University of Windsor Scholarship at UWindsor

Electronic Theses and Dissertations

2018

Mixed-Signal Neural Network Implementation with Programmable Neuron

Bahar Youssefi

Follow this and additional works at: https://scholar.uwindsor.ca/etd

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Mixed-Signal Neural Network Implementation with Programmable Neuron

by

Bahar Youssefi

A Dissertation

Submitted to the Faculty of Graduate Studies through the Department of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at the University of Windsor

Windsor, Ontario, Canada 2018

© 2018 Bahar Youssefi

All Rights Reserved. No Part of this document may be reproduced, stored or otherwise retained in a retreival system or transmitted in any form, on any medium by any means without prior written permission of the author.

Mixed-Signal Neural Network Implementation with Programmable Neuron

by

Bahar Youssefi

APPROVED BY:

F. Mohammadi, External Examiner Ryerson University

A. Jaekel School of Computer Science

K. Tepe Department of Electrical and Computer Engineering

H. Wu Department of Electrical and Computer Engineering

J. Wu, Co-Advisor Department of Electrical and Computer Engineering

M. Mirhassani, Advisor Department of Electrical and Computer Engineering

November 16, 2017

Declaration of Co-authorship / Previous Publication

I. Co-authorship

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

Chapter 2 of the thesis was co-authored with A.J. Leigh, as an outstanding scholar under under the supervision of Dr. M. Mirhassani. A.J. Leigh contributed to the layout design and editing of the manuscript. Chapter 5 of this thesis was co-authored with S. Abdollahi, as a research associate under the supervision of Dr. M. Mirhassani. S. Abdollahi provided feedback on refinement of ideas.

In all cases, the key ideas, primary contributions, designs, schematics, block diagrams, data analysis, interpretation, and writing were performed by the author.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes 5 original papers that have been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication Title	Publication status
Chapter 2	Bahar Youssefi, Alexander J. Leigh, Mitra Mirhassani, and Jonathan Wu, "Tunable Neuron PWL Approximation Based on the Minimum Operator," IEEE Transactions on Circuits and Systems II: Express Briefs	Minor revisions requested
Chapter 3	Bahar Youssefi, Mitra Mirhassani, and Jonathan Wu, "Efficient Mixed-Signal Synapse Multipliers for Multi-Layer Feed-Forward Neural Networks," <i>IEEE</i> <i>International Midwest Symposium on Circuits and Systems</i> , pp. 814-817, Oct. 2016.	Published
Chapter 4	Bahar Youssefi, Mitra Mirhassani, and Jonathan Wu, "Hardware Realization of Mixed-Signal Neural Networks with Modular Synapse-Neuron Arrays," <i>IEEE</i> <i>International Symposium on Circuits and Systems</i> , 2018	Submitted
Chapter 5	Bahar Youssefi, Siamak Abdollahi, Mitra Mirhassani, and Jonathan Wu, "Nonlinear Dynamics of Single Sigmoid Neural Network," <i>IEEE Transactions</i> on Neural Networks and Learning Systems	Submitted
Chapter 6	Bahar Youssefi, Mitra Mirhassani, and Jonathan Wu, "A Current-Mode Mixed-Signal Approach to Realize the Distributed-Arithmetic-Based FIR Filters," <i>IEEE Access Journal</i> , The institution of engineering and technology (IET)	Under revision for resubmission

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

This thesis introduces implementation of mixed-signal building blocks of an artificial neural network; namely the neuron and the synaptic multiplier. This thesis, also, investigates the nonlinear dynamic behavior of a single artificial neuron and presents a Distributed Arithmetic (DA)-based Finite Impulse Response (FIR) filter. All the introduced structures are designed and custom laid out.

A novel VLSI implementation of a reconfigurable neuron based on choosing the minimum operator utilizing the winner-take-all circuit is proposed. The neuron estimates the Sigmoid-shape activation function using the piece-wise linear approximation method and achieves the adaptability by taking advantage of the body effect of PMOS transistors. The structure covers a variety of activation functions such as rectified linear, hard-limit, and different precision sigmoid functions which aims to improve the generalization ability in neural networks.

An area and power-efficient synaptic multiplier is proposed which works based on the combination of the digital gates and weighted current mirrors. A 4-3-2 neural network containing the modular synapse-neuron building blocks is successfully tested for pattern recognition. The proposed artificial neural network addresses the area-efficiency considering the inevitable growth in the size of the current networks.

Moreover, the nonlinear behavior of a single sigmoidal neuron is investigated to discuss the oscillatory behavior of a single neuron and its possible applications in the future generation of oscillators.

The proposed FIR filter is designed aiming to address the efficient VLSI implementation which works based on the distributed arithmetic. There is trade-off between the computation efficiency of the DA-based processing and area-efficiency of multiply-and accumulate (MAC)-based ones. The proposed FIR filter reduces the required area for a DAbased filter by employing mixed-signal approach. An 8-bit 16-tap FIR filter is designed and successfully tested for a BPF and LPF at 10MHz and 48KHz respectively.

Acknowledgments

I wish to express my most sincere gratitude to my supervisor Dr. Mitra Mirhassani who has been more than an advisor to me. During the past years, she was my mentor and the source of motivation. I would like to thank my co-advisor Dr. Jonathan Wu for his constant support throughout the course of this work.

In addition to my advisors, I would like to thank my committee members, Dr. Kemal Tepe, Dr. Huapeng Wu and, Dr. Arunita Jaekel, and Dr. Farah Mohammadi for their constructive comments and feedback.

I would also like to thank my friends and colleagues Parham H. Namin, Babak Zamanlooy, Iman Taha, and Alexander Leigh for their support and all my colleagues in the ECE department, ASM and RCIM labs.

Finally, my deepest gratitude goes to my husband, Siamak Abdollahi, and my dearest parents for their unconditional love, support, and encouragement.

Contents

Declaration of Co-authorship / Previous Publication	iv
Abstract	vii
Acknowledgments	ix
List of Figures	xiii
List of Tables	xviii
List of Abbreviations	xix
 1 Introduction 1.1 Outline of the Dissertation and List of the Contributions	1 5
References	7
2 Reconfigurable Neuron PWL Approximation Based on the Minimum Opera- Tor	10

	2.1	Introduction	10
	2.2	PWL approximation for a non-Monotonic function	12
	2.3	The proposed reconfigurable structure of the neuron	16
	2.4	The Reconfigurability and the simulation results	19
	2.5	Conclusion	23
Re	eferen	ces	25
3	Mixe	d-Signal Synapse Multipliers for Feed-Forward Neural Networks	27
	3.1	Introduction	27
	3.2	Neural Network Configurations	28
	3.3	Building block's components	30
		3.3.1 Neuron	30
		3.3.2 Mixed-Signal Multiplier	31
	3.4	Conclusion	36
Re	eferen	ces	38
4	Hard	ware Realization Of Mixed-Signal Neural Networks	40
	4.1	Introduction	40
	4.2	Self-adjustable distributed Neuron	42
	4.3	distributed neural network	44
		4.3.1 Synaptic Multiplier	45
	4.4	Pattern recognition	47
	4.5	Conclusion	49

References 5 Dynamic Behavior Of A Single Sigmoid

5 Dynamic Behavior Of A Single Sigmoidal Neuron: Stable To Period Doubling	54
5.1 Introduction	54
5.2 Background and Theory	55
5.3 Stability Analysis of the single neuron structure	57
5.4 Simulation Results	64
5.5 Conclusion	65
References	66
6 Low-Power Mixed-Signal Implementation of the DA-based FIR Filter	68
6.1 Introduction	68
6.2 Distributed Arithmetic	71
6.3 Proposed Current-Mode Distributed Arithmetic Structure	72
6.4 Proposed Filter Implementation	76
6.4.1 DAC	79
6.4.2 Current-Mode Delay Cell	84
6.5 Results Discussion	90
6.6 Conclusion	93
References	94
7 Conclusions and Future Works	96
7.1 Summary of Contributions	96
7.2 Suggested Future Work	98
VITA AUCTORIS	99

List of Figures

2.1	The sigmoid function of $\frac{19}{1+e^{-0.1x}}$ and the corresponding 5- pieces PWL	
	approximation.	13
2.2	The reconfigurable neuron schematic.	14
2.3	(a) The output voltages V_A (dashed line) and V_B (solid line) for V_G of	
	700mV, $730mV$, and $800mV$. (b) ideal sigmoid function, PWL approx-	
	imation achieved from least fit method, and the simulation result of the	
	neuron output function (solid line)	15
2.4	The deviation error for the 5-piece PWL approximation of $\frac{K}{1+e^{-0.1x}}$ for	
	K = 19, 10, and 5.	19
2.5	The PWL neuron outputs that show the reconfigurable sigmoid functions	
	for $V_G = 710mV$ and the linear functions for $V_G = 250mV$. Different	
	shades of the sigmoid and the linear functions are shown for V_S of 2V,	
	2.3V, 2.4V, and 2.5V	20
2.6	2-bit voltage DAC corner analysis result for different conditions ff , ss ,	
	fs, and sf .	20
2.7	Post-layout simulation results for different conditions of ff , ss , fs , and sf	
	for the temperatures of -55 C , 27 C , and 125 C .	22

The corner and temperature post-layout analysis for different conditions of	
ff, ss , sf , and fs at temperatures of 27C, -55C, and 125C showing the	
deviation from the activation function at 27 <i>C</i>	23
System-level configuration of the proposed mixed-signal neural network.	29
General configuration of the mixed-signal neural network[7]	30
Non-linear neuron activation function which approximates the sigmoid func-	
tion	31
Multiplying the two less significant and the two most significant bits of Y	
with the analog value of the input X \hdots	31
The proposed modular mixed-signal multiplier to be used in distributed	
feed-forward neural network	32
MA/MB output result for Y=1111for corner analysis fast-fast (ff), slow-	
slow (SS), slow-fast (sf) and fast-slow (fs) to show the process variation	
effect	34
Multiplication results for Y=0010, 0111, 1010, 1011, and 1100. Ideal and	
simulation results are shown with dashed and solid lines respectively	34
The resistive-type neuron circuit modified to a robust current-mode struc-	
ture. The synaptic multiplier's output current is applied to the neuron as I_{in}	
via terminal T_1 . The output current is shaped by a self-adjustable sigmoid	
function	42
The variation of the neuron's activation functions for the input ranges of	
$(-60\mu A, 60\mu A)$ shown by the solid line, $(-100\mu A, 100\mu A)$ shown by dot-	
ted line, and $(-200\mu A, 200\mu A)$ shown by dashed line	43
The 1000 runs Monte Carlo simulation results of the current-mode neuron's	
activation function.	43
	The corner and temperature post-layout analysis for different conditions of ff , ss , sf , and fs at temperatures of $27C$, $-55C$, and $125C$ showing the deviation from the activation function at $27C$

4.4	The system level configuration of a 4-3-2 distributed neural network. W_{ij}	
	and C_{ij} are the digital synaptic weights corresponding to the second and	
	third layers respectively. I_1 to I_4 are the input currents representing the	
	input patterns.	45
4.5	The modular signed multiplying DAC that performs as the synapse [12].	
	T_1 is connected to the terminal with the same name in Fig. 4.1 to build the	
	synapse-neuron module.	46
4.6	The corner analysis simulation results of tt (Typical NMOS Typical PMOS),	
	ff (Fast NMOS Fast PMOS), fs (Fast NMOS Slow PMOS), sf (Slow NMOS	
	Fast PMOS), and ss (Slow NMOS Slow PMOS) that show the process vari-	
	ation effect on the multiplication performance for the input current of $1\mu A$	
	that is multiplies to a digital weight that varies from -11111 to 11111	47
4.7	The structure of the current comparators that are connected to the O_1 and	
	O_2 terminals of Fig. 4.4.	49
4.8	The input templates that are used to test the functionality of the neuron	50
4.9	Simulation results of the 4-3-2 distributed neural network to prove the pat-	
	tern recognition capability.	50
4.10	The 4-3-2 mixed-signal network layout.	51
4.11	Simulation results of the 4-3-2 distributed neural network to show the sen-	
	sitivity regarding three critical paths	51
5.1	A single sigmoidal neuron in a feedback configuration.	57
5.2	The stationary solution for the arbitrary values of $\mu = 2, \beta = 3$, and	
	$x_0 = 0.55.\ldots$	58
5.3	The bifurcation map of the structure shown in Fig. 5.2 when $x_0 = 0.55$ and	
	$y_0 = 0.001728.$	58
5.4	Stationary solution of y_0 for various β and μ values	59

5.5	Stability map achieved from the system eigenvalues which show the two	
	possible behaviours, stable and period doubling, for the system	59
5.6	Time domain behavior of the system for two different arbitrary sets of μ an	
	β . (a) shows the oscillatory behavior for $\mu = 6$ and $\beta = 3$. (b) shows the	
	stable behavior for $\mu = 0.3$ and $\beta = 0.15$.	63
6.1	The proposed DA architecture for a 16-tap 8-bit mixed-signal FIR filter. x_{ij}	
	is the j^{th} bit of the i^{th} input X_i . I_{Ci} is the i^{th} filter coefficient and $y[n]$ is	
	the final output current. (a) The compelete current-mode DA architecture	
	(b) The structure at the $high$ state of the first $N-1$ clock cycles. (c) The	
	structure at the low state of the first $N-1\ {\rm clock}\ {\rm cycles}\ ({\rm d})$ The structure at	
	the N^{th} clock cycle when the operation is done	70
6.2	Multiplying stage of the proposed mixed-signal filter	73
6.3	DA-based delay/division stage of the proposed current-mode mixed-signal	
	filter	74
6.4	Overall conceptual operating waveforms of the proposed filter. The nota-	
	tions show the signal level at the specific clock cycle	75
6.5	The 5-bit DAC structure [13]	80
6.6	The 5-bit DAC output current of I_{Ci} (solid line), the exact error calculated	
	from $Error(\mu A) = I_{Ci} - I_{ideal}$ shown by dashed line, the error percentage	
	calculated from $\frac{100 \cdot (I_{Ci} - I_{ideal})}{I_{ideal}}$ shown by dash-dotted line	81
6.7	The family plot of the DAC output current vs. the analog equivalent of the	
	digital input achieved from 500 runs of Monte Carlo simulations	81
6.8	The input and output currents of two cascaded delay cells and a current	
	divider for four random input currents of $4.98\mu A$, $20.02\mu A$, $60\mu A$, and	
	$99.98\mu A$	82

6.9	Corner analysis of tt (Typical NMOS Typical PMOS), ff (Fast NMOS Fast	
	PMOS), fs (Fast NMOS Slow PMOS), ss (Slow NMOS Slow PMOS), and	
	sf (Slow NMOS Fast PMOS) of the error percentage occurs in the feedback	
	branchs output current.	83
6.10	The 1000 runs Monte Carlo simulation family plot of the second delay cell	
	output current for four random input currents of 100μ , $58\mu A$, $24\mu A$, and	
	$6\mu A.\ldots$	86
6.11	The frequency and phase responses of the DA-based BPF and LPF	87
6.12	The layout of the 8-bit 16-tap mixed-signal filter based on DA	92

List of Tables

2.1	The neuron schematic transistors dimensions.	18
2.2	The intersection points of (x_1, y_1) and (x_2, y_2) , the slopes, and the y-intercepts	
	of the 5-piece PWL approximation	18
3.1	Simulation and ideal multiplication result for Y=1111 at different input	
	levels and the measured error	35
3.2	Simulation and ideal multiplication result for Y=1111 at different input	
	levels and the measured error	36
4.1	Sizes of the transistors of the multiplier.	47
4.2	The comparison table of the proposed 4-3-2 distributed NN and other sim-	
	ilar structures.	49
6.1	DAC transistors dimentions.	83
6.2	Filter's Coefficients.	91
6.3	Comparison of the proposed filter with recent published filters	92

List of Abbreviations

ADC	Analog to Digital Converter.
ANN	Artificial Neural Network.
BPF	Band-Pass Filter.
CMOS	Complementary Metal Oxide Semiconductor.
DA	Distributed Arithmetic.
ff	Fast NMOS Fast PMOS.
FIR	Finite Impulse Response.
fs	Fast NMOS Slow PMOS.
LPF	Low-Pass Filter.
LTA	Loser-Take-All.
LTI	Linear Time-Invariant.
MAC	Multiply-Accumulate.
MDAC	Multiplying DAC.
NMOS	Negative Metal Oxide Semiconductor.
NN	Neural Network.
PMOS	Positive Metal Oxide Semiconductor.
PWL	Piece-wise Linear.
ROC	Region Of Convergence.
sf	Slow NMOS Fast PMOS.
SS	Slow NMOS Slow PMOS.
tt	Typical NMOS Typical PMOS.
VLSI	very large-scale integration.
WTA	Winner-Take-All.

Chapter 1

Introduction

In this chapter, a brief overview of the mixed-signal approach towards different signal processing building blocks especially the neural network implementation is presented. This chapter shortly investigates diverse types of artificial neurons and the importance of utilizing an adjustable neuron. Also, the possible nonlinear behavior of the neural network is explored briefly.

The mixed-signal approach integrates both analog and digital elements in a single silicon chip [1, 2]. There is an impressive attention drew to the mixed-signal ICs because of the two trends in the design industry [3]. First, the transistor dimensions have been scaled down to deep submicron levels to allow millions of transistor and complex systems integrated into a solo die. Second, multifaceted, complex systems need to put the digital signal processors and the analog circuitries together on a single chip by using the digital to analog converters (DAC) and analog to digital converters (ADC).

