3-21-2019

Digital Implementation of Bio-Inspired Spiking Neuronal Networks

Shaghayegh Gomar
University of Windsor

Recommended Citation

Gomar, Shaghayegh, "Digital Implementation of Bio-Inspired Spiking Neuronal Networks" (2019). *Electronic Theses and Dissertations*. 7663.
https://scholar.uwindsor.ca/etd/7663

Digital Implementation of Bio-Inspired Spiking Neuronal Networks

by

Shaghayegh Gomar

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
2019

Digital Implementation of Bio-Inspired Spiking Neuronal Networks

By

Shaghayegh (Rose) Gomar

APPROVED BY:

F. Salem, External Examiner
ECE Michigan State University

A. Jaekel,
School of Computer Science

R. Rashidzadeh
Department of Electrical & Computer Engineering

M. Khalid
Department of Electrical & Computer Engineering

M. Ahmadi, Advisor
Department of Electrical & Computer Engineering

March 8, 2019

Declaration of Co-Authorship/Previous Publication

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 Dr. Mitra Mirhassani, Prof. Majid Ahmadi, and Prof. Mehrdad Saif. The design, numerical simulations and experiments were performed by the main author while the contribution of Dr. Mitra Mirhassani was editing the manuscript, Dr. Ahmadi provided the supervision and Dr. Seif provided the final feedbacks to improve the manuscript.

Chapter 3, 4 and 5 was co-authored with Prof. Ahmadi. The design, numerical simulations and experiments were performed by the main author while the contribution of Dr. Ahmadi was providing supervision, feedback and editing the manuscripts.

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.

This thesis includes 4 original papers that have been previously published or submitted for publication in peer reviewed journals and conferences, as follows:

Thesis Chapter	Publication Title	Publication status
Chapter 2	Gomar, Shaghayegh, et al. ”A digital neuromorphic circuit for neural-glial interaction.” Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, 2016.	Published
Chapter 3	Gomar, Shaghayegh, and Majid Ahmadi. ”Digital Realization of PSTDP and TSTDP Learning.” 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.	Published
Chapter 4	Gomar, Shaghayegh, and Majid Ahmadi. ”Digital Hardware Implementation of Gaussian WilsonCowan Neocortex Model.” IEEE Transactions on Emerging Topics in Computational Intelligence 3.1 (2019): 24-35.	Accepted
Chapter 5	Gomar, Shaghayegh, and Majid Ahmadi. ”A Digital Central Pattern Generator Based on Hindmarsh-Rose Spiking Neuron Model” IEEE Transactions on Circuit and Systems I	Submitted

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.

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

Spiking Neural Network as the third generation of artificial neural networks offers a promising solution for future computing, prosthesis, robotic and image processing applications. This thesis introduces digital designs and implementations of building blocks of a Spiking Neural Networks (SNNs) including neurons, learning rule, and small networks of neurons in the form of a Central Pattern Generator (CPG) which can be used as a module in control part of a bio-inspired robot. The circuits have been developed using Verilog Hardware Description Language (VHDL) and simulated through Modelsim and compiled and synthesised by Altera Qurtus Prime software for FPGA devices.

Astrocyte as one of the brain cells controls synaptic activity between neurons by providing feedback to neurons. A novel digital hardware is proposed for neuron-synapse-astrocyte network based on the biological Adaptive Exponential (AdEx) neuron and Postnov astrocyte cell model. The network can be used for implementation of large scale spiking neural networks. Synthesis of the designed circuits shows that the designed astrocyte circuit is able to imitate its biological model and regulate the synapse transmission, successfully. In addition, synthesis results confirms that the proposed design uses less than 1% of available resources of a VIRTEX II FPGA which saves up to 4.4% of FPGA resources in comparison to other designs.

Learning rule is an essential part of every neural network including SNN. In an SNN, a special type of learning called Spike Timing Dependent Plasticity (STDP) is used to modify the connection strength between the spiking neurons. A pair-based STDP (PSTDP) works on pairs of spikes while a Triplet-based STDP (TSTDP) works on triplets of spikes to modify the synaptic weights. A low cost, accurate, and configurable digital architectures are proposed for PSTDP and TSTDP learning models. The proposed circuits have been compared with the state of the art methods like Lookup Table (LUT), and Piecewise Linear approximation (PWL). The circuits can be employed in a large-scale SNN implementation due to their compactness and configurability.

