor	
V_{DD}	1.8 V	R_f	400 k Ω
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 Ω , 1.5 M Ω
$K_{on/off}$	-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	± 0.5 V
T_r	5 μ s	T_w	5 μ s
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}

5.5 Performance 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 V. Furthermore, circuit variations and noise sources are quantified and validated, as listed in [5].

The basic deterministic functionality of the three-bit logarithmic ADC/DAC is demonstrated during training by the online SGD algorithm. Figure 5.3(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. 5.3(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. Fig. 5.4 shows the INL and DNL plots. After training, the ADC is almost fully calibrated, monotonic, and accurate: $INL \approx 0.26$ LSB, and $DNL \approx 0.62$ 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 [2], 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 [5]. The proposed networks can also be trained to perform linear ADC/DAC using linearly quantized teaching data-sets. Table VI lists the full performance metrics and comparison with the linear ADC/DAC.

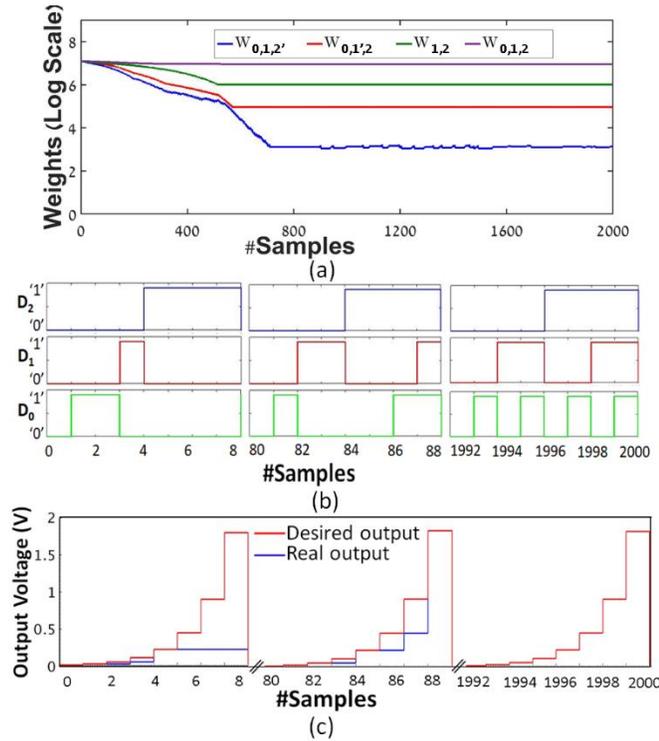


Figure 5.3: Logarithmic ADC training evaluation. (a) Synapse reconfiguration (in log scale) during training for $N=3$, $V_{FS}=1.8V$ and $f_s=100KSPS$. The weight is equal to the ratio between R_f and the corresponding memristor; thus, it has no units. (b) The actual digital outputs D_i (logical value) 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.

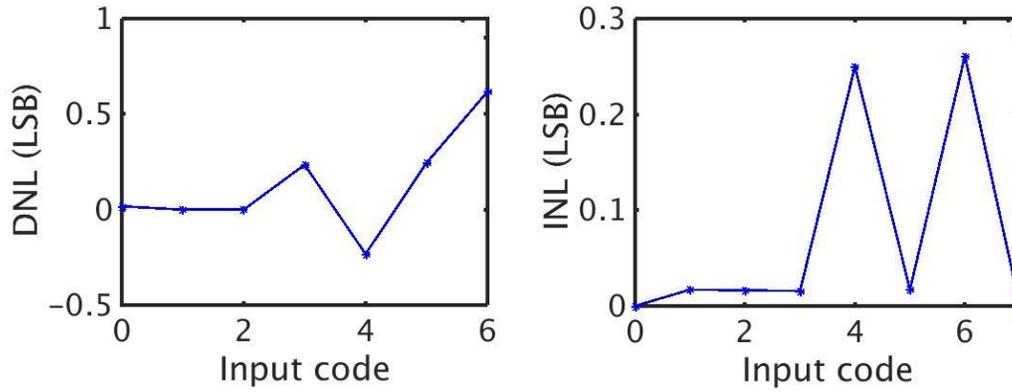


Figure 5.4: DNL and INL plots for the logarithmic ADC

TABLE VI: LOG ADC/DAC PERFORMANCE EVALUATION

Metric	Logarithmic ADC	Linear ADC [2]
<i>N</i>	3 bits	4 bits
<i>INL</i>	0.26 LSB	0.4 LSB
<i>DNL</i>	0.62 LSB	0.5 LSB
<i>DR</i>	42.114 dB	24.08 dB
<i>SNDR</i>	17.1 dB	24.034 dB
<i>ENOB</i>	2.55	3.7
<i>P</i>	45.18 μ W	100 μ W
<i>FOM</i>	77.19 pJ/conv	0.136 nJ/conv
<i>Training time</i>	20 ms	40 ms
Metric	Logarithmic DAC	Linear DAC [5]
<i>N</i>	3 bits	4 bits
<i>INL</i>	0.163 LSB	0.12 LSB
<i>DNL</i>	0.122 LSB	0.11 LSB
<i>Training time</i>	80 ms	30 ms

Chapter 6

Conclusion

This report presents a novel pipelined neural network ADC architecture. This large-scale design was based on coarse-resolution neuromorphic ADC and DAC, modularly cascaded in a high-throughput pipeline and precisely trained online using SGD algorithm for multiple full-scale voltages, and sampling frequencies. The learning algorithm successfully tuned the neural network in non-ideal test conditions and configured the network as an accurate, fast, and low-power ADC. The hybrid CMOS–memristor design with 1.8 V full-scale voltage achieved 0.97 fJ/conv FOM at the maximum conversion rate.