These two trends in the IC industry, the upsurge in the number of the transistors packable in a single die and the growing extensiveness of electronic systems, make the mixedsignal circuit design a demandable approach in IC market [3]. Another attractiveness of the mixed-signal approach is the ability to partially skip the disadvantages of both analog and digital implementations when blending the advantages. Addition, subtraction, and division by a constant number are examples of operations that can be done effortlessly in analog domain [4].

Multipliers which are one of the essential building blocks of the filters and neural networks can be built by using multiplying DACs [5, 6, 7] which perform the multiplication in the analog domain by digital controls. Some of the mixed-signal circuits can work with a fewer number of DAC or ADC or avoid using any, by utilizing MDACs as multipliers. The fewer number of DACs and ADCs means the inevitable reduction in the area and power consumption.

Another multiplication method which can benefit from mixed-signal implantation is Distributed Arithmetic (DA) [8]. Distributed arithmetic is a bit-serial computational method that performs the inner product in a different fashion than multiply-accumulate (MAC) operations [9]. Crosier et al. [10] introduced the DA concept for the first time; then the method was used for digital implementation of FIR filters [11]. In the DA approach, the clock cycles that are needed to compute the inner product is fixed and depends on the resolution of the input data. This approach has been utilized in image coding [12], filter implementation [8, 13], vector quantization [14], and discrete cosine transform [15]. Compared to MAC operations, DA is more efficient regarding computations and mechanizations; the advantage is more visible when the system needs to deal with a large length input vector [8, 9]. It should be noted that the previous structures of the DA replaced the multipliers by large memories, shift registers, and adders which increased the area and power consumption. The proposed mixed-signal implementation would be beneficial by eliminating the adders, subtractors, and dividers. The details of the proposed architecture are presented in Chapter 6.

Artificial neural network (ANN) is another computational method that can benefit from

the mixed-signal implementation. ANNs have the ability of being trained to provide solution to different types of problems for which analytical solutions do not exist or hard to be calculated [16] such as pattern recognition [17], memories [18, 19], nonlinear signal prediction, time-series prediction [20, 25], and action recognition [26].

ANNs have been drawing attention due to their generalization ability which leads to the better prediction performance [20], therefore, a solution that could improve their generalization ability is critical.

Considering the capability of the neural network in solving unknown problems [16], they can be used in some specific applications such as wearable sensors that are used to control the patients conditions continuously [21] or as wireless sensor network (WSN) that employ a network of several sensors to monitor environmental conditions [22].

These real-life applications can give us a view of the design characteristic that should be considered in the neural network implantations. The first aspect that should be considered in the neural network realization is that the design should be able to perform the parallel computation to follow the network principles [23]. The area and power consumption are the issues to be paid attention to in portable and battery-powered devices such as wearable sensors and WSNs. Accuracy is a less of concern criteria in neural network implementations since inaccurate elements performance can be modified during the training of the network [23, 24]. In general, the analog implementation performs the parallel processing while provides more area and power-efficiency compared to the digital realizations whereas showing less accuracy. The mixed-signal approach can benefit from the parallel calculation, area and power-efficient characteristic of the analog design while it shows higher accuracy in comparison.

Two basic composing blocks of an ANN are neuron and synapse. In the synapse, the synaptic weight is multiplied to an input; then the result passes the neuron which shapes the synapse output due to its activation function. In analog implementations [27, 28], synaptic weights, processing, and neuron's activation function are implemented by analog circuits

usually providing a higher efficiency compared to digital implementation [6, 29].

Analog implementations can realize the highly parallel nature of the biological neural networks, however, they are not as accurate as the digital realizations. The inaccuracy of analog implementations can be compensated by increasing the number of neurons [6]. Mixed-signal implementation can improve the accuracy compared to analog implementation while still benefiting from analog circuits advantages. A modular multiplying DAC can adequately perform as a synapse module in which the digital synaptic weights are stored in shift registers [29].

Another challenge of the ANNs implementation is the realization of the neuron activation function which can be sigmoid, hyperbolic tangent, hard-limit, Poslin, and linear. Area, power consumption, and accuracy are the criteria that are considered in the implantation. Reconfigurability of the neuron's activation function is another specification that give the neuron the ability to change shape post-fabrication. A programmable (reconfigurable) neuron is primarily can be used in multiresolution learning paradigm which has been proposed as a method that improves the ANNs generalization feature significantly [20, 25]. The multiresolution method works based on adjusting the activation function corresponding to the resolution they need, that means to start with the coarse tuning activation functions and increase the resolution as going further [20, 25]. In this fashion, an adaptable analog implementation of a neuron activation function would be beneficial for analog and mixed-signal applications that are aiming to achieve improvement in generalization property.

To get a deeper knowledge of how the neural network performs, knowing the behavior of a single neuron is an immense help. In this thesis, the nonlinear dynamic behavior of a single sigmoidal neuron with a feedback synaptic weight is investigated and the possible applications are proposed.

The analog and mixed-signal research lab at the University of Windsor has been focused on the pattern recognition and in a special case movement recognition using neural networks. This thesis is motivated by the basic blocks that are used in pattern recognition and challenges brought up in mixed-signal implementations.

1.1 Outline of the Dissertation and List of the Contributions

- In Chapter 2, A very large-scale integration (VLSI) prototype of a reconfigurable neuron is proposed and realized for the first time. The programmable neuron can be used in analog and mixed-signal networks. The activation function of the neuron can be accustomed off-chip or on-chip by a 2-bit voltage digital to analog converter (DAC) to provide the hard-limit, linear, and variable slope sigmoid functions. Since the proposed neuron is able to provide adjustable precision, it would be invaluable for neural network applications such as signal prediction which use multi-resolution learning paradigm to increase the efficiency of the system by improving the general-ization ability of the network.
- In Chapter 3, an area and power-efficient synaptic multiplier is realized in TSMC CMOS 0.18µm technology. The mixed-signal MDAC is highly modular making it suitable to be used to multiply digital synaptic weights and the analog inputs. The structure reduced the dimensions of the required transistors and decrease the need for weighted current mirrors compared to conventional MDACs.
- In Chapter 4, a 4-3-2 mixed-signal neural network is employed for pattern recognition application and a series of patterns are tested successfully. The network building blocks are the proposed neuron introduced in Chapter 2 and the proposed synaptic multiplier presented in Chapter 3.
- In Chapter 5, it is shown that a single sigmoid neuron with a feedback synaptic weight shows the oscillation behavior which is the simplest system which can realize

a neural oscillator for the first time. The frequency of oscillation only depends on the propagation delay of the system which is promising to reduce the dependency of VLSI implementations on the process and fabrication variations.

- In Chapter 6, the distributed arithmetic principles is used to implement a mixedsignal FIR table without need to us a lookup table. DACS with the current-mode outputs are utilized to do the multiplication between digital inputs and the analog coefficients. The current-mode multiplication eliminates the required adders and dividers, consequently, reduces the required area and power consumption. Two 16-tap 8-bit FIR filters (a BPF and an LPF) are realized by using the proposed architecture.
- Lastly, Chapter 7 highlights the contributions of the research and introduced the possible future works.

References

- [1] M. Burns, G. W. Gordon, *An introduction to mixed-signal IC test and measurement*, New York: Oxford University Press, vol. 2001, Mar. 2001.
- [2] S. Davidson "An Introduction to Mixed-Signal IC Test Measurement [Book Review]," IEEE Design Test, vol. 30, no. 3, pp.94–96, Jun. 2013.
- [3] B. Kaminska, K. Arabi, I. Bell, P. Goteti, J. K. Huertas, B. Kim, A. Rueda, M. Soma, "Analog and mixed-signal benchmark circuits-first release," *in Proceedings International Test Conference 1997, IEEE*, pp.183–190, Nov. 1997.
- [4] B. Razavi, *Design of analog CMOS integrated circuits*, Boston, MA:McGraw-Hill, 2001.
- [5] E. I. El-Masry, H. K. Yang, M. A. Yakout "Implementations of artificial neural networks using current-mode pulse width modulation technique," *IEEE transactions on neural networks*, vol. 8, no. 3, pp.532–548, May. 1997.
- [6] H. Djahanshahi, A robust hybrid VLSI neural network architecture for a smart optical sensor, University of Windsor, 1999.
- [7] H. Djahanshahi, M. Ahmadi, G. A. Jullien, and W. C. Miller, "Design and VLSI implementation of a unified synapse-neuron architecture," *in Proceedings of the 6th Great Lakes Symposium on VLSI*, pp.228-233, Mar. 1996.
- [8] E. Zalevli, W. Huang, P. E. Hasler, and D. E. Anderson, "A Reconfigurable Mixed-Signal VLSI Implementation of Distributed Arithmetic Used for Finite-Impulse Response Filtering," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 3, pp.510–521, Mar. 2008.
- [9] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Magazine*, vol. 6, no. 3, pp.4–9, Jul. 1989.
- [10] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo "Digital Filter for PCM Encoded Signals," U.S. Patent, 3 777 130, Dec. 1973.

- [11] A. Peled and B. Liu "A new hardware realization of digital filters," *IEEE Transactions* on Acoustics, Speech, and Signal Processing, vol. 22, no. 6, pp.456-462, Jun. 1974.
- [12] S. N. Merchant and B. V. Rao"Distributed arithmetic architecture for image coding," *Fourth IEEE Region 10 International Conference*, vol. 55,no. 3, pp.7477, Nov. 1989.
- [13] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson"LMS adaptive filters using distributed arithmetic for high throughput," *IEEE Transactions on Circuits* and Systems I: Regular Papers, vol. 52, no. 7, pp.1327–1337, Jul. 2005.
- [14] H. Q. Cao and W. Li"VLSI implementation of vector quantization using distributed arithmetic," *IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World*, vol. 2, pp. 668-671, May 1996.
- [15] M. T. Sun, T. C. Chen, and A. M. Gotlieb, "VLSI implementation of a 16x16 discrete cosine transform," *IEEE transactions on circuits and systems*, vol. 36, no. 6, pp.610-617, Jun. 1989.
- [16] J. Heikkonen, J. Lampinen, and A. M. Gotlieb, "Building industrial applications with neural networks," *InProceedings of the European symposium on intelligent techniques*, pp.3-4, 1999.
- [17] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp.85–117, Jan. 2015.
- [18] J. A. Anderson, "A simple neural network generating an interactive memory," *Mathematical biosciences*, vol. 14, no.3-4, pp. 197–220, Aug. 1972.
- [19] T. Kohonen, "VLSI implementation of a 16x16 discrete cosine transform," *IEEE transactions on computers*, vol. 100, no. 4, pp.353–359, Apr. 1972.
- [20] M. T. Sun, T. C. Chen, and A. M. Gotlieb, "Improving signal prediction performance of neural networks through multiresolution learning approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 2, pp.341–352, Apr. 2006.
- [21] S. Rhee, B. H. Yang, and H. H. Asada, "Artifact-resistant power-efficient design of finger-ring plethysmographic sensors," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 7, pp.795–805, Jul. 2001.
- [22] I. F. Kakkar, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp.393-422, Mar. 2002.
- [23] V. Kakkar, "Comparative study on analog and digital neural networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 7, pp.14-21, Jul. 2009.

- [24] J. Van der Spiegel, P. Mueller, D. Blackman, P. Chance, C. Donham, R. Etienne-Cummings, and P. Kinget, "An Analog Neural Computer with Modular Architecture for Real-Time Dynamic Computations," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 1, pp.82-92, Jan. 1992.
- [25] Y. Liang and E. W. Page, "Multiresolution learning paradigm and signal prediction," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp.2858-2864, Nov. 1997.
- [26] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1110–1118, 2015.
- [27] D. Anguita, A. Boni, "Neural network learning for analog VLSI implementations of support vector machines: a survey," *Neurocomputing*, vol. 55, no. 1, pp.265-283, Sep. 2003.
- [28] J. Cosp, J. Madrenas, and D. Fernndez, "Design and basic blocks of a neuromorphic VLSI analogue vision system," *Neurocomputing*, vol. 69, no. 16, pp.1962–1970, Oct. 2006.
- [29] B. Zamanlooy, M. Mirhassani, "Mixed-signal VLSI neural network based on Continuous Valued Number System," *IEEE transactions on circuits and systems*, vol. 221, pp.15–23, Jan. 2017.

Chapter 2

Reconfigurable Neuron PWL Approximation Based on the Minimum Operator

2.1 Introduction

Hardware implementation of neural networks relies on efficient implementation of neurons [1, 2, 3] and their synapses [4]. A challenge in the implementation and application of the neural networks is to enhance their generalization capability which leads to the better prediction performance. Multiresolution learning paradigm is a relatively new approach which improves the neural network generalization ability significantly [5]. The method works based on tuning the neurons activation function during the training process; consequently, a neuron with an adaptable transfer is required in this method. Although the multiresolution learning process has been proposed before, the hardware implementation

faces challenges in creating adjustable neurons.

In this paper, a universal and programmable analog neuron is proposed that can change the shape of its transfer function on demand without the need to redesign the neuron circuit. There are several very large-scale integration (VLSI) implementations and applications for various activation functions such as sigmoid function [1], hard-limit [2] and linear activation functions [3]. However, to the best of the authors knowledge, a reconfigurable analog structure that can provide different transfer functions on demand without redesigning the neuron circuit is not proposed. The neuron can generate the various sigmoid functions, as well as hard-limit, and linear activation functions.

The proposed neuron can be distributed in the network, where each node is composed of several sub-neurons. This type of neurons has been shown to improve the network performance [6]. Moreover, the distributed sub-neurons scaled over the input range, therefore prevented the neuron to become a band-limiting or a low gain linear function. However, the sub-neuron activity was not the result of training, rather the range of input values caused the scaling effect. On the other hand, the proposed neuron can generate the desired functions based on the multiresolution training algorithm and under the control of the designer.

The desired function of the proposed neuron is generated by choosing the minimum operator based on the piecewise linear (PWL) approximation method. The PWL approximation of an exponential function utilizing a Winner-Take-All (WTA) circuit was presented for the first time in [7]. However, the method cannot provide a solution for estimating non-monotonic functions.

The architecture proposed in this chapter works based on choosing the minimum basis function (operator) to provide the PWL approximation of the looked-for neuron activation function. Therefore, it can be suitably called a Loser-Take-All (LTA) structure. The design method is suitable for the PWL approximation of different non-monotonic functions; in this chapter, the proposed method is used to estimate a tunable sigmoid function. The proposed neuron can change shape and is tunable to form different *sigmoid* functions as well as the

linear ones.

2.2 PWL approximation for a non-Monotonic function

The *sigmoid* function, defined by $\frac{K}{1+e^{-ax}}$, is the base of the PWL approximation in this paper. It should be mentioned that the proposed method can be used to estimate other non-monotonic functions as well.

The general approach to PWL approximation is to divide the input range into several subintervals and to define a linear basis function that fits the curve in each subinterval. A WTA-based approach suggested in [7] uses the fact that the exponential function increases monotonically to approximate e^{ax} function in a positive interval of $[0, X_N]$. However, most neuron activation functions, including the *hard-limit*, *hyperbola*, *poslin*, and *sigmoid* functions are defined in the interval of $[-X_N, X_N]$ and are not monotonic [8].

In the proposed approach, the first step is to find the best estimation which fits the curve with the minimum error. The least square fit is a curve fitting modeling method that is widely used for a different variety of functions and is a routine approach for optimized approximation [9].

Fig. 2.1 shows a case study of the sigmoid function of $\frac{19}{1+e^{-0.1x}}$ and its 5-pieces PWL approximation achieved by applying the least square curve-fitting method. In each subinterval, the minimum operator is the one which fits the original function. Reasonably, a Loser-Take-All (LTA) circuit that chooses the minimum correspondence function in a subinterval among all the basis functions can successfully approximate the circuit. In the positive interval, the output can be described as follows:

$$\frac{K}{e^{-ax}+1} \approx y_{apx} = min[m_1x + b_1, m_2x + b_2, ..., m_nx + b_n]$$
(2.1)

in which m_n and b_n are the slope and the y-intercept of each basis function respectively.

Each basis function can be built by a current mirror with the dimension ratio of m_n set



Figure 2.1: The sigmoid function of $\frac{19}{1+e^{-0.1x}}$ and the corresponding 5- pieces PWL approximation.

with respect to the dimensions of the input transistor, W_{in}/L_{in} , as follows:

$$m_n = \frac{W_n/L_n}{W_{in}/L_{in}} = \frac{K}{(x_n - x_{n-1})(1 + e^{-ax_n})(1 + e^{-ax_{n-1}})}$$
(2.2)

The y-intercept is a DC current offset that is added to the output of the corresponding current mirror.