Most of the neuron models represented in the literature are introduced to model the behavior of a single neuron. Since there is a large number of neurons in the brain, a population-based model can be helpful in better understanding of the brain functionality, implementing cognitive tasks and studying the brain diseases. Gaussian Wilson-Cowan model as one of the population-based models represents neuronal activity in the neocortex region of the brain. A digital model is proposed for the Gaussian Wilson-Cowan and examined in terms of dynamical and timing behavior. The evaluation indicates that the proposed model is able to generate the dynamical behavior as the original model is capable of. Digital architectures are implemented on an Altera FPGA board. Experimental results show that the proposed circuits take maximum 2% of the resources of a Stratix Altera board. In addition, static timing analysis indicates that the circuits can work in a maximum frequency of 244 MHz.

Central Pattern Generators (CPGs) are neural circuitries which are responsible for controling the locomotion part of the animals by generating rhythmic patterns. A hardware implementation of bio-inspired CPG can be used in robotic and control applications. In this dissertation, a CPG architecture is constructed by coupling Hindmarsh-Rose (HR) neuron modules to generate the anti-phase patterns in the output of the network. Aiming an efficient implementation, a novel digital implementation is proposed for HR neuron module using a combined methodology which decreases the computational complexity suitable for digital hardware implementation. Timing and dynamical analysis verifies that the model is successfully able to reproduce the bifurcation and patterns which are observed in the original model. The proposed architecture of CPG is made of coupled digital HR which is implemented on an FPGA device aiming a compact, flexible and configurable platform for generating periodic rhythmic antiphase patterns. Results indicate that the generated waveforms from the RTL design follows the simulations successfully and the CPG parameters can be adjusted for an adaptive behavior. Static Timing Analysis indicates that the circuit can work in a maximum frequency of 98.43 MHz which is a 20% improvement in comparison with the current works.

Acknowledgments

I wish to express my most sincere gratitude to my supervisor Prof. Majid Ahmadi for his invaluable guidance and constant support throughout the course of this work.

In addition to my advisor, I would like to thank my committee members, Dr. Rashid Rashidzadeh, Dr. Mohammad Khalid and, Dr. Arunita Jaekel, and Dr. Fathi Salem for their constructive comments and feedback.

I would also like to thank my friend Bahar Youssefi and all my colleagues in the ECE department, ASM and RCIM labs.

Finally, my deepest gratitude goes to my loving fiance, Farid Bahiraei and my dearest parents and sisters for their unconditional love, support, and encouragement.

Contents

List of Figures

List of Tables

List of Abbreviations

AdEx	Adaptive Exponential.
ANN	Artificial neural network.
CPG	Central Pattern Generator.
DHR	Digital Hind-marsh rose Neuron.
DST	Dynamical System Theory.
DWC	Digital Wilson-Cowan.
EP	Equilibrium Point.
ExU	Exponential Unit.
FES	Functional Electrical Stimulation.
FIFO	First in-First out.
FPAA	Field Programmable Analog Arrays.
FPGA	Field programmable Gate Array.
GWC	Gaussian Wilson-Cowan.
HH	Hodgkin-Huxley neuron model.
IF	Integrate and Fire neuron model.
Izh	Izhikevich neuron model.
LUT	Look Up Table.
NE	Neuromorphic Engineering.
PSTDP	Pair-based Spike Timing Dependent Plasticity.
PWL	PieceWise Linear.
RMSE	Root Mean Square Error.
SNN	Spiking Neural Network.
SDC	Synapse Design Constraints.

STA	Static Timing Analysis.
STDP	Spike Timing Dependent Plasticity.
TSTDP	Triplet-based Spike Timing Dependent Plasticity.
VHDL	Verilog Hardware Description Language.
VLSI	Very Large Scale Integrated.

Chapter 1

Introduction

In this chapter, a brief overview of Spiking Neural Network (SNN) and a comparison between the conventional artificial neural network and SNN, Neuromorphic Engineering (NE), and digital hardware implementation methodologies for NE is presented. In the last part of this chapter, the outline of the dissertation and contributions are given.