The report also presents 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 presented 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. The simulations achieved a 77.19 pJ/conv FOM, exceeding the performance of a linear ADC. I believe that this work constitutes a milestone with promising results for the realization of large-scale neuromorphic data converters for real-time adaptive applications.

References

- [1]. Y. Chiu, B. Nikolic and P. R. Gray, "Scaling of Analog-to-Digital Converters into Ultra-Deep-Dubmicron CMOS," *Proceedings of the IEEE 2005 Custom Integrated Circuits Conference*, pp. 375-382, 2005.
- [2]. L. Danial, N. Wainstein, S. Kraus and S. Kvatinsky, "Breaking Through the Speed-Power-Accuracy Tradeoff in ADCs Using a Memristive Neuromorphic Architecture," *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 2, No. 5, pp. 396-409, Oct. 2018.
- [3]. E. O. Neftci, "Data and Power Efficient Intelligence with Neuromorphic Learning Machines," *iScience*, Vol. 5, pp. 52–68, 2018.
- [4]. A. Tankimanova and A. P. James, "Neural Network-Based Analog-to-Digital Converters," *Memristor and Memristive Neural Networks*, Apr. 2018.
- [5]. L. Danial, N. Wainstein, S. Kraus and S. Kvatinsky, "DIDACTIC: A Data-Intelligent Digital-to-Analog Converter with a Trainable Integrated Circuit using Memristors," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 8, No. 1, pp. 146-158, March 2018.
- [6]. L. Danial, K. Sharma, and S. Kvatinsky, "A Pipelined Memristive Neural Network Analog-to-Digital Converter" IEEE International Symposium on Circuits and Systems (ISCAS), 2020 (under review).
- [7]. L. Danial, K. Sharma, S. Dwivedi, and S. Kvatinsky, "Logarithmic Neural Network Data Converters using Memristors for Biomedical Applications", *Proceedings of the IEEE Biomedical Circuits and Systems (BioCAS)*, October 2019.
- [8]. L. Chua, "Memristor-The missing circuit element," *IEEE Transactions on Circuit Theory*, Vol. 18, No. 5, pp. 507-519, September 1971.
- [9]. D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The Missing Memristor Found," *Nature*, Vol. 453, No. 7191, pp. 80-83, 2008.
- [10]. D. B. Strukov, J. J. Yang, M. D. Pickett, and J. L. Borghetti, "Switching dynamics in titanium dioxide memristive devices", *Journal Applied Physics*, page 1-6, 2009.
- [11]. J. J. Yang, M. D. Pickett, X. Li, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevice", *Nature Nanotechnology*, page 429-433, 2008.

- [12]. S. Kvatinsky, M. Ramadan, E. G. Friedman and A. Kolodny, "VTEAM: A General Model for Voltage-Controlled Memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 62, No. 8, pp. 786-790, Aug. 2015.
- [13]. J. Sandrini, B. Attarimashalkoubeh, E. Shahrabi, I. Krawczuk, and Y. Leblebici, "Effect of Metal Buffer Layer and Thermal Annealing on HfOx-based ReRAMs," *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, pp. 1-5, Nov. 2016.
- [14]. Sanghyeon Choi, Seonggil Ham and Gunuk Wang (March 29th 2019). Memristor Synapses for Neuromorphic Computing [Online First], IntechOpen, DOI: 10.5772/intechopen.85301. Available from: <https://www.intechopen.com/online-first/memristor-synapses-for-neuromorphic-computing>.
- [15]. D. Soudry, D. Di Castro, A. Gal, A. Kolodny and S. Kvatinsky, "Memristor-Based Multilayer Neural Networks With Online Gradient Descent Training," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 26, No. 10, pp. 2408-2421, Oct. 2015.
- [16]. 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.
- [17]. J. Mahattanakul, "Logarithmic Data Converter Suitable for Hearing Aid Applications," *IET*, Vol. 41, No. 7, pp. 394 – 396, Mar. 2005.
- [18]. 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.
- [19]. 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.
- [20]. 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.
- [21]. 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.
- [22]. Y. Sundarasaradula *et al.*, "A 6-bit, Two-Step, Successive Approximation Logarithmic ADC for Biomedical Applications," *ICECS*, pp. 25-28, 2016.
- [23]. 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.