It should be noticed that this function is symmetric with respect to the point x_0 . By means of this symmetry, the function in the negative interval, $(-65\mu A, 0)$, is generated from the function in positive interval, $(0, 65\mu A)$, by subtracting f(x) from a constant current, such that $(g(x) = I_C - f(x))$. The constant current I_C is equal to the higher horizontal asymptote of the *sigmoid* function as shown in Fig. 2.1. f(x) and g(x) define the *sigmoid* function in the positive and the negative intervals respectively. As shown in Fig. 2.1 the minimum operator between the basis functions (1), (2), and (3) shows the PWL estimation in the positive interval. The dimension ratios of the corresponding current mirrors which



Figure 2.2: The reconfigurable neuron schematic.

generate these basis functions are as follows:

$$m_{1} = K \frac{e^{-ax_{0}} - e^{-ax_{1}}}{(x_{1} - x_{0})(1 + e^{-ax_{0}})(1 + e^{-ax_{1}})}$$
$$= K \frac{1 - e^{-a\Delta x}}{(2\Delta x)(1 + e^{-a\Delta x})}$$
(2.3)

$$m_{2} = K \frac{e^{-ax_{1}} - e^{-ax_{2}}}{(x_{2} - x_{1})(1 + e^{-ax_{1}})(1 + e^{-ax_{2}})}$$
$$= K \frac{e^{-a\Delta x}(1 - e^{-ap\Delta x})}{(p\Delta x)(1 + e^{-a\Delta x})(1 + e^{-a(p+1)\Delta x})}$$
(2.4)

$$m_3 = 0$$
 (2.5)

in which $x_1 - x_0 = \Delta x$ and $x_2 - x_1 = p \Delta x$.



Figure 2.3: (a) The output voltages V_A (dashed line) and V_B (solid line) for V_G of 700mV, 730mV, and 800mV. (b) ideal sigmoid function, PWL approximation achieved from least fit method, and the simulation result of the neuron output function (solid line).

The DC offset of the current mirrors are achieved as follows:

$$b_{1} = \frac{K}{2}$$

$$b_{2} = \frac{K}{p} (p + e^{-a\Delta x} ((p+1)e^{-ap\Delta x} - 1))$$

$$b_{3} = K$$
(2.6)

To simplify the structure, the second basis function is generated from the combination of the first and the third ones such that $m_2 = \alpha(m_1 + m_3)$ and $b_2 = \beta(b_1 + b_3)$. By substituting these equations in to the equations 5.4 to 5.7, α and β are achieved as follows:

$$\alpha = \frac{2e^{-a\Delta x}(1 - e^{-ap\Delta x})}{p(1 - e^{-a\Delta x})(1 + e^{-a(1+p)\Delta x})}$$

$$\beta = \frac{3p}{2(p + e^{-a\Delta x}((p+1)e^{-ap\Delta x} - 1))}$$
(2.7)

In this section, the structure and the design method of the current-mode neuron activation function of $\frac{19}{1+e^{-0.1x}}$ in the input range of $[-65\mu A, 65\mu A]$ is represented. Achieving the second basis function from the averaging of the first and the third ones not only leads to a simpler structure but keeps the activation function smooth. That means any mismatch or process variation that may arise in the first or the third current mirror block cells during fabrication would similarly affect the result of the average and thus would eliminate discontinuity in the PWL approximation result.

2.3 The proposed reconfigurable structure of the neuron

Fig. 2.2 shows the structure of the proposed reconfigurable neuron with the sigmoid activation function. A two-section WTA (shown in the dashed box) is employed to compare the mirrored currents corresponding to the first basis function, $I_1 = m_1 I_{in} + b_1$, and the third one, $I_3 = b_3$. Due to the nature of WTA, the output voltage of V_A goes high only if $I_1 > I_3$.
In the case that $I_1 < I_3$ the voltage of V_A goes low while V_B goes high. The voltages V_A and V_B are used to control the switches which let the minimum operator pass through the output transistor, M_{13} . NOT gates are used as the push-pull amplifiers to generate V_A and V_B that are able to reach the absolute value of 0 and V_{dd} .

In the conventional WTA circuits, M_3 and M_6 work in the saturation region and the voltages V_A or V_B can be high or low because even a small difference between their currents reduces the drain voltage of the transistor with the lower current, in a chain loop this reduction goes further until V_{o2} becomes almost 0 and V_{o1} becomes 1 [10]. When M_3 and M_6 work in the triode region, an overlap region is generated at which V_A or V_B can be low at the same time due to the higher dependency of the current to the drain voltage. The bias voltage of V_G determines the range of input currents at which the overlap occurs. Two cascaded NOT gates are used to push V_A/V_B to zero or pull them to V_{dd} . In this way, the dependency of these voltages to the effective threshold voltages of NOT gates is neglectable. Fig. 2.3(*a*) shows the overlap range variation for different values of V_G . As shown in this figure, the overlapping gap increases when the lower V_G is applied to the circuit.

The signals, $\overline{V_A}$, $\overline{V_B}$, and $V_A \cdot V_B$ are used to control the switches which allow one of the basis functions to pass through to the output. At the positive input range, \overline{sign} is 1 which closes the S_1 ; in this case, the output current would be equal to I_o as shown in Fig. 2.2. At the negative input range, S_2 , S_3 , and S_4 which are controlled by Sign are closed, and the output current would be equal to $I_{out} = b_3 - I_o$.

Fig. 2.3(*b*) illustrates the simulation result for V_G of 710mV, the ideal sigmoid function, and the PWL approximation from Fig. 2.1. The voltage V_G is chosen in a way that it provides the most similar subintervals to the sigmoid approximation shown in Fig. 2.1. The transistors dimensions and ratios are selected considering formulas (5.4) to (5.8) and are summarized in Table 2.1.

The accuracy of this method can be shown by using the standard deviation which is defined as $E = \frac{K}{1+e^{-ax}} - y_{apx}$. In the least square method, the subintervals are not equal

and are chosen to provide the best fit to the curve. If the first subinterval $(x_1 - x_0)$ is chosen to be the reference and is considered to be equal to Δx , the subinterval $(x_n - x_{n-1})$ can be assumed to be $p\Delta x$ while the $x_{n-1} - x_0 = q\Delta x$. In this case, the standard deviation, E, for the subinterval of (x_n, x_{n-1}) is achieved as follows:

$$\frac{E}{K} = \frac{1}{1 + e^{-ax}} - \frac{xe^{-aqx}(1 - e^{-ap\Delta x}) + p\Delta x +}{p\Delta x(1 + e^{-aq\Delta x})(1 + e^{-a(p+q)\Delta x})} \times \frac{q\Delta x^{-aqx}(e^{-ap\Delta x} - 1) + p\Delta xe^{-a(q+p)\Delta x}}{\Delta x(1 + e^{-aq\Delta x})(1 + e^{-a(p+q)\Delta x})}$$
(2.8)

As shown in the above equation, the standard deviation depends on the subinterval, a, and K. The maximum standard deviation occurs at the two ends of the subinterval, x_n or/and x_{n-1} . Consequently, it is simplified to:

$$\frac{E_{max}}{K} = \frac{2q\Delta x e^{-aq\Delta x}}{p\Delta x (1 + e^e - aq\Delta x)(1 + e^{-a(p+q)\Delta x})}$$
(2.9)

Table 2.1: The neuron schematic transistors dimensions.

$W_{in}/L_{in} = 7.5/1$	$W_2/L_2 = W_3/L_3 = 4/1$
$W_1/L_1 = W_8/L_8 = 3.5/1$	$W_4/L_4 = W_5/L_5 = 4/1$
$W_9/L_9 = 1/1$	$W_6/L_6 = W_7/L_7 = 0.25/0.18$

Table 2.2: The intersection points of (x_1, y_1) and (x_2, y_2) , the slopes, and the y-intercepts of the 5-piece PWL approximation.

Basis function	intersection points (x,y)	Slope	y-intercept
1	(-12.4,3.698), (12.4,15.4)	0.47	9.555
2	(12.4,15.4), (37.2,19.01)	0.145	13.6
3	(37.2,19), (37.2,18.99)	0	19
4	(-12.4,3.698), (-37.2,0.094)	0.145	5.5
5	(-37.2,0.094), (-62,0.112)	0	0.0112

The intersection points $(x_0 \text{ to } x_n)$ of the PWL approximation of the *sigmoid* function of $\frac{K}{1+e^{-ax}}$ depends on the number of basis functions and *a*. Accordingly, for a 5-piece



Figure 2.4: The deviation error for the 5-piece PWL approximation of $\frac{K}{1+e-0.1x}$ for K = 19, 10, and 5.

PWL approximation of $\frac{K}{1+e^{-0.1x}}$ the deviation error only depends on K. Fig. 2.4 shows the deviation error for three different values of K. The intersection points, the slopes, and the y-intercept of each basis function are shown in Table 2.2.

2.4 The Reconfigurability and the simulation results

The main advantage of the proposed neuron over the previously proposed structures is that it provides the ability to be controlled off or on-chip to generate a wide variety of transfer functions based on the requirements and applications. The form and slope of the neuron can get adjusted by externally changing voltages and programming it during the training when the neuron is used for a chip-in-the-loop or online configuration. The proposed structure is implemented in CMOS $0.18\mu m$ technology and uses the power supply of 2.5V. The area and power of the structure are measured $94.4\mu m^2$ and 0.92mW respectively.

The reconfigurability of the neuron is realized by controlling the substrate voltage of V_S of the PMOS transistors as shown in Fig. 2.2. The voltage difference between the substrate and the source of the PMOS transistors, $V_{dd} - V_S$, changes the threshold voltage of V_{th} of



Figure 2.5: The PWL neuron outputs that show the reconfigurable sigmoid functions for $V_G = 710mV$ and the linear functions for $V_G = 250mV$. Different shades of the sigmoid and the linear functions are shown for V_S of 2V, 2.3V, 2.4V, and 2.5V.



Figure 2.6: 2-bit voltage DAC corner analysis result for different conditions ff, ss, fs, and sf.

the corresponding transistors due to the body effect. The variation of the substrate-source voltage affects the drain current as a function of V_{th} . Consequently, the current ratios between the transistors of the three basis functions and the input transistor vary corresponding to the variations in the substrate voltage.

Fig. 2.5 shows the post-layout simulation results of the neuron transfer function for different values of the substrate voltage of V_S while the bias voltage, V_G , is 250mV for *linear* functions and 710mV for variable *sigmoid* functions. As shown in this figure, the lower V_S results in a higher slope for the first and the second basis functions. At $V_G = 710mV$ and $V_S = 2V$ the slopes change to the point that the *sigmoid* function ultimately reshapes to a *hard-limit*. That means the parameter *a* which controls the shape of $\frac{1}{1+e^{-ax}}$ can be controlled off-chip via the substrate voltage. For $V_S = 2V$, 2.3V, 2.4V, and 2.5V the *a* is realized 1.2, 0.3, 0.2, and 0.1 as shown in Fig. 2.5.

When the voltage V_G is lower than 300mV, both V_A and V_B go high. Consequently, only the second basis function can go through to the output, results in generating a *linear* activation function. The linear transfer functions shown in Fig. 2.5 are generated at V_G of 250mV and V_S of 2V, 2.3V, 2.4V, and 2.5V. The highest slope is correspondent to V_S of 2V as expected.

The substrate voltage of the mentioned PMOSs is controlled by a 2-bit voltage digital to analog converter (DAC) which is shown in the dashed-dotted box in Fig. 2.2. The output voltage of this block is V_S and depends on which of the transistors are on at the time. If and only if the x_1x_0 is 00, the transistor M_{18} turns off and shows a high impedance at the output node providing the output voltage of $V_S = V_{dd} = 2.5V$. When x_1x_0 is equal to 01, 10, and 11, transistors M_{16-18} turn on respectively to provide the corresponding V_S of 2V, 2.3V, and 2.4V. The output voltage of the voltage DAC vs. different values of x_1x_0 is shown Fig. 2.6.

The corner analysis for different conditions of ff (Fast NMOS Fast PMOS), ss (Slow NMOS Slow PMOS), fs (Fast NMOS Slow PMOS), and sf (Slow NMOS Fast PMOS)



Figure 2.7: Post-layout simulation results for different conditions of ff, ss, fs, and sf for the temperatures of -55C, 27C, and 125C.

is performed, and the result of that is shown in Fig. 2.6 as well to investigate the process variation effect on the output result. The maximum output fluctuation of the voltage DAC happens for $x_1x_0 = 01$ where V_S changes from 1.93V at ff to 2.07V at fs conditions. In this case, V_S is supposed to be equal to 2V and is correspondent to the *hard-limit* neuron shape and does not affect the neuron shape significantly. Moreover, the *hardlimit* neuron shape is considered the coarse tuning when used in multiresolution learning paradigm [5]; meaning that small variations in the transfer function can be compensated during the fine-tuning stages. The dimensions of transistors M_{14-18} are $\frac{0.25}{2}$, $\frac{1.25}{2}$, $\frac{1.25}{2}$, $\frac{4.2}{2}$, and $\frac{3.8}{2}$ respectively.

The process and temperature variations impact on the neuron function is investigated by performing the corner analysis for the neuron transfer function. The post-layout simulations are shown in Fig. 3.6 for a = 0.1, a = 1.2 at $V_G = 800mV$ and for the linear function at $V_G = 200mV$ and at the temperatures of 27*C*, -55*C*, and 125*C*. The deviation from the *tt* (Typical NMOS Typical PMOS) condition at the room temperature of 27*C* is represented in Fig. 2.8. As shown in this figure, the maximum deviation occurs at the *ff* condition for both -55*C* and 125*C*.



Figure 2.8: The corner and temperature post-layout analysis for different conditions of ff, ss, sf, and fs at temperatures of 27C, -55C, and 125C showing the deviation from the activation function at 27C.

Similar to other analog structures, the proposed neuron is disposed to mismatch. However, the parametric analysis simulation results show that the circuit works well by considering 10% mismatch of transistors. It should be noted that the proposed neuron when used in an on-chip or a chip-in-the-loop configuration, can enjoy some flexibility because the adaptation of the synapses during the training will be based on the actual and physical characteristics of the fabricated neuron. Moreover, the mechanism for adjusting the neuron transfer function allows more flexibility in the transistor mismatches while the network is getting trained.

2.5 Conclusion

A novel reconfigurable neuron is proposed in this paper to be used in analog and mixedsignal neural networks with different requirements. The shape of the neuron can be changed on the spot to provide different shades of *sigmoid*, *hard-limit*, and *linear* activation functions. The structure is based on the piecewise linear approximation of the desired transfer function and the controllability is obtainable by adjusting the substrate voltage of PMOS transistors. A 2-bit voltage DAC is used to adjust the shape of the neuron on or off-chip. The proposed structure is an invaluable part of the analog or mixed-signal networks that use multiresolution learning process.

References

- C. H. sai, Y. T. Chih, W. H. Wong, and C. Y. Lee, "A Hardware-Efficient Sigmoid Function With Adjustable Precision for a Neural Network System," *IEEE Trans. Circuits Syst. II*, vol. 62, no. 11 pp. 1073–1077, Nov. 2015.
- [2] Q. Liu, and J. Wang, "Finite-Time Convergent Recurrent Neural Network with a Hard-Limiting Activation Function for Constrained Optimization with Piecewise-Linear Objective Functions," *IEEE Trans. Neural Netw.*, vol.22, no.4, pp. 601–613, Mar. 2001.
- [3] T. Qiu, X. Wen, and F. Zhao, "Adaptive-Linear-Neuron-Based Dead-Time Effects Compensation Scheme for PMSM Drives," *IEEE Trans. Power Electron.*, vol. 31, no.3, pp. 2530–2538, Mar. 2016.
- [4] Yang Zhang, Yi Li, Xiaoping Wang, and Eby G. Friedman, "Synaptic Characteristics of Ag/AgInSbTe/Ta-Based Memristor for Pattern Recognition Applications, *IEEE Trans. Electron. Dev.*, vol. 64, no. 4, pp. 1806-1811, Apr. 2017.
- [5] Y. Liang and X. Liang, "Improving signal prediction performance of neural networks through multiresolution learning approach," *IEEE Trans. Syst., Man, Cybern. B*, vol. 36, no. 2, pp. 341–352, Apr. 2006.
- [6] G. Khodabandehloo, M. Mirhassani, and M. Ahmadi, "A Prototype CVNS Distributed Neural Network Using Synapse-Neuron Modules," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 7, pp. 1482–1490, 2012.
- [7] D. Moro-Frias, C. A. De La Cruz-Blas, and M. T. Sanz-Pascual, "PWL Current-Mode CMOS Exponential Circuit Based on Maximum Operator," *IEEE Antennas Wireless Propag. Lett.*, vol.5, no.1, pp. 450-453, Dec. 2006.
- [8] H. B. Demuth, M. H. Beale, O. De Jess, M. T. Hagan, Neural network design, 1996.
- [9] B. Yang, and C. A. Balanis, "Least square method to optimize the coefficients of complex finite-difference space stencils, *IEEE Trans. signal processing*, vol.45, no.11, pp. 2858-2864, Nov. 1997.

[10] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of O (n) complexity, *Advances in neural information processing systems*, pp. 703-711, 198

Chapter 3

Mixed-Signal Synapse Multipliers for Feed-Forward Neural Networks

3.1 Introduction

In Analog Neural Networks (Analog NN) [1, 2, 3] neurons can be realizing with simple and elegant non-linear analog circuits and with only a few transistors. Moreover, the addition of values can be performed by simple nodal summation of currents as long as it can drive the circuit of the next stage. However, the accuracy of analog circuits has always been a limiting factor for the realization of large size multi-layer Analog NNs. A multi-layer network requires storing a large number of synaptic values. In analog circuits, these values are typically stored on capacitors which may change due to leakage currents; hence, periodic refreshments are required. The issue of storage has been proven to cause limitation in size and complexity of such networks.