1.1 Artificial Neural Network vs Spiking Neural Network

Artificial Neural Network (ANN) are based on extremely simplified brain dynamics. ANNs have been powerful to solve complicated problems such as pattern recognition, classification, and function estimation tasks. Spiking Neural Network (SNN) as the third generation of ANN is inspired by brain and mimics the brain behavior more closely in comparison with the conventional ANN. One of the differences between ANN and SNN is that SNN takes another element, time, in their functionality [1]. In SNN, information is transferred by sequences of timing events called spike similar to what is observed in the biological counterpart. Since SNN takes time as an element in their functionality and learning process, it is a good candidate to be used to solve complicated time-dependent pattern recognition problems. In addition to time-dependent pattern recognition tasks, SNN can model the central nervous system which is a neurocircuit called Central Pattern Generator (CPG) [2]-[3]. An electronic CPG can be used in a functional electrical stimulation (FES) system to recover the motor function and stimulate the human muscle to help people with disabilities. Furthermore, hardware implementation of SNN can be employed in several applications such as [4]-[5]:

- Neuroprostheses
- Bio-robotic
- Brain-Machine Interface
- Medical studying

- Neuroprostheses
- Bio-robotic
- Brain-Machine Interface
- Medical studying

1.2 SNN Elements

Each spiking neuron is made of three main part: dendrite tree, soma and axon [6]. Dendrites receive the input information from other neurons. The inputs produce electrical transmembrane currents, if the summation of the current is high enough to change the membrane potential to a threshold voltage, a sudden signal, called spike is emitted. The spike generated by the soma is transferred through axon to the other neurons. Where each two spiking

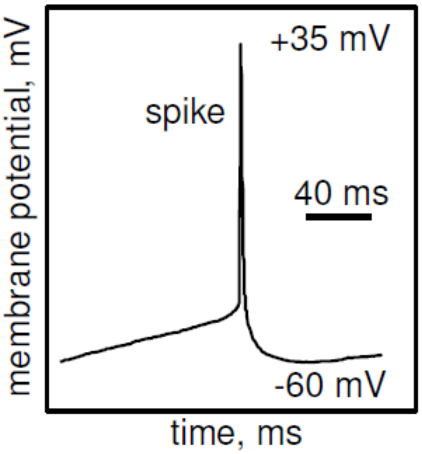


Figure 1.1: A typical spike waveform [6]

neurons are connected is called synapse. The neuron which receives the inputs from other neurons is called postsynaptic neuron, and the neuron which sends the data through axon to the other neurons is called presynaptic neuron. Fig. 1 and Fig. 2 shows a typical spike waveform and a spiking neuron which is connected to another neuron through synapse, respectively [6].

There are also other cells in brain which regulate the activity of the neurons by providing a feedback. Astrocyte is one of the cells which controls the synaptic connection between the neurons. Basically, in a network composed of pre and post synaptic neurons, synapse and astrocyte both neurons and astrocyte will affect each other performance by a feedback which is controlled by a set of parameters [8].

In any neural network, a learning algorithm is required to modify the neurons connection for a specific purpose. Learning is defined as a mathematical rule which modifies the synaptic weights accordingly. The learning algorithm which is used for SNN, is called Spike Timing Dependent (STDP) [8] algorithm which is based on the timing differences between consecutive spikes.

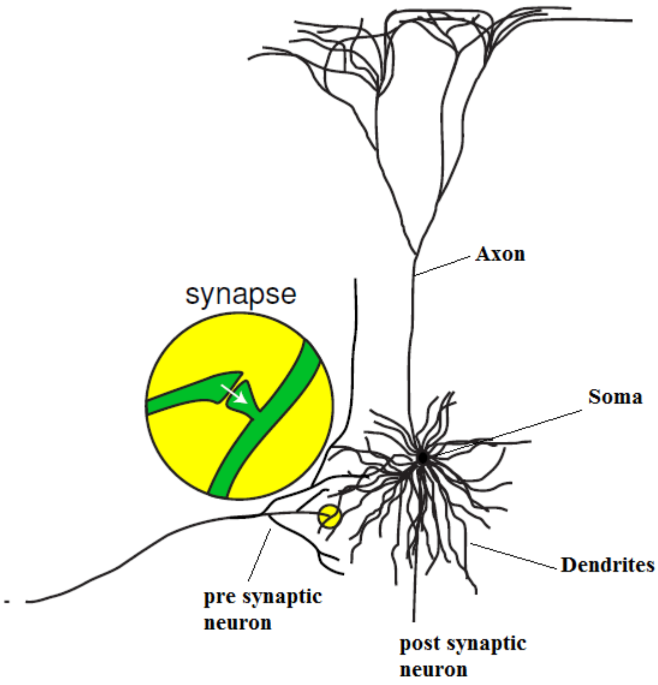


Figure 1.2: A spiking neuron component[6]

1.3 Spiking Neuron Models: Single Models and Population-Based Models

Over the few last decades, scientists have tried to model the behavior of a spiking neuron using mathematical equations to mimic the behavior of the neurons in the brain. A wide range of mathematical models have been introduced to describe the non-linear behavior of a single spiking neuron with different levels of biological accuracy and mathematical complexity. In this