Mixed-signal approach is shown to be an intriguing choice for neural networks imple-

mentations [4, 5, 6, 7]. In such systems, advantages of both analog and digital [9, 10, 11, 8] domains are gathered in one place in order to overcome the design challenges to accomplish smaller area, lower power consumption, higher speed, and smoother activation function realization.

One of the most efficient approaches to implement the synapse in mixed-signal circuitry is based on the Multiplying Analog to Digital Converter (MDAC) which is used to multiply the synapse value by the neuron input. Conventional MDACs work based on the weighted summation of currents, that means weighted current mirrors are required in the network. Therefore, in each layer of the network with N neurons, N^2 MDAC units are required. Optimization of the size of the multiplier would significantly affect the feasible size of the network and hence its performance.

In this chapter, a programmable mixed-signal MDAC multiplier is proposed to be used in the feed-forward neural network. The proposed structure is modular and easy to be adopted for different network configurations while the area is reduced by using digital gates to ease the multiplication and avoid using large-size transistors. Moreover, synaptic weights are stored in registers which eliminate the need for capacitors and refreshing circuitries.

3.2 Neural Network Configurations

In this section, the general configuration of one layer of the mixed-signal neural network is presented. There are three main building blocks for the mixed-signal implementation of neural networks: programmable MDACs for synapse multipliers, adders, and non-linear neurons that create an integrated synapse-neuron building block.

In the proposed architecture, multiplication operation between the synaptic weights and the network inputs is performed by the MDAC, where synaptic weights are stored in digital registers and are multiplied by the analog inputs. Multiplication result of each multiplier passes through an s-shape neuron and then is added to other multiplication results coming



Figure 3.1: System-level configuration of the proposed mixed-signal neural network.

from other blocks.

Fig. 3.1 shows the block diagram of a sample 2-2-1 network. As it can be seen in this figure outputs of n building blocks are connected in parallel to generate a neuron. The number of MDACs in each layer is equal to the number of inputs to that layer.

The digital registers store the value of the synaptic weight and are programmable based on the network training. The weights are denoted by Y_{mn} in this figure, where m and n represent the number of corresponding neuron and inputs of each layer, respectively.

Since the circuit design is based on the current-mode operation, addition in the network is based on the summation of currents. Neurons are resistive non-linear functions which are distributed in the network.

The proposed network is trained off-line, where weights and network parameters are calculated off the chip and downloaded later into the weight registers. However, the network can be easily adjusted for on-line training by adding extra hardware for weight adjustment calculations.



Figure 3.2: General configuration of the mixed-signal neural network[7].

3.3 Building block's components

Neuron and its simulation result are presented in this section, followed by the proposed multiplier structure that plays an important role in reliability and accuracy of the network.

3.3.1 Neuron

Neurons for this network are resistive-type and distributed in order to increase the signal to noise ratio of the network [7, 5]. The neuron transfer function self-adjusts, preventing the saturation of neurons when the total number of input increases. The neuron uses the fundamental nonlinearity in V-I characteristics of the MOS transistors to approximate the sigmoid-like function. Fig. 3.2 represents the resistive-type neuron. The 6-transistor design [7] is biased to operate in both triode and saturation regions and has an accurate approximation to the original sigmoid function. The simulation result of the transfer function of the neuron is displayed in Fig. 3.3.



Figure 3.3: Non-linear neuron activation function which approximates the sigmoid function.



Figure 3.4: Multiplying the two less significant and the two most significant bits of Y with the analog value of the input X

3.3.2 Mixed-Signal Multiplier

In its most general form, multiplication between two binary values (X, and Y) can be preformed as follows:

$$M = \sum_{i=0}^{3} x_i 2^i \cdot \sum_{j=0}^{3} y_j 2^j$$
(3.1)



Figure 3.5: The proposed modular mixed-signal multiplier to be used in distributed feed-forward neural network

where x_i and y_j are i^{th} and j^{th} bit of X and Y respectively. In mixed-signal multiplication, one of the numbers (X) is an analog value. the synapse receives an analog input and multiplies it by a digital weight in the first layer. The multiplication result passes the S-shaped nonlinear neuron; then it is added to other multiplication results come from other branches in the first layer.

In conventional MDACs, the principle of multiplying is to use weighted current mirrors. That means that if the size of the first transistor in a weighted current mirror is W/L, we need transistors of the size of twice, four times, and eight times of W/L are required to perform the digital to analog multiplication and conversion. The proposed modular multiplier reduces the size of transistors significantly by introducing a new method to do the multi-

plication. In the proposed approach, the analog input is multiplied to two bits of the digital weight (Y). Fig. 3.4 represents the concept of multiplication of two least and most significant bits of the weight (Y) to the analog input of X separately. Based on this separation, the multiplication result from (1) can be rewritten as:

$$MA = (y_0 + 2y_1) \cdot (x_0 + 2x_1 + 2^2x_2 + 2^3x_3)$$
(3.2)

$$MB = (y_2 + 2y_3) \cdot (x_0 + 2x_1 + 2^2x_2 + 2^3x_3)$$
(3.3)

As it can be seen in equation (2), the output of the MA block can be 0, X, 2X, or 3X depending on what the value of y_0 and y_1 are. In this method, combinations of y_0 and y_1 are used to generate controlling signals that let 0, X,2X or the addition of them (3X) pass through to the output. The MB block has the same structure as the MA but it's controlling signals are generated by combinations of y_2 and y_3 .

Fig. 3.5 represents the modular architecture for a 4-bit to 4-bit equivalent mixed-signal multiplier. In this figure, S_{iA} and S_{jB} are controlling signals generated by pair of y_0 and y_1 and pair of y_2 and y_3 . Respectively. S_{0A} (S_{0B}) lets the same value of input X pass through M_1 (M_3) if y_1y_0 (y_3y_2)=01. Due to the same logic, twice of the value of X passes through $M_2(M_4)$ when y_1y_0 (y_3y_2)=10. When y_1y_0 (y_3y_2)=11, passes 1 and 2 are open and the addition of M_1 (M_3) and $M_2(M_4)$ passes through M_5 (M_6) which is output of MA (MB) block.

To confirm the validity and accuracy of the operation, the simulation result and ideal expected multiplication result for Y=1111 are compared in Table I. In case of Y=1111, MA and MB reach their maximum values that fully load the multiplier. In this case, the error and power consumption are at their maximum levels. Also, they have the same value because of the modularity of the design. As it can be seen in this table the maximum error is equal to 0.24μ A and is occurred for analog inputs of 8μ A and 9μ A. The maximum error



Figure 3.6: MA/MB output result for Y=1111for corner analysis fast-fast (ff), slow-slow (SS), slow-fast (sf) and fast-slow (fs) to show the process variation effect.



Figure 3.7: Multiplication results for Y=0010, 0111, 1010, 1011, and 1100. Ideal and simulation results are shown with dashed and solid lines respectively.

percentage is 1.3% for 4μ A. Corner analysis results represented in Fig. 3.6 shows the result considering process variations.

The multiplication results for more five digital weights and the ideal multiplication results for an input range of 0 to 15μ A is shown Fig. 4.6. The maximum error of 0.24μ A which is one-fourth of the multiplier accuracy of 1μ A. That means the error is not only within the acceptable range but also it can increase the accuracy to 5 bit.

To confirm the validity and accuracy of the operation, the simulation result and ideal expected multiplication result for Y=1111 are compared in Table I. In case of Y=1111,

X=1111				
Y	Ideal Result	Experiment result	error	error percentage
1	3=000011	2.97	-0.032	1
2	6=000110	6.05	0.051	0.8
3	9=001001	9.12	0.116	1.2
4	12=001100	12.16	0.165	1.3
5	15=001111	15.19	0.191	1.2
6	18=010010	18.21	0.216	1.1
7	21=010101	21.23	0.237	1.1
8	24=011000	24.24	0.240	0.9
9	27=011011	27.24	0.240	0.7
10	30=011110	30.23	0.230	0.7
11	33=100001	33.23	0.230	0.6
12	36=100100	36.19	0.190	0.5
13	39=100111	39.17	0.170	0.4
14	42=101010	42.12	0.120	0.2
15	45=101101	45.07	0.070	0.1

Table 3.1: Simulation and ideal multiplication result for Y=1111 at different input levels and the measured error.

MA and MB reach their maximum values that fully load the multiplier. In this case, the error and power consumption are at their maximum levels. Also, they have the same value because of the modularity of the design. As it can be seen in this table, the maximum error is equal to 0.24μ A and occurs for analog inputs of 8μ A and 9μ A. The maximum error percentage is 1.3% for 4μ A. Corner analysis results represented in Fig. 3.6 shows the result considering process variations.

The multiplication results for more five digital weights and the ideal multiplication results for an input range of 0 to 15μ A is shown Fig. 4.6.

Here, the LSB is considered to be 1μ A at the output, so, 1μ A to 15μ A were seen as equivalent to four bits. This structure is expandable for higher resolution considering the fact that the error is less than 0.5μ A. Also, every two digits increase in weights resolution needs another MA block to be added to the circuitry.

The current-based structure of this multiplier eliminates the need of extra storages to store multiplication results that consequences a huge saving in the area. Moreover, the combination of digital gates and analog circuit reduce the total area and static power consumption significantly comparing to a conventional MDAC.

Three recent conventional mixed-signal multipliers are compared to our proposed modular multiplier in Table II. The dash sign indicates that the content was not reported in the original paper.

	Proposed	[12]	[13]	[14]
Technology(um)	0.18	0.35	0.09	0.35
Power supply (V)	1.8	-	1.2	3.3
Chip area (um^2)	244	775	-	-
Power (mW)	0.23	-	406	30.73
Largest transistor (um/um)	4.5/1.4	125/0.6	-	-

Table 3.2: Simulation and ideal multiplication result for Y=1111 at different input levels and the measured error.

The area of this multiplier is $244um^2$ and the maximum power consumption is measured 0.23 mW. The small area of this multiplier makes it an excellent choice for deeplearning neural networks. Also, this highly modular and scalable VLSI architecture that can be unified with neurons structure is capable of increasing the number of synapse per die area and being used in different distributed neural network applications.

3.4 Conclusion

A modular mixed-signal multiplier architecture is implemented in CMOS 0.18μ um for multi-layer neural networks applications. The multiplier receives analog inputs and linearly multiply it with digital weights stored in registers and gives out the result as a current. This structure reduced area and static power consumption by using a new technique in multiplication that multiplies every two bits of weights separately to the whole analog value. The area and power consumption at the maximum input level of $244um^2$ and 0.23mW with the measured output current error of less than 0.5μ A respectively. Area and powerefficiency of this structure in addition to the modularity feature make this structure and an easy and an excellent choice for the neural network design especially for multi-layers one in that these criteria are in high demand. Corner analysis results confirm the robustness of this structure.

References

- C. Lu and B.-X. Shi and L. Chen, "An On-Chip BP Learning Neural Network with Ideal Neuron Characteristics and Learning Rate Adaptation," *Analog Integrated Circuits and Signal Processing*, Vol. 31, pp. 55–62, 2002.
- [2] V. F. Koosh and R. M. Goodman, "Analog VLSI neural network with digital perturbative learning," *IEEE Transaction on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 49, pp. 359–368, 2002
- [3] L. Gatet and H. Tap-Beteille and M. Lescure, "Real-Time Surface Discrimination Using an Analog Neural Network Implemented in a Phase-Shift Laser Rangefinder," IEEE Journal on Sensors, Vol. 7, pp. 1381–1387, 2007.
- [4] G. Zatorre-Navarro and N. Medrano-Marques and S. Celma-Pueyo, "Analysis and Simulation of a Mixed-Mode Neuron Architecture for Sensor Conditioning", *IEEE Transactions on Neural Networks*, Vol. 17, pp. 1332–1335, 2006.
- [5] H. Djahanshahi and M. Ahmadi and G. A. Jullien and W. C. Miller, "Quantization noise improvement in a hybrid distributed-neuron ANN architecture," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 48, pp. 842–846, 2001.
- [6] M. Mirhassani and M. Ahmadi and G. Jullien, "Robust low-sensitivity Adaline neuron based on Continuous Valued Number System," *Analog Integrated Circuits and Signal Processing*, Vol. 56, pp. 223–231, 2008.
- [7] G. Khodabandehloo and M. Mirhassani, and M. Ahmadi, "Resistive-Type CVNS Distributed Neural Networks With Improved Noise-to-Signal Ratio," *IEEE Transactions* on Circuits and Systems II: Express Briefs, Vol. 57, pp. 793–797, 2009.
- [8] B. Zamanlooy and M. Mirhassani, "Efficient VLSI implementation of neural networks with hyperbolic tangent activation function," *IEEE Trans. VLSI Syst.*, vol. 22, no. 1, pp. 39–48, January 2014.

- [9] S. Bettola and V. Piuri, "High performance fault-tolerant digital neural networks," *IEEE Transaction on Computers*, Vol.5, No.23, pp. 230–233, 1997.
- [10] D. Zhang and M.I. Elmasry, "VLSI compressor design with applications to digital neural networks," *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol.47, No. 3, pp. 1085–1091, 2006.
- [11] K. Basterretxea and J.M. Tarela and I. del Campo, "Approximation of sigmoid function and the derivative for hardware implementation of artificial neurons," *IEE Proceedings on Circuits, Devices and Systems*, vol.151, Issue 1, pp.18–24, 2004.
- [12] Z. Gafsi, N. Hassen, M. Mhiri and K. Besbes, "A New Efficient Silicon Area MDAC Synapse," *American Journal of Applied Sciences*, vol.4(6), pp.378–385, 2007.
- [13] G. Khodabandehloo, M. Mirhassani and M. Ahmadi, "16-level CVNS memory with fast ADC," *IEE Electronics Letters*, vol.45, No. 16, 2009.
- [14] Y. Su, N. Ning, and Q. Yu, "A novel 2.5bit SHA-less MDAC design for 10bit 100Ms pipeline ADC," *ICCp2011Proceedings*, 2011.

Chapter 4

Hardware Realization Of Mixed-Signal Neural Networks

4.1 Introduction

Artificial neural networks (ANN) are popular adaptive trainable systems that are employed in the vast field of applications from the prediction of nonlinear time series [1] and financial data forecasting [2] to the pattern recognition applications [3]. However, VLSI implementation of these systems faces challenges due to the complications associated with implementing a large fully parallel system especially when low complexity is required. Flexibility, area, power-efficiency, and reliability are some of the most significant challenges to overcome in the hardware implementation of such systems. Moreover, as the size and complexity of the network grows, its training becomes more difficult and it takes longer to complete due to increased number of parameters.

In terms of hardware realization, a mixed-signal implementation approach was chosen

to address the above-mentioned issues [3, 4]. This approach uses the analog circuit's advantages such as the small, well-designed neurons and the current-mode structure to simplify calculations [7, 6, 5]. In the same way, the mixed-signal structure avoids the drawbacks typically associated with the analog structures such as lower accuracy compared to the digital implementations and the large capacitors which analog circuits require to store analog weights. These storages cause more design complexity and limit the size of the implementable neural network.

In the proposed structure, neurons are divided into sub-neurons which are effective in reducing the effect of quantization noise in the circuit [8]. Moreover, the neurons that are used in this work, are effective in increasing the generalization capacity of the network. There are several methods proposed in the literature that attempt in improving the network performance and generalization capability. These include reduction of weight parameters through weight sharing [9, 10], and multi-resolution learning [11]. However, these methods are tailored for software simulations of neural networks and face challenges and difficulties for hardware implementation.

It should be noted that the Sigmoid neurons become ineffective, when the input values to a neuron increases, forcing the neuron to act more similar to a threshold neuron rather than a non-linear sigmoid neuron. The neurons used in this chapter, however, are able to self-scale their non-linear gain and do not require to be redesigned. The neuron used in this chapter is simple and suitable for hardware realization of neural networks.

In this chapter, a modular synapse-neuron building block is introduced based on a mixed-signal synapse and a distributed neuron. Due to the current-mode performance, the addition, subtraction, and division are done in the most area-efficient way. The synaptic multiplier utilizes AND gates and weighted current mirrors instead of using the weighted summation of currents as it is used in conventional DACs. The synaptic weights are stored in digital registers; consequently, the storage capacitors are avoided. The design is laid out in the TSMC CMOS $0.18\mu m$ process, and simulation results for a 4-input pattern recogni-



Figure 4.1: The resistive-type neuron circuit modified to a robust current-mode structure. The synaptic multiplier's output current is applied to the neuron as I_{in} via terminal T_1 . The output current is shaped by a self-adjustable sigmoid function.

tion are provided to prove the performance of the design.

4.2 Self-adjustable distributed Neuron

In a distributed neural network, the neurons with small areas are desirable since there is a sub-neuron for each synaptic multiplier forming a synapse-neuron module. An areaefficient resistive-type neuron was introduced in [13]; however, was sensitive to the mismatch and the process variations. Here, the resistive-type neuron has been improved to generate an output current of I_{out} from the input current of I_{in} to generate the transfer function. Moreover, the neuron required current to voltage conversions, additions, and divisions which are eliminated here. The modified circuit of the sub-neuron is presented in Fig. 4.1.

The gates of M_{14} and M_{15} should be set at $V_{dd}/2$ the gates of M_{14} and M_{15} should be set at $V_{dd}/2$ to keep the transfer function symmetric. The diode-connected transistors are used for biasing and sized in a way that they can provide the required biasing current. The voltage of $Bias_1$ is 250mV to keep the M_{18} ON even when the gate voltage is close to zero. Fig. 4.2 shows the post-layout simulation results presenting the transfer function of the improved neuron for three ranges of input currents of $(-65\mu A, 65\mu A), (-100\mu A, 100\mu A),$ and $(-200\mu A, 200\mu A)$.



Figure 4.2: The variation of the neuron's activation functions for the input ranges of $(-60\mu A, 60\mu A)$ shown by the solid line, $(-100\mu A, 100\mu A)$ shown by dotted line, and $(-200\mu A, 200\mu A)$ shown by dashed line.



Figure 4.3: The 1000 runs Monte Carlo simulation results of the current-mode neuron's activation function.

As shown in Fig. 4.2, the neuron is adjustable and can change its non-linear gain region depending on the input current. This means that the neuron's transfer function tunes itself to go into the saturation region for larger input current depending on the input range. The fact that the value of the weights might increase during the training, causes a neuron with fixed transfer function to behave similarly to a threshold neuron for larger input values. This has been shown to create difficulties during the training phase and has caused some networks to rely more on linear-style neurons. The proposed neuron can adjust its transfer function depending on the input values, and this can be achieved by the circuit without any changes in the circuit.

The robustness of the proposed structure is shown by Monte Carlo analysis. Fig. 4.3 illustrates the 1000 runs of post-layout Monte Carlo analysis results considering the process variation and the mismatch of circuits parameters.

In the following section, the neuron is used in a full network in a modular format. The network is only for a proof of concept, to test the circuit operation. However, the neuron and the modular synapse-neuron module can be used for hardware implementation of various network sizes.

4.3 distributed neural network

In this section, the system level of a distributed neural network structure is introduced and the unified current-mode synapse-neuron circuits are presented.

Fig. 4.4 shows the 4-3-2 configuration of the distributed feed-forward neural network. The dashed box represents the modular block that contains a mixed-signal synaptic multiplier and a sigmoidal neuron. As shown in this figure, the synapse-neuron block and digital registers are the only parts that are needed to form a multi-layer current-mode neural network. The j^{th} programmable synaptic weight corresponding to the i^{th} neuron in the second layer is denoted by W_{ij} which are multiplied by the current-form inputs I_{1-4} . The inputs

of the hidden layer, I_{a-c} , are multiplied by the synaptic weights corresponding to the third layer represented by C_{ij} . The biases related to each multiplier are denoted by b_{ij} .

The circuits of the synapse-neuron block are discussed in the following subsections.

4.3.1 Synaptic Multiplier

Conventional synaptic multiplier also called multiplying digital to analog converter (MDAC) principally works based on the weighted current mirrors, which leads to the use of large transistors, especially in the most significant bit [12].

In this chapter, a small-area power-efficient multiplier is used as the synapse. The 5-bit multiplier is designed by adding the sign bit to the multiplier that the authors proposed in [12]. The synaptic multiplier works based on separating the two less significant and the two most significant bits of synaptic weights and performing the signed multiplication by combining the simple digital gates (AND and NAND) and weighted currents and adding the sign bit od b_4 at the end of the structure. This method results in the two identical



Figure 4.4: The system level configuration of a 4-3-2 distributed neural network. W_{ij} and C_{ij} are the digital synaptic weights corresponding to the second and third layers respectively. I_1 to I_4 are the input currents representing the input patterns.



Figure 4.5: The modular signed multiplying DAC that performs as the synapse [12]. T_1 is connected to the terminal with the same name in Fig. 4.1 to build the synapse-neuron module.

module shown as $\underline{1}$ and $\underline{2}$ dashed boxes in Fig.4.5 which diminish the mismatch effects of transistors. Moreover, the size of the largest transistors is reduced significantly.

Fig. 4.5 shows the circuitry of the synaptic multiplier. Here, the input currents of I_{1-4} shown in Fig. 4.4 are denoted by I_{in} which is multiplied to the two least significant bits of the synaptic weight in <u>1</u> and to the two most significant bits in <u>2</u>. The addition of the output current of <u>1</u> with the four times of the output of <u>2</u> will be the multiplication result of the input current and four bits of the synaptic weight. The direction of I_{out} is determined by the sign bit of b_4 which is positive in case of a 0 and negative in the event of a 1. The maximum power consumption and the dimensions of a single multiplier are measured $328\mu W$ and $103.25\mu m \times 36.2\mu m$ respectively. The dimensions of transistors that are used



Figure 4.6: The corner analysis simulation results of tt (Typical NMOS Typical PMOS), ff (Fast NMOS Fast PMOS), fs (Fast NMOS Slow PMOS), sf (Slow NMOS Fast PMOS), and ss (Slow NMOS Slow PMOS) that show the process variation effect on the multiplication performance for the input current of $1\mu A$ that is multiplies to a digital weight that varies from -11111 to 11111.

in this neural network are listed in table 4.1. Fig. 4.6 presents the corner analysis of the

M_0, M_1, M_2, M_3	4.5/1.4	M_8, M_9	2.5/1
M_4, M_5, M_6	1/1	M_{10}, M_{11}	6/1
M_7	4/1		

Table 4.1: Sizes of the transistors of the multiplier.

results of an input of 1μ A multiplying to the 5-bit weight changing from -15 (11111) to +15 (01111). The corner analysis represents the process variation effect of the multiplier's performance and shows the maximum deviation from the ideal multiplication results occurs for the ff (Fast NMOS Fast PMOS) and is equivalent to 0.7uA, which is less than the input current, and thus the accuracy is correctly considered to be 1μ A. Via terminal T_1 the output current of this multiplier passes the neuron that is presented in the following subsection.

4.4 Pattern recognition

In this section, the performance of the 4-3-2 feed-forward neural network that is shown in Fig. 4.4 is discussed. The network is built with the synapse-neuron block that was presented

in the last section to show the feasibility of the design. This network is used to classify the pattern templates as shown in Fig. 4.8.

Here, the network is trained offline. However, the design can be modified for the chip in the loop training. The 5-bit weights and biases that were calculated off line in MATLAB are as follows:

$$W_{ij} = \begin{bmatrix} 10011 & 11000 & 01011 & 10111 \\ 11010 & 00110 & 00001 & 11110 \\ 00110 & 01111 & 00001 & 11001 \end{bmatrix}$$
$$C_{ij} = \begin{bmatrix} 11001 & 01101 & 00001 \\ 01010 & 01010 & 11001 \end{bmatrix}$$
$$b_{i1} = \begin{bmatrix} 00000 \\ 11000 \\ 00100 \end{bmatrix} b_{2j} = \begin{bmatrix} 00000 \\ 10010 \\ 10010 \end{bmatrix}$$

The output currents of the neural network are compared to a reference current at the terminals of O_1 and O_2 and provide the voltage of 0 (1) in case the current is lower (higher) than the reference. The current comparator structure used in the proposed neural network is shown in Fig. 4.7.

Fig. 4.9 shows the input currents that are introduced to the network and the output voltages Out_1 and Out_2 . As seen in this figure the outputs are as we expected for any valid combination of the input.

The layout of the neural network is presented in Fig. 6.12. The dimensions and the average power consumption are measured $318.950\mu m \times 446.150\mu m$ and 0.93mW respectively. The corner analysis results supports the notion that the network is not affected by process variation due to the tunable reference currents.

A comparison of three 4-3-2 neural networks is shown in table 4.2. The table includes the comparison between areas, power consumptions, and number of the tested templates.

Fig. 4.11 represents the sensitivity and robustness of the network performance to the most critical paths in the design. The output currents are shown for the 5% variation in

	Proposed	[13]	[8]
Technology(um)	0.18	0.18	1.2
Chip area (um^2)	142299.5	385320	-
Average Power (mW)	0.93	-	3.65
Power per synapse(mW)	0.33	5	-
Number of tested templates	6	6	4

Table 4.2: The comparison table of the proposed 4-3-2 distributed NN and other similar structures.



Figure 4.7: The structure of the current comparators that are connected to the O_1 and O_2 terminals of Fig. 4.4.

the input current (dotted lines), 5% changes in the width and length of the transistor M_6 in the multipliers of the input layer (dashed lines), and 5% variations in the width and length of the transistor M_1 8 of the second layer (solid lines). As shown in the figure, the outputs Out_1 and Out_2 remain the same while these variations are applied.

4.5 Conclusion

A mixed-signal distributed neural network designed for a pattern recognition application is implemented in TSMC CMOS 0.18μ m. The network, which uses a power-efficient synaptic multiplier consumes a low power and occupies a small area. The 5-bit digital synaptic weights are introduced to the synapse where they are multiplied with analog input currents.

Templates	Input equivalent bits	Output bits
	0101	10
	0011	01
	1010	01
	1100	10
	1001	00
	0110	11

Figure 4.8: The input templates that are used to test the functionality of the neuron.



Figure 4.9: Simulation results of the 4-3-2 distributed neural network to prove the pattern recognition capability.

Following the multiplication, the resulting current passes through the neuron which applies a sigmoid-shaped transfer function to deliver the output currents. The area of the network is measured $142299.5\mu m^2$. The average power consumption is measured 0.93mW.



Figure 4.10: The 4-3-2 mixed-signal network layout.



Figure 4.11: Simulation results of the 4-3-2 distributed neural network to show the sensitivity regarding three critical paths.

References

- Liang Y and Liang X, "Improving signal prediction performance of neural networks through multiresolution learning approach," *IEEE Trans. Syst., Man, Cybern. B*, vol. 36, no. 2, pp. 341–352, Apr. 2006.
- [2] Reid D, Hussain A, and Tawfik H, "spiking neural network for financial data prediction," *Proc. IJCNN*, pp. 1–10, Aug. 2013.
- [3] Calitoiu D, Oommen B, and Nussbaum D, "Desynchronizing a chaotic pattern recognition neural network to model inaccurate perception," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 3, pp. 692–704, Jun. 2007.
- [4] Luo C, Ying Z, Zhu X, Chen L, "A Mixed-Signal Spiking Neuromorphic Architecture for Scalable Neural Network," *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1, pp. 179–182, Aug. 2017.
- [5] Gatet L, Tap-Beteille Land, Lescure M, "Real-Time Surface Discrimination Using an Analog Neural Network Implemented in a Phase-Shift Laser Rangefinder," IEEE Journal on Sensors, Vol. 7, pp. 1381–1387, 2007.
- [6] Koosh V, Goodman R, "Analog VLSI neural network with digital perturbative learning," *IEEE Transaction on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 49, pp. 359–368, 2002.
- [7] Lu C, Shi B, Chen L, "An On-Chip BP Learning Neural Network with Ideal Neuron Characteristics and Learning Rate Adaptation," *Analog Integrated Circuits and Signal Processing*, Vol. 31, pp. 55–62, 2002.
- [8] Djahanshahi H, Ahmadi M, Jullien G, Miller W, "Quantization Noise Improvement in a Distributed Neuron Architecture," *Proc. of 40th Midwest Symposium on Circuits and Systems*, Vol. 2, pp. 1282–1285, Aug. 1997.
- [9] K. J. Lang, A. H. Waibel, "A time-Delay Neural Network Architecture for Isolated Word recognition," *Neural Network Journal*, Vol. 3, pp. 23–43, 1990.
- [10] E. A. Wan, "Time Series Prediction by Using a Connection Network with INternal Delay Lines," *Time Series Prediction:Forecasting the Future and Undersading the Past*, pp. 195–218, 1190.
- [11] Y. Liang, X. Laing, "Improving Signal Prediction Performance of Neural Networks Through Multiresolution Learning Approach," *IEEE Transaction on Systems, Man,* and Cybernetics, Vol. 36, No. 2, pp. 341–352, 2006.
- [12] Bahar Youssefi, Mitra Mirhassani, Jonathan Wu, "Efficient Mixed-Signal Synapse Multipliers for Multi-Layer Feed-Forward Neural Networks,"2016 IEEE 59th International Midwest Symposium on Circuits and Systems, 822–825, 2016.
- [13] Khodabandehloo, Golnar, Mitra Mirhassani, and Majid Ahmadi, "A prototype CVNS distributed neural network using synapse-neuron modules," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol.59, no. 7, pp. 1482-1490, 2012.

Chapter 5

Dynamic Behavior Of A Single Sigmoidal Neuron: Stable To Period Doubling

5.1 Introduction

Neural networks are getting more popular in signal processing and computation due to their ability of learning which can target different applications such as nonlinear signal prediction, time-series approximation, pattern recognition, and medical purposes [1, 2, 3]. Complex neural network systems consist of a few to a large number of neurons the dynamic behavior of which determines the performance of the system. Consequently, a deep understanding of a single neuron dynamic behavior plays a key role in understanding the nature of the neural networks and expanding different approaches towards different problems.

Neurons' activation functions fall into two main categories, artificial, and spiking which are both popular and progressive in parallel to expand the applications of neural networks. Artificial neuron's activation functions such as hyperbolic tangent, sigmoid, Poslin etc. usually vary from -1 to 1 [4] and there have been various implantation methods proposed so far [5, 6, 7]. On the other hand, spiking neurons are more similar to the neuron's biological model due to their oscillatory behavior which also makes them more difficult to be realized. The oscillatory behavior of spiking neurons can be described by a dynamic system based on a set of two coupled differential equations [8, 9]. The neural oscillator is also has been discussed by the computational model proposed by Freeman which show the oscillatory behavior of the interconnective groups of neurons [10, 11, 12]. The oscillatory behavior of coupled artificial neurons has been also discussed [13].

In this chapter, we are interested to provide the possible oscillatory behavior of a single sigmoid neuron which is much easier to be realized compared to spiking neuron and Freeman's model. The oscillatory behavior of such a simple structure may open a way towards realizing the neural oscillation without implementing the coupled differential equations.

In this chapter, we analyze the dynamic behavior of a single neuron using the bifurcation and stability maps for investigating the oscillatory behavior of the sigmoidal neuron.

5.2 Background and Theory

In this section, the theory of the dynamic behavior of a single artificial neuron in a feedback configuration is discussed. The sigmoidal neuron with the synaptic weight of β is shown Fig. 5.1. The output value of y is updated in the discrete time domain. At the instant n, part of the output signal is sampled and added to the input of x_0 by the synaptic weight of β . The addition of $x_0 + \beta y[n]$ passes through the sigmoidal activation function to generate the output at the instant n + 1. The local map of the sigmoid activation function is:

$$f(x) = \frac{1}{1 + e^{-\mu x}} \tag{5.1}$$

in which, the neuron gain, μ , is a positive number that determines the maximal slope of the sigmoid function. It should be noted that μ is a variable number and affects the dynamic

behavior of the system significantly; this effect will be discussed in detail later. Based on the power flow between the input and the output, the dynamic behavior of the system is described as follows:

$$y[n+1] = \frac{1}{1 + e^{-\mu(x_0 + \beta y[n])}}$$
(5.2)

in which, n, n + 1, n + 2, ..., are discrete time instants that present the experienced propagation delay of ΔT by the system such that $(n + 1) - n = \Delta T$.

For a given constant value of the input, x_0 , the equation (5.2) describes the transient behavior of the output. From the perspective of dynamic behavior, the system is defined as the one-dimensional map of f(y[n]) if μ , β , and x_0 are constant. Therefore, the dynamic of the system is explained by the iteration map point of view on the value range of the neuron on I = [0, 1] [13].

Due to the nonlinear activation function of the system, we expect to observe a broad spectrum of the nonlinear behavior which is described by the iteration map.

There are two main methods to study the dynamic behavior of nonlinear systems regulated by the iteration map, the linearization method, and Lyapunov stability analysis [14]. In this paper, we choose to go on with the linearization method for the local stability assessment. The linearization method is much less complex compared to Lyapunov analysis. Furthermore, developed mathematical tools for studying the dynamic behavior of linear time-invariant (LTI) systems can be employed in this approach.

The system's essential requirement that makes it adequate to use the linearization method is to have a stationary point for any arbitrary value of parameters. Based on the Brouwer fixed point theorem [15], if the local map of f is continues for any values of x_0 , β , and μ while the state space of I is a subdivision of \mathbf{R} , then the system always has a stationary solution. The existence of the stationary solution allows us to use the linearization method to investigate the systems stability. The stability map of the single neuron configuration is displayed in Fig. 5.2 for the random values of x_0 , β , and μ .



Figure 5.1: A single sigmoidal neuron in a feedback configuration.

After linearization, the next step is to discuss the stability of the LTI system by studying the locus of the eigenvalues of the system in **Z**-plane which is also known as the region of convergence (ROC).

The linearization of this system which is performed by employing the first-order perturbation stability analysis and the locus of the eigen values of our sigmoidal system are discussed in detail in the following section.

5.3 Stability Analysis of the single neuron structure

The First-order perturbation stability analysis is a well-known method to study the dynamic behavior of nonlinear systems [16] by linearizing the iterative map about a fixed-point. In this approach, the complex nonlinear system is approximated by using an exact solution of y_0 to a related but easier system through applying a small perturbation term of ε to the fixed point.

That means the solution to the complex system is approximated from the combination of the exact solution in a fixed point and the small perturbation term. Upon the condition and the parameters values of the system, the fixed point can be asymptotically stable, stable, or unstable. The fixed point of the equation (5.2) is obtained for the state variable of y



Figure 5.2: The stationary solution for the arbitrary values of $\mu = 2$, $\beta = 3$, and $x_0 = 0.55$.



Figure 5.3: The bifurcation map of the structure shown in Fig. 5.2 when $x_0 = 0.55$ and $y_0 = 0.001728$.



Figure 5.4: Stationary solution of y_0 for various β and μ values.



Figure 5.5: Stability map achieved from the system eigenvalues which show the two possible behaviours, stable and period doubling, for the system.

when it is time-independent such that:

$$y[n+1] = y[n] = y_0 \tag{5.3}$$

In this paper, equation (5.3) is solved numerically by graphing to find the stationary point due to the analytical complexity of the system. In this approach, equation (5.3) is substituted in equation (5.2) to define the objective function of G as follows:

$$G(y_0) = y_0(1 + e^{-\mu(x_0 + \beta y_0)} - 1)$$
(5.4)

The stationary solution of y_0 is the point where the objective function is globally minimum such that $|G(y_0) \approx 0|$ as shown in Fig. 5.2.

To study the dynamic behavior, we apply the first-order perturbation term of ε to the stationary solution of y_0 as follows:

$$y[n] = y_0 + \varepsilon[n] \tag{5.5}$$

By Substitution of equation (5.5) into the dynamic discerption of the system described by equation (5.2) for two consecutive time instants, equation (5.6) and equation (5.7) are achieved as follows:

$$y_0 + \varepsilon[n+1] = \frac{1}{1 + e^{-\mu(x_0 + \beta(y_0 + \varepsilon[n]))}}$$
(5.6)

$$y_0 + \varepsilon[n] = \frac{1}{1 + e^{-\mu(x_0 + \beta(y_0 + \varepsilon[n-1]))}}$$
(5.7)

If we note that ε is much smaller than y_0 , then the following equation is realized by subtracting (5.7) from (5.6):

$$\varepsilon[n+1] - \varepsilon[n] = \frac{e^{-\mu(x_0 + \beta y_0)} (e^{\mu\beta\varepsilon[n]} - e^{\mu\beta\varepsilon[n-1]})}{(1 + e^{-\mu(x+\beta y_0)})^2}$$
(5.8)

in which the higher order perturbation terms were ignored. The equation is approximated by utilizing the first-order Maclaurin expansion of $e^{\mu\beta} [n]$ and $e^{\mu\beta} [n-1]$ as follows:

$$\varepsilon[n+1] - \varepsilon[n] = \frac{e^{-\mu(x_0 + \beta y_0)}(1 + \mu\beta\varepsilon[n] - 1 - \mu\beta\varepsilon[n-1])}{(1 + e^{-\mu(x+\beta y_0)})^2}$$
(5.9)

which is summarized as:

$$\varepsilon[n+1] - \varepsilon[n] = \frac{e^{-\mu(x_0 + \beta y_0)}(\mu\beta)(\varepsilon[n] - \varepsilon[n-1])}{(1 + e^{-\mu(x + \beta y_0)})^2}$$
(5.10)

with a mathematical manipulation of $\varepsilon[n+1] - \varepsilon[n] = \delta[n+1]$ and $\varepsilon[n] - \varepsilon[n-1] = \delta[n]$ and keeping the first order perturbation terms the following equation is achieved:

$$\delta[n+1] = \frac{-(\mu\beta)e^{-\mu(x+\beta y_0)}}{(1+e^{-\mu(x+\beta y_0)})^2}\delta[n]$$
(5.11)

The perturbative elements at time instants of n + 1, n, n - 1, n - 2, ... are related together as

$$\delta[n+1] = Z\delta[n] = Z^2\delta[n-1] = Z^3\delta[n-2]$$
(5.12)

in which Z represents the eigenvalue of the system in **Z**-space. The locus of Z in ROC determines the different dynamic behaviors of the neural network. For a system with several eigenvalues, the system is stable if and only if all the eigenvalues are within the conver-

gence region. The system experiences the instability even if only one of the eigenvalues is outside of the ROC. The instable behavior of the system for different eigenvalues can be summarized as follows:

/

$$f(x) = \begin{cases} Re\{Z\} \ge 1, Im\{Z\} = 0, & \text{Bistable} \\ Re\{Z\} \le -1, Im\{Z\} = 0, & \text{Period doubling} \\ Re\{Z\} \le 1, Im\{Z\} = 0, & \text{Self-Pulsation} \end{cases}$$
(5.13)

The system is bistable when there are two stable equilibrium states which the system can relax in either of two [17] depending on the values of x_0 , μ , and β . The region of period doubling is explained from the bifurcation map point of view. Bifurcation occurs when a change in the system parameter causes the system experiencing a qualitative variation in the output. The bifurcation map related to the parameters of Fig. 5.2 is demonstrated in Fig. 5.3.

In the period doubling regime, for a single parameter of μ , two possible outputs occur. The chance that one of these states is excited is identical. The transition from one state to the other does not show the hysteresis behavior on the contrary of what occurs in bistability. Consequently, the output oscillates between these two states with the period of twice the propagation delay of the original system, ΔT .

To understand how period doubling works, Euler representation of the eigenvalue is employed as $Z = re^{j\omega\Delta T}$, where r is the magnitude of the eigenvalue and ω is the frequency of oscillation. To have period doubling (also know as Ikeda) instability, $re^{j\omega\Delta T}$ should be less than -1 [18] leading to $e^{j\omega\Delta T} = e^{j\pi}$. The phase shift can be rewritten as $2\pi f_{osc}\Delta T = \pi$, therefore, the oscillation's frequency is $T_{osc} = 2\Delta T$. This type of oscillation was observed by Ikeda et. al. in the nonlinear optical ring resonator [18].

Self-pulsation represents another form of oscillatory behavior of the system where the oscillation occurs for a constant DC input. It should be noted that the system does not need a forced oscillation from the input that has harmonic elements. the frequency of oscillation



Figure 5.6: Time domain behavior of the system for two different arbitrary sets of μ an β . (a) shows the oscillatory behavior for $\mu = 6$ and $\beta = 3$. (b) shows the stable behavior for $\mu = 0.3$ and $\beta = 0.15$.

highly depends on the system parameters such as μ and β , which causes the phase noise and jitter if is used in the local oscillator-based systems.

Among these three unstable behaviors, period doubling is promising because of its possible application as a local oscillator due to its jitter free nature of oscillation. In the period doubling regime, the frequency only depends on the propagation delay of the system while in the self-pulsation it depends on all the parameters of the system, μ , β , and x_0 .

The bistable regime has possible applications in flip-flops and latches or any other devices such as memories which need to store binary data.

5.4 Simulation Results

In this section, we present the simulation results to show the dynamic behavior of the sigmoidal neuron structure.

According to the discussion of the section 5.3, the first step to analyze the dynamic behavior is to find the stationary solution. for an arbitrary value of x_0 -let's say 0.5- the stationary solution is a function of μ and β as shown in Fig. 5.4.

Since y_0 is a function of the design parameters of μ and β , the corresponding eigenvalue location can be determined from equation (5.11). From the locations of eigenvalues which represent the dynamic behavior of the system, the stability phase map can be regulated as a function of the design parameters as shown in Fig. 5.5. According to the stability phase map, the system can be either stable or in the period doubling region.

The time domain behavior of the system is shown in Fig. 5.6. As shown in Fig. 5.6(*b*) the system rests at the steady state after a short transient state. For the period doubling region, the system oscillates between two constant values both of which depend on y_0 .

The main advantage of the oscillatory behavior shown in Fig. 5.6(a) is that the frequency of oscillation is only dependent on the propagation delay of the system and can be considered constant unless a delay is intentionally introduced to the system for the frequency tuning purposes.

The analysis, also, determines the region in which the neuron shows the oscillatory behavior. Therefore, the parameters can be chosen to increase the tolerance of the system to the process variations. The frequency of oscillation does not depend on the μ , β , or x_0 as long as the system is kept in the period doubling region which minimizes the process and fabrication effect on the oscillation frequency.

5.5 Conclusion

In this paper, the nonlinear behavior of a single sigmoidal neuron with a feedback synaptic weight is discussed. The analysis as well as the bifurcation and phase stability maps, prove that there are only two possible behaviors for the system, stable, and period doubling. The system oscillates with the period of twice of the propagation delay in the period doubling region. The oscillation's frequency does not depend on any other system's parameters except the propagation delay which suggests promising applications in the VLSI implementations of the oscillatory system by reducing the dependency to the fabrication variations. The proposed structure is the simplest neural oscillation structure that has been proposed so far.

References

- Y. Liang and X. Liang, "Improving signal prediction performance of neural networks through multiresolution learning approach," *IEEE Trans. Syst., Man, Cybern. B*, vol. 36, no. 2, pp. 341–352, Apr. 2006.
- [2] L. Ngo and J. H. Han, "Multi-level deep neural network for efficient segmentation of blood vessels in fundus images," *Electronics Letters*, vol. 53, no. 16, pp. 1096–1098, Jun. 2017.
- [3] P. Shi and F. Li, L. Wu, C. C. Lim "Neural network-based passive filtering for delayed neutral-type semi-Markovian jump systems," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 9, pp. 2101–2114, Sep. 2017.
- [4] H. B. Demuth, M. H. Beale, O. De Jess, M. T. Hagan, "Neuron Model and Network Architectures," *Neural Network Design*, 2nd edition, Boston, PWS Publishing Co. 1996.
- [5] C. H. sai, Y. T. Chih, W. H. Wong, and C. Y. Lee, "A Hardware-Efficient Sigmoid Function With Adjustable Precision for a Neural Network System," *IEEE Trans. Circuits Syst. II*, vol. 62, no. 11 pp. 1073–1077, Nov. 2015.
- [6] Q. Liu, and J. Wang, "Finite-Time Convergent Recurrent Neural Network with a Hard-Limiting Activation Function for Constrained Optimization with Piecewise-Linear Objective Functions," *IEEE Trans. Neural Netw.*, vol.22, no.4, pp. 601–613, Mar. 2001.
- [7] T. Qiu, X. Wen, and F. Zhao, "Adaptive-Linear-Neuron-Based Dead-Time Effects Compensation Scheme for PMSM Drives," *IEEE Trans. Power Electron.*, vol. 31, no.3, pp. 2530–2538, Mar. 2016.
- [8] X. Wu, V. Saxena, K. Zhu, S. A. Balagopal, "A cmos spiking neuron for brain-inspired neural networks with resistive synapses and in situ learning." *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 11, pp. 1088–1092, Nov. 2015.
- [9] G. Indiveri and S. Fusi, "Spike-based learning in VLSI networks of integrate-and-fire neurons," *in IEEE Int. Symp. Circuits Systems (ISCAS 2007)*, pp. 33713374, May 2007.

- [10] W. Freeman, J. C. Principe Mass Action in the Nervous System. New York: Academic, 1975.
- [11] G.N.Borisyuk, R.M. Borisyuk, A.B.Kirillov, V.I.Kryukov, and W.Singer," Modeling of oscillatory activity of neuron assemblies of the visual cortex.," *In Neural Networks, IJCNN International Joint Conference*, pp. 431-434. Jun. 1990.
- [12] D. Xu,J. C. Principe "Dynamical analysis of neural oscillators in an olfactory cortex model," *IEEE transactions on neural networks*, vol. 15, no. 5, pp.1053–1062, Sep. 2004.
- [13] X. Wang, "Period-doublings to chaos in a simple neural network: An analytical proof," *Complex Systems*, vol. 5, no. 4, pp. 425–441, 1991.
- [14] H. K. Khalil, J. C. Principe Noninear Systems Prentice-Hall, New Jersey, no.5, 1996.
- [15] D. Gale, "The game of Hex and the Brouwer fixed-point theorem," *The American Mathematical Monthly*, vol. 86, no. 10, pp. 818–827, Dec.1979.
- [16] P. Kokotovic, H. K. Khalil, and J. O'reilly *Singular perturbation methods in control: analysis and design*. Society for Industrial and Applied Mathematics, 1999.
- [17] Y. Jia, and J. R. Li, "Steady-state analysis of a bistable system with additive and multiplicative noises," *Physical Review E*, vol. 53, no. 6, pp. 5786, Jun. 1996.
- [18] K. Ikeda, and J. R. Li, "Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system," *Optics communications*, vol. 30, no. 2, pp. 257– 261, Aug. 1979.

Chapter 6

Low-Power Mixed-Signal Implementation of the DA-based FIR Filter

6.1 Introduction

For portable devices that are used for dynamic signal processing, stability, power, and area efficiency are ever-present need. Feeding these needs means a reduction in system size and the off-chip communication while increasing the battery life which are the main design concerns in portable devices. FIR filters are usually used at the early stage of dynamic signal processing applications due to their stability. Since the filter is one of the largest components in the system, it is vital to be designed low-power and area-efficient.

In digital signal processing, the inner product, which is the essence of the many processing functions including FIR filters, is characteristically realized based on multiplyaccumulate (MAC) operations. Although the MAC units can be easily programmed, they negatively affect the throughput of the filter, especially of the high order ones. The lower throughput means the computation needs the higher clock rate for the system which increases the power consumption. In fact, the computation time and the number of required MAC operations increase linearly with the length of the input vector and the filter order respectively. Consequently, implementation of a real-time and low-power filter would be a challenging task as the order increases [1, 2, 3, 4].

Distributed Arithmetic (DA) [3, 4, 5, 6, 7] is an efficient alternative for decreasing the power consumption in real-time applications. In this method, multipliers are replaced by adders and shift registers and multiplication is performed in fixed cycles of time while coefficients are stored on the chip. The fixed-cycle performance makes the structure computationally efficient especially when the input length is large. In digital signal processing, despite the computation efficiency, the DA approach would not be area efficient compared to MAC due to a considerable number of memory units that it must use. This problem could be eased by using mixed-signal implementation of DA Multiplying [7, 8], or switched-current techniques [3, 9, 10, 11] with limitations on power and speed [8].

The mixed-signal approach proposed in this chapter provides a subtle solution to large area occupation problem of DA-based structures by focusing on the processing stage rather than the analog storages. To demonstrate the efficiency of our approach, we implemented a 16-tap 8-bit adaptable current-mode FIR filter with a low area and power consumption. An LPF and a BPF are realized at different sampling frequencies to prove the efficiency of the proposed structure,

In this chapter, a new structure for the current-mode mixed-signal FIR filter is proposed and implemented based on the DA. The proposed structure is low-power and area-efficient taking advantage of the current-mode and DA-based structures. An LPF and a BPF are implemented to prove the efficiency of the proposed structure.



Figure 6.1: The proposed DA architecture for a 16-tap 8-bit mixed-signal FIR filter. x_{ij} is the j^{th} bit of the i^{th} input X_i . I_{Ci} is the i^{th} filter coefficient and y[n] is the final output current. (a) The compelete current-mode DA architecture (b) The structure at the *high* state of the first N - 1 clock cycles. (c) The structure at the *low* state of the first N - 1 clock cycles. (d) The structure at the N^{th} clock cycle when the operation is done.

6.2 Distributed Arithmetic

The DA concept [12] is used for the calculation of the inner product of two vectors in a bit-serial mode, in which the output Y[n] is generated by the addition of the delayed and weighted samples of the digital input X[n].

An inner product which represents an FIR filter is computed as follows [4, 7]:

$$Y[n] = \sum_{i=0}^{M-1} I_C[i] \cdot X[n-i]$$
(6.1)

where I_{Ci} and M denote filter coefficients and number of taps respectively.

To achieve the DA formulation, X is represented in 2's-complement format. Assuming that X[n] is an N-bit word, it can be represented by separating it's sign bit in the following form:

$$X[n-i] = -x_{i0} + \sum_{j=1}^{N-1} x_{ij} 2^{-j}$$
(6.2)

in which x_{i0} is the most significant bit of i^{th} component in X[n] vector indicating sign, and x_{ij} is the j^{th} bit of i^{th} component.

By substituting the X[n] from (6.2) in (6.1), Y[n] can be written as follows [3]:

$$Y[n] = -\sum_{i=0}^{M-1} I_{Ci} x_{i0} + \sum_{j=1}^{N-1} 2^{-j} \sum_{i=1}^{M-1} x_{ij} I_{Ci}$$
(6.3)

There are N - 1 clock cycles of divisions and feedbacks required to realize the second term of equation (6.3) utilizing shift registers, multipliers, and adders [3, 4, 7].

At the N^{th} clock cycle, the last input is subtracted from the feedback to generate the first term of (6.3). At the $N + 1^{th}$ clock cycle, the system resets to set the feedback zero and getting ready for the next operations.

That means N-bit serial input M-tap filter which works based on the DA concept re-

quires fixed number of clock cycles, N to perform the filtering operation. It should be noted that the same filter which is MAC-based needs M MAC units in typical DSP approaches.

The DA-based approach offers lower power consumption of the filter especially of the higher order ones. The mixed-signal structure proposed in the following section offers even more efficiency by simplifying computations and a hardware realization that is needless of large memories.

6.3 Proposed Current-Mode Distributed Arithmetic Structure

The proposed mixed-signal DA architecture is composed of the following main components: sixteen (number of taps) 8-bit (number of bits of the digital input) digital shift registers to introduce digital inputs to the filter, eight 5-bit shift registers to store the multiplicands, eight 5-bit digital to analog converters (DAC), and two current delay/divider cells.

It should be noted that the number of multiplicands storages and DACS are reduced from sixteen to eight according to the symmetry of the filters coefficients.

The configuration of the proposed DA structure is showed in Fig. 6.1(*a*). As shown in this figure, two's complement inputs, X_0 to X_{15} , are serially fed to the system through the 8bit shift registers. At j^{th} clock cycle, the least significant bit of i^{th} input (x_{ij}) is multiplied by the current-form multiplicand I_{Ci} which is the output of the i^{th} DAC. Current-mode multiplication results are added to each other through the connection node SUM. The filtering computation is performed in N = 8 clock cycles. Figures 6.1(*b*) to (*d*) show the performance of the delay and division section in various clock cycles and present the filtering process in more details as follows.

Fig. 6.1(b) displays the performance of the delay and division section during the *high* state of the first N-1 = 7 clock cycles. In this period of time, S_1 and S_2 are close and open



Figure 6.2: Multiplying stage of the proposed mixed-signal filter.









Figure 6.4: Overall conceptual operating waveforms of the proposed filter. The notations show the signal level at the specific clock cycle.

respectively and the output current of I_{out} feeds back to the first current delay cell, D_1 . The output of D_1 and D_2 pass the current trough if and only if their inputs are disconnected first. Consequently, during the *high* state of the first N - 1 = 7 clock cycles, D_1 stores the current while it is disconnected from D_2 . Simultaneously, the current stored in D_2 in the previous clock cycle is added to the current coming from the node SUM, I_S .

Each of D_1 and D_2 stores the current for half a clock cycle releases it in the second half. Fig. 6.1(c) presents what happens at the *low* state of the first N - 1 = 7 clock cycles where the current stored in D_1 is divided by two and fed to D_2 .

The equation (6.3) is fully generated at the N^{th} clock cycle by subtracting I_s from the feedback current, I_f as depicted in Fig. 6.1(*d*). At this time, I_S and I_f are equal to the first and the second terms of the equation (6.3) respectively. The output current of this step is the final result for the filter.

The CLK signal controls the speed of the operation by controlling the input shift registers and delay cells. The RESET signal opens the switch S_1 at the $N - 1^{th}$ clock cycle to cut off the feedback branch and clear the data at the end of each N cycle stream.

In the proposed structure, the inverse of some of the controlling signals such as CLK and RESET are also noted. It does not mean that \overline{CLK} and \overline{RESET} are really utilized as the controlling signals but CLK and RESET control PMOS switches instead of NMOS ones.

6.4 **Proposed Filter Implementation**

In the previous section, the basis of the DA-based FIR filter was discussed. In this section, the novel implementation of a 16-tap 8-bit FIR filter based on the DA is proposed.

Fig. 6.2 and Fig. 6.3 show the proposed filter structure. Fig. 6.2 presents the multiplying stage at the output of that connects to the processing stage circuitry shown in Fig. 6.3.

Filter coefficients, I_{Ci} , are signed values that are fed to the circuit in the current form

through 5-bit shift registers and DACs as presented in Fig. 6.2. The negative sign means a change in the direction of the coefficient currents.

Current-mode multiplication between the filter coefficient (I_{Ci}) and the j^{th} bit of i^{th} input (x_{ij}) is performed through weighted binary switches illustrated in Fig. 6.2. The input x_{ij} lets the flow of I_{Ci} in case of a 1 or opens the switch and cuts off the current in the event of a 0. The multiplied currents are added together by connecting the node SUM. The addition is performed considering coefficients' signs. That means the direction of the addition result, I_s , can be leftward or rightward at the node SUM depending on the values of x_{ij} at the existing clock cycle.

The addition result, I_s , reaches it's smallest possible negative value when all weighted binary switches related to positive I_{Ci} s are open and all switches related to negative I_{Ci} s are close. In this case, I_s would be equal to the addition of negative coefficients flowing in the leftward direction. It should be noted that the leftward direction is unwanted because it is not compatible with the following stage, delay/division.

To avoid this problem and keep the I_s positive in any condition, a compensating current of I_b is added to the coefficient currents at the node SUM. The value of I_b is equal to the absolute value of the addition of all negative coefficients, consequently, I_S is guaranteed to be positive and ready to go to the delay/division stage.

The current mirrors generating I_b pass the current if and only if the corresponding sign bit is 1. In this way, the I_b has the minimum required value to keep I_s positive which is necessary to minimize the power consumption when programming the filter with different coefficients. It should be noted that adding a constant value of I_b to the coefficients does not affect the filter performance as will be proven in the following.

The following equation below is derived from equation (6.3) considering the presence of the constant current I_b :

$$Y = -\sum_{i=0}^{M-1} (I_{Ci}x_{i0} + I_b) + \sum_{j=1}^{N-1} 2^{-j} \sum_{i=1}^{M-1} (x_{ij}I_{Ci} + I_b)$$
(6.4)

The equation (6.9) can be written as:

$$Y = -I_b - \sum_{i=0}^{M-1} I_{Ci} x_{i0} + \sum_{j=1}^{N-1} 2^{-j} \sum_{i=1}^{M-1} x_{ij} I_{Ci} + I_b \sum_{j=1}^{N-1} 2^{-j}$$
(6.5)

The geometric series of $\sum_{j=1}^{N-1} 2^{-j}$ converges absolutely to 1 for an infinite series. For a 16-tap filter, the mentioned geometric series converges to 1 with the error of %0.003 which is negligible in the filter performance. Consequently, (6.5) can be rewritten to:

$$Y = -I_b - \sum_{i=0}^{M-1} I_{Ci} x_{i0} + \sum_{j=1}^{N-1} 2^{-j} \sum_{i=1}^{M-1} x_{ij} I_{Ci} + I_b$$
(6.6)

In which, $-I_b$ and I_b cancel out each other which makes the equations of (6.3) and (6.6) equal.

Going back to the filter' structure, the positive I_s enters the delay/division stage through the transistor M_s at every clock cycle as depicted in Fig. 6.2.

A case study of a random I_s generated by random digital inputs is compared to the clock and other relevant operating waveforms in Fig. 6.4. At every clock cycle, I_s is added to the delayed feedback current of I_f which has the initial value of zero as shown in Fig. 6.4. The summation results, $I_{in} = I_s + I_f$, is delayed and divided by two in each clock cycle for the first seven clock cycles while the *RESET* switch is close to generate I_f for the next clock cycle, such that $I_f(T) = Iin(T - 1)$. The division by two is achieved by a current mirror at the feedback branch, in which width of M_2 is half of that of M_1 . The delay and division steps take one clock cycle to generate the feedback current, I_f , which is added to the new I_s at the next rising edge of the clock. It should be noted that I_f is zero at the beginning of every 8 clock cycles stream because of the *RESET* switch.

At 8^{th} clock, the switches related to \overline{RESET} are close which change the direction of I_s through the transistor M_9 and let the subtraction happens instead of the addition, such

that $I_{out} = I_{in}(T_8) = I_f(T_8) - I_s(T_8)$.

The direction of the output current, I_{out} , can be positive or negative depending on which one of I_s or I_f is higher. To take the direction of I_{out} into consideration, a current-mode comparator (see the dashed-box in Fig. 6.3) produces the *Control* signal which goes low in case of $I_f < I_s$. In this case, the positive I_{out} flows through transistor M_3 . When I_{out} is negative, the *Control* signal goes high and M_4 provides the path for the negative current to flow.

In the following subsections, the DAC and delay/division stage performance are discussed.

6.4.1 DAC

To generate the coefficient currents of I_{Ci} from the digital coefficients of Y_i , a group of eight 5-bit DACs are utilized. The configuration of the DAC employed in the proposed filter structure is shown in Fig. 6.5. This design is the modified architecture of the MDAC that proposed in[13] which utilizes the combination of AND gates and weighted current mirrors to reduce the area and static power consumption compared to the conventional MDACs [13].

As shown in the dash-dotted box in Fig. 6.5, a reference current of I_{ref} is fed to the input of the DAC. Here, I_{ref} is $1\mu A$ and equivalent to the one-bit weighted current. The reference current of $1\mu A$ is then multiplied to the digital coefficient value of $Y = y_4 y_3 y_2 y_1 y_0$ through the AND gates and weighted current mirrors. The direction of I_{Ci} is determined by the sign bit of y_4 and can be leftward in the case of $y_4 = 1$ or rightward in the event of $y_4 = 0$.

As shown in Fig. 6.5, if all the bits of Y are zeros the output of the DAC would be zero because all the switches of SW_{10-22} are open. However, there would be a power dissipation due to the reference current producer. To eliminate the power consumption due to the reference current, the transistor M_{12} turns on by the OR of Y bits. Therefore, M_{12} is on letting I_{ref} flow if and only if one the bits of Y is 1.



Figure 6.5: The 5-bit DAC structure [13]



Figure 6.6: The 5-bit DAC output current of I_{Ci} (solid line), the exact error calculated from $Error(\mu A) = I_{Ci} - I_{ideal}$ shown by dashed line, the error percentage calculated from $\frac{100 \cdot (I_{Ci} - I_{ideal})}{I_{ideal}}$ shown by dash-dotted line.



Figure 6.7: The family plot of the DAC output current vs. the analog equivalent of the digital input achieved from 500 runs of Monte Carlo simulations.



Figure 6.8: The input and output currents of two cascaded delay cells and a current divider for four random input currents of $4.98\mu A$, $20.02\mu A$, $60\mu A$, and $99.98\mu A$..



Figure 6.9: Corner analysis of tt (Typical NMOS Typical PMOS), ff (Fast NMOS Fast PMOS), fs (Fast NMOS Slow PMOS), ss (Slow NMOS Slow PMOS), and sf (Slow NMOS Fast PMOS) of the error percentage occurs in the feedback branchs output current.

Table 6.1: DAC transistors dimentions.

M_0, M_1, M_2, M_3	1/3	M_8, M_9	2.5/1
M_4, M_5, M_6	4.5/3	M_{10}, M_{11}	6/1
M_7	6	$M_{12}, M_{13}, M_{14}, M_{15}$	3/2.5

The transistor M_{12} turns on in the triode region while the diode connected transistors of M_{13-15} are in the saturation region. The transistor M_{15} has the same size as M_{12} and added to the design to decrease the V_{GS} of M_{12} furthermore leading to reduce the reference current to the desirable value of $1\mu A$ without increasing the sizes of M_{14} and M_{16} . The dimensions of the DAC transistors are shown in table 6.1.

Fig. 6.6 shows the output results of the DAC achieved from the post-layout simulation. As shown in this figure, the output current of I_{Ci} varies from $-15\mu A$ to $15\mu A$ corresponding to the coefficients changing from 11111 to 01111. The exact error is measured by subtracting the expected output of I_{ideal} from the measured value of I_{Ci} such that $Error(\mu A) = I_{Ci} - I_{ideal}$. The exact error is too small compared to the output of the DAC, I_{Ci} , and cannot be presented properly in the same figure (See the dashed line in Fig. 6.6). Consequently, the error percentage (dash-dotted line) is calculated from $\frac{100 \cdot (I_{Ci} - I_{ideal})}{I_{ideal}}$ to provide more readable error measurements.

The effect of the components' mismatch and process variation on the accuracy of the designed DAC is shown by the 500 runs of Monte Carlo simulations the results of that are illustrated in Fig. 6.7.

The DAC output error causes nonideality in the filter performance which is modeled in the filter equation as follows:

$$Y[n] = -\sum_{i=0}^{M-1} (I_{Ci} + \delta_i) x_{i0} + \sum_{j=1}^{N-1} 2^{-j} \sum_{i=1}^{M-1} x_{ij} (I_{Ci} + \delta_i)$$
(6.7)

in which, δ_i is the error corresponding to the coefficient I_{Ci} . The error can be achieved by the subtraction of equation (6.7) from (6.3)as follows:

$$E = -\sum_{i=0}^{M-1} \delta_i x_{i0} + \sum_{j=1}^{N-1} 2^{-j} \sum_{i=1}^{M-1} \delta_i x_{ij}$$
(6.8)

The error value varies from the minimum of $-\delta_0 + (-\delta_1 + \delta_1) + (-\delta_2 + \delta_2) + \dots + (-\delta_{M-1} + \delta_{M-1}) = -\delta_0$ in case that all bits are 1 to the maximum of $\sum_{i=0}^{M-1} \delta_i$. The minimum error can be reduced to zero if all bits are 1 except x_{00} .

The error in the DACs outputs of I_{Ci} can introduce ripple in the pass-band, reduce the pass-band width of the filter, and decrease the stop-band attenuation [7]. Keeping the coefficients error within the current range equivalent to one bit reduces these unwanted effects. In our design, the current range equivalent to one bit is $1\mu A$ and due to the results shown in Fig. 6.5 and Fig. 6.6 the error introduced by the DAC is less than $1\mu A$.

6.4.2 Current-Mode Delay Cell

As explained in section 6.3, the delay/division steps are needed to be repeated 8 times to generate the output of the filter. That means the accuracy of the two cascaded delay cells

of D_1 and D_2 plays a key role in the filter performance. As shown in Fig. 6.3, the cascaded delay cells of D_1 and D_2 work with the controlling signals of CLK and \overline{CLK} respectively and connected to each other by a current mirror.

Although the filter structure can afford all the inverse controlling signals, *signals*, to be generated by utilizing a *NOT* gate in the path of the corresponding controlling signal, here, we use PMOS switches to be controlled by *signals* instead of by $\overline{signals}$ to avoid the delays introduced to the signal paths by *NOT* gates and to take advantage of better time matching.

The delay cell (consider D_1) works based on the transistor M_5 and the capacitors. The transistor M_5 needs to be able to store the input current of I_{in} at the *high* level of the *CLK* signal and release it at the *low* level of that. At the *high* level of the clock cycle, switches S_a , S_b , and S_c are closed and I_{in} is charging capacitors C_{1-3} . When capacitors are fully charged, the current flowing through them is zero (sampling stage). That means the whole I_{in} passes through M_5 and M_6 . The cascaded transistor M_6 working in the saturation region is utilized to reduce the channel length modulation effect and to keep the V_{ds} of M_5 constant while it is disconnected from the input current at the low level of *CLK* (hold stage).

At the low level of the clock, M_8 is on because of the capacitors connecting to it's gate keeping the current of M_5 at the same value it had at the *high* level of the clock. In fact, the source follower M_8 is utilized to prevent the changes at the output current to affect the sampled current.

In the sampling stage, the switch S_b is closed slightly earlier than S_c which itself is closed slightly earlier than S_a to reduce the switches charge injection using feedthrough techniques. Otherwise, the charge injection changes the voltage stored in C_1 by ΔV which consequently changes the stored current by $g_m \Delta V$ [14, 15].

Considering this technique, the values of capacitors are set considering the following



Figure 6.10: The 1000 runs Monte Carlo simulation family plot of the second delay cell output current for four random input currents of 100μ , $58\mu A$, $24\mu A$, and $6\mu A$.

rules [14]:

$$C_1 = C_2 = 10C_3 \tag{6.9}$$

Here, capacitors' sizes are chosen to be $C_1 = C_2 = 100 fF$ and $C_3 = 10 fF$. Biasing voltages, V_{bias1} and V_{bias2} are 1.4V and 560mV respectively to keep M_5 biased in the saturation region.

Fig. 6.8 displays the post-layout simulation results of the two cascaded delay cells of D_1 and D_2 , and the current divider for four sample input currents of $91.55\mu A$, $45.09\mu A$, $19.55\mu A$, and $4.38\mu A$ in CMOS $0.18\mu m$ technology. Each of the delay cells detains the current for half of the clock cycle. The output current of the first delay cell, I_{D1} , is equal to $\frac{1}{2}I_{in}(T-1/2)$ and is introduced as the input current to D_2 . The feedback branch output



Figure 6.11: The frequency and phase responses of the DA-based BPF and LPF.

current is shown as $I_f = I_{D2}$.

The ideal output current of the feedback branch is expected to be equal to the half of the input current at one clock cycle earlier. Consequently, the error percentage is calculated as follows:

$$E = \frac{I_{D2}(T-1) - I_{in}(T)/2}{I_{in}(T)/2}$$
(6.10)

Fig. 6.9 provides the error percentage for the input current range varies from 0 to $100\mu A$ to evaluate the performance of the filters feedback branch. Using the mentioned formula the error percentage for Fig. 6.8 (*a*), (*b*), (*c*), and (*d*) are calculated 1.38%, 0.56%, 0.8%, and 1.12%.

The corner analysis results are presented in Fig. 6.9 to estimate the effect of the variation of fabrication parameters on the feedback branch performance. As shown in this figure, the maximum error percentage is 2.3% which occurs for ff (Fast NMOS Fast PMOS) and fs (Fast NMOS Slow PMOS) corner analysis at the input current of $1\mu A$ and for ss (Slow NMOS Slow PMOS) at the input current of $100\mu A$. For the most values of input currents, error percentage is less than 1% considering fabrication parameters variation.

Fig. 6.10 shows the 1000 runs Monte Carlo family plot of the output current of the second delay cell for four random input currents of 100μ , $58\mu A$, $24\mu A$, and $6\mu A$ to represent the mismatch and process variation effect on the performance of the feedback branch.

The error introduced by the feedback branch affects only the second part of the equation 6.3. The error can be modeled as a constant error to which another constant error is added at every clock cycle as shown as follows:
$$Y = -(I_{C0}x_{00} + I_{C1}x_{10} + \dots + I_{C(M-1)}x_{i(M-1)}) + 2^{-1}(x_{11}I_{C1} + x_{21}I_{C2} + \dots + x_{(M-1)1}I_{C(M-1)} + \delta_{1}) + 2^{-2}(x_{12}I_{C1} + x_{22}I_{C2} + \dots + x_{(M-1)2}I_{C(M-1)} + \delta_{1}2^{-1} + \delta_{2}) + \dots + 2^{-(N-1)}(x_{1(N-1)}I_{C1} + x_{2(N-1)}I_{C2} + \dots + x_{(M-1)(N-1)}I_{C(M-1)} + \delta_{1}2^{-(N-2)} + \delta_{2}2^{-(N-3)} + \dots + \delta_{N-1})$$
(6.11)

In which, δ_j is the error which comes from the j^{th} feedback in the filter. The deviation of the equation (6.11) from the equation (6.3) is the error introduced by the feedback branch calculated by:

$$E = 2^{-1}\delta_1 + 2^{-2}(\delta_1 2^{-1} + \delta_2) + \dots + 2^{-(N-1)}$$

$$\cdot (\delta_1 2^{-(N-2)} + \delta_2 2^{-(N-3)} + \dots + \delta_{N-1})$$
(6.12)

Which can be rewritten as:

$$E = (2^{-1}\delta_1 + 2^{-2}\delta_2 + \dots + 2^{-(N-1)}\delta_{N-1})$$

.(1 + 2⁻² + 2⁻⁴ + \dots + 2^{-(N-2)}) (6.13)

By substituting (6.13) in (6.11) and using the geometric series of $\sum_{j=1}^{N-2} 4^{-j}$, (6.11) is rewritten as:

$$Y = -\sum_{i=0}^{M-1} I_{Ci} x_{i0} + \sum_{j=1}^{N-1} 2^{-j} \left(\delta_j \frac{1 - \frac{1}{4}^{N-2+1}}{1 - \frac{1}{4}} + \sum_{i=1}^{M-1} x_{ij} I_{Ci} \right)$$
(6.14)

Based on (6.13) and (6.14), the maximum error in the output of an N-bit filter caused by the error introduced by analog circuits of the feedback branch is calculated considering $\delta_1 = \delta_2 = \cdots = \delta_{N-1} = \delta_{max}$ and using geometric series of $\sum_{j=1}^{N-1} 2^{-j}$ and $\sum_{j=0}^{\frac{N-2}{2}} 4^{-j}$ as follows:

$$E = \delta_{max} \left(\frac{1 - \frac{1}{2}^N}{1 - \frac{1}{2}} - 1 \right) \left(\frac{1 - \frac{1}{4}^{\frac{N-4}{2}}}{1 - \frac{1}{4}} \right)$$
(6.15)

Assuming the error that is added to the feedback branch at every clock cycle is equal to the maximum possible error, the maximum total error generated in the feedback branch is $0.69\mu A$ which occurs for $I_{in} = 100\mu A$. For an 8-bit filter, the maximum output error due to the feedback branch error is $0.69\mu A \cdot 1.24 = 0.86\mu A$. The error would be $0.92\mu A$ for the infinite number of bits.

6.5 **Results Discussion**

In this section, the results of the proposed 16-tap 8-bit FIR filter implemented in CMOS $0.18\mu m$ technology are presented. The DACs provide the external access to the filter coefficients for a reconfigurable structure. A band-pass and a low-pass FIR filter are implemented to show the adjustability of the proposed architecture.

For the BPF, the sampling frequency of f_s is 10MHz, therefore, based on our discussion in sub-section 6.4.2, the *CLK* period is set to 100ns.

The LPF is designed with f_s of 48KHz, while for both filters, the input precision and number of taps are eight and sixteen respectively.

Table 6.2 demonstrates the coefficients of both filters. Ideal coefficients are achieved from MATLAB for the 16-tap filters with the defined pass-band and stop-band. To convert these coefficients into the (mapped) currents within the affordable range of the utilized DACs, they are all multiplied to a constant to keep the consistency of the coefficients. The inputs of the DACs (DAC(in)) are chosen considering the mapped current values to provide the closest 5-bit number that can generate the similar mapped current value. As an instant, the input of the DAC that provides the closest value to mapped current of $4.8\mu A$ is 00101 which generates the current of $5\mu A$. DAC (out) denotes the measured output currents of

BPF				LPF			
ideal	$\begin{array}{c} \text{mapped} \\ (\mu A) \end{array}$	DAC(in)	DAC(out) (µA)	ideal	$\begin{array}{c} \text{mapped} \\ (\mu A) \end{array}$	DAC(in)	DAC(out) (µA)
-0.02424	-1.29	10001	-1.02	-0.04843	-1.74	10010	-2.06
+0.02885	1.53	00010	2.11	0.03164	1.14	00001	1.01
+0.02091	1.11	00001	1.02	0.06631	2.38	00010	2.09
+0.02114	11.26	01011	11.34	0.01611	0.58	00001	1.02
+0.09029	4.8	00101	5.04	-0.07617	-2.74	10011	-2.98
-0.16071	-8.56	11000	-8.26	-0.04178	-1.5	10001	-1.02
-0.24275	-12.93	11101	-13.21	0.18376	6.61	00111	7.15
+0.28154	15.01	01111	15.1	0.41718	14.99	01111	15.13
+0.28154	15.01	01111	15.1	0.41718	14.99	01111	15.13
-0.24275	-12.93	11101	-13.21	0.18376	6.61	00111	7.15
-0.16071	-8.56	11000	-8.26	-0.04178	-1.5	10001	-1.02
+0.09029	4.8	00101	5.04	-0.07617	-2.74	10011	-2.98
+0.02114	11.26	01011	11.34	0.01610	0.58	00001	1.02
+0.02091	1.11	00001	1.02	0.06630	2.38	00010	2.09
+0.02885	1.53	00010	2.05	0.03164	1.14	00001	1.01
-0.02424	-1.29	10001	-1.02	-0.04843	-1.74	10010	-2.06

Table 6.2: Filter's Coefficients.

the DAC while all connected to each other at the node SUM.

The digital random numbers are serially introduced to the circuit through the sixteen 8-bit shift-registers as the input of the filters. At each clock cycle, the inputs bits move forward to complete the filtering cycle. The magnitude and phase responses of the BPF and the LPF are illustrated in Fig. 6.11. The number of data points that are collected to get these frequency responses are 2184. Process variation is considered and represented in these simulations by performing corner analysis. The phase responses of these symmetrical filters are shown to be linear within the pass-band. The solid lines in Fig. 6.11(c) and (d) represent the ideal phase and the dotted lines show the post-layout simulation results.

The layout of the proposed design is shown in Fig. 6.12 and the area of the DA architecture considering shift registers and DACs is $0.071mm^2$. The maximum power consumption is measured 2.2mW. Table 6.3 provides a summary of the performance of the proposed implementation and a comparison between different implementations of FIR filters in the number of taps, sampling frequency, power dissipation, supply voltage, and the technology node.



Figure 6.12: The layout of the 8-bit 16-tap mixed-signal filter based on DA.

Filter	# taps	Technology node	Power consumption	Sampling frequency	Supply voltage	Area (<i>mm</i> ²)

Table 6.3: Comparison of the proposed filter with recent published filters

		noue	consumption	inequency	voltage	(111111)
Proposed	16	$0.18 \mu m$	2.2mW	10MHz	1.8	0.071
[7]	16	$0.5 \mu m$	16mW	50kHz	5	1.125
[9]	4	$0.8 \mu m$	—	1MHz	2	1.3
[16]	5	$0.18 \mu m$	3.6542mW	—	5	—
[17]	6	90nm	4.35mW	—	1	0.239
[18]	4	$0.18 \mu m$	4.1mW	—	1.8	0.52

The proposed filter's power consumption is significantly lower than that of the similarly implemented FIR filters. This difference becomes more significant when it takes into consideration that increasing order of the filter would increase the power consumption. Moreover, high sampling frequency increases the power consumption of digital sections of these circuits such as the shift registers and switching transistors. It should be mention that the sampling frequency is not reported for some of these works. The cut-off frequency, instead, is reported to be equal to 13.5MHz and 10MHz in [17, 18] respectively, for [16] bandwidth of 40MHz is reported. As can be seen in table 6.3, the proposed structure's area, and power efficiency makes it an excellent choice for portable devices where these two criteria matter the most.

6.6 Conclusion

A current-mode mixed-signal implementation of a distributed arithmetic-based FIR filter is proposed in this chapter. The proposed structure is utilized to implement a band-pass and a low-pass filter to prove the tunability. Sixteen-tap 8-bit filters are realized at different sampling frequencies in $0.18\mu m$ CMOS technology, and magnitude and phase responses are achieved considering the process variations parameters. Avoiding current to voltage converters, adders, and dividers results in a low-power area-efficient structure that is an excellent choice for portable devices. Sampling frequencies for the BPF and the LPF are 10MHz and 48KHz correspondingly. The area and the maximum power consumption of the proposed structure are $0.071mm^2$ and 2.2mW respectively that are significantly lower compared to that of the similar works considering tap numbers, input data number of bits, and sampling frequency.

References

- [1] S. Zohar, "New Hardware Realizations of Non-recursive Digital Filters," *IEEE Trans. Comput.*, vol. C-22, no. 4, pp. 328-338, Apr. 1973.
- [2] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, no. 6, pp. 456-462, Dec. 1974.
- [3] P. Sirisuk, A. Worapishet, S. Chanyavilas, and K. Dejhan, "Implementation of switched-current FIR filter using distributed arithmetic technique: Exploitation of digital concept in analogue domain," *IEEE International Symposium on Communication Technologies*, vol. 1, pp. 143-148, Oct. 2004.
- [4] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, LMS adaptive filters uisng distributed arithmetic for high throughput, *IEEE Trans. Circuits Syst. I*, Reg. Papers, vol. 52, no. 7, pp. 13271337, Jul. 2005.
- [5] S. N. Merchant and B. V. Rao, Distributed arithmetic architecture for image coding, in *Proc. IEEE Int. Conf. TENCON89*, pp. 7477, Nov. 1989.
- [6] S. A. White, Applications of distributed arithmetic to digital signal processing: A tutorial review, *IEEE Trans. Acoust, Speech, Signal Process.*, vol. 37, no. 1, pp. 419, Jan. 1989.
- [7] E. zalevli, W. Huang, P. E. Hasler, and D. E. Anderson, "A Reconfigurable Mixed-Signal VLSI Implementation of Distributed Arithmetic Used for Finite-Impulse Response Filtering," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 2, pp. 510-521, Mar. 2008.
- [8] P. K. Sharma, M. T. Khan, and S. R. Ahamed, "An alternative approach to design reconfigurable mixed signal VLSI DA based FIR filter," *In IEEE Technology Symposium* (*TechSym*), pp. 284-288, Sep. 2016.
- [9] F. A. Farag, C. Galup-Montoro, and M. C. Schneider, "Digitally Programmable Switched-Current FIR Filter for Low-Voltage Applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, pp. 637-641, Apr. 2000.

- [10] A. Worapishet, R. Sitdhikorn, A. Spencer, and J. B. Hughes, "A Multirate Switched-Current Filter Using Class-AB Cascoded Memory," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 11, pp. 1323-1327, Nov. 2006.
- [11] R. Wilcock, B. M. Al-Hashimi, and P. Wilson, "Integrated high bandwidth wave elliptic lowpass switched-current filter in digital CMOS technology," *Electron. Lett.*, vol. 41, no. 5, pp. 222-223, Mar. 2005.
- [12] A. Croisier, D. Esteban, M. Levilion, and V. Riso, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 1973. *IEEE J. Solid-State Circuits*, No. 1, pp. 27-33, Feb. 1988.
- [13] B. Youssefi, M. Mirhassani, J. Wu, "Efficient Mixed-Signal Synapse Multipliers for Multi-Layer Feed-Forward Neural Networks," *IEEE International Midwest Symposium* on Circuits and Systems, pp. 814-817, Oct.2016.
- [14] S. J. Daubert, D. Vallancourt, Y. P. Tsividis, "Current copier cells," *Electron. Lett.*, vol. 24, no. 25, pp. 1560-1562, Dec. 1988.
- [15] G. Wegmann, and E. A. Vittoz. "Basic principles of accurate dynamic current mirrors," *IEE Proceedings G-Circuits, Devices and Systems*, No. 2, pp. 95-100, Apr. 1990
- [16] S. Arvind Rathod, and S. Yellampalli, "Design of Fifth Order Elliptic Filter With Single-Opamp Resonator", *IEEE ICAECC*, pp. 1-6, Oct. 2014.
- [17] M. S. Oskooei, N. Masoumi, M. Kamarei, and H. Sjoland, "A CMOS 4.35-mW +22dBm IIP3 Continuously Tunable Channel Select Filter for WLAN/WiMAX Receivers," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, Jun. 2011.
- [18] S. DAmico, M. Conta, and A. Bashirotto, "A 4.1-mW 10-MHz fourth-order sourcefollower based continuous-time filter with 79-dB DR," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2713-2719, Dec. 2006.

Chapter 7

Conclusions and Future Works

7.1 Summary of Contributions

In this dissertation, it is mainly focused on the mixed-signal design of a fully parallel artificial neural network. First, the advantages of mixed-signal circuit design were explored, then the two different composing building blocks, synapse, and neuron were introduced. Lastly, the generalization ability of the neural networks as a widespread problem in NNs was discussed.

In Chapter 2, we proposed the VLSI implementation of a programmable neuron to address the generalization issue of ANNs. The proposed structure provides different maximal slopes of sigmoid and linear functions. The mentioned various activation functions can be chosen on-chip or off-chip by a 2-bit voltage DAC.

The programmability was achieved by using body effect via controlling the substrate voltage of PMOS transistors. The post-layout simulations, Monte Carlo, and corner analysis were performed to confirm the robustness of the design. To the best of authors knowl-

edge, the proposed architecture is the first analog VLSI implementation that can provide different shapes of activation function post-fabrication.

In Chapter 3, a mixed-signal synaptic multiplier was proposed. The structure works based on the weighted current mirror in combination with AND gates. Using this structure, we addressed the area and power efficiency by avoiding the largest size transistor in the conventional multiplying DACs and cutting out the currents when the operation is not needed. To reduce the mismatch effect in weighted current mode, we designed the structure with two similar building blocks and avoid the size differences between transistors.

In Chapter 4, the proposed synaptic DAC and a current-mode area-efficient neuron were used together as a synapse-neuron building block of a feed-forward 4-3-2 ANN. A series of patterns were successfully recognized with this structure. The area was measured $142299\mu m^2$ which is one of the smallest reported areas of the synaptic multipliers. The average power was measured 0.93mW which is much lower compared to the state of the art designs.

In Chapter 5, the nonlinear dynamic behavior of a single neuron with the sigmoid activation function and a feedback synaptic weight was investigated. We were interested in the possible oscillatory behavior of this structure to be used for neural oscillation applications. The linearization method was used to investigate the dynamic behavior of the structure by linearizing the function around a fixed point to assess the local stability. Different dynamic behaviors of this system which were achieved based on the locus of Z in ROC were investigated.

In Chapter 6, a novel mixed-signal structure of a DA-based FIR filter was proposed. The structure is current-mode and employs DACS with the current-mode outputs to do the multiplication between digital inputs and the analog coefficients. Two 16-tap 8-bit filters were implemented, one is a BPF with the sampling frequency of 10MHz and the other is an LPF with the sampling frequency of 48KHz. The area is a t least 5 times smaller that similar works.

7.2 Suggested Future Work

Multiresolution learning paradigm is an issue which can be further studied to investigate the effect on the generalization ability as well as the learning speed. The learning speed of the neural network discussed in Chapter 4 can be improved by controlling the activation function of each neuron. Also, an improvement in the pattern recognition is expected by using the programmable neuron suggested in Chapter 2.

It is also highly suggested to test the neural network with distributed neuron-synapse blocks by assigning different activation function in each layer. That is possible if we use the adjustable neuron proposed in Chapter 2. In this way, we most probably can address the saturation and overfitting the weights in a higher level compared to non-distributed circuits.

The nonlinear dynamic behavior of a single neuron with the sigmoid activation function which is investigated in Chapter 5 suggests the possibility to be used in oscillator and spiking neuron applications. The suggested structure can be implemented both in analog and digital circuitries. Base on the nature of this work, the implemented oscillator would be robust with a very low jitter. Also, the analog implementation of this structure is highly recommended to realize the spiking neuron. This implementation should most probably is more robust and area-efficient by avoiding the capacitors that has been used so far in spiking neurons implementations.

VITA AUCTORIS

NAME: Bahar Youssefi PLACE OF BIRTH: Tehran, Iran YEAR OF BIRTH: 1984 EDUCATION: University of Isfahan, B.Sc., Isfahan, Iran, 2008 Tarbiat Modares University, M.Sc., Tehran, Iran, 2011 University of Windsor, Ph.D. Windsor, ON, 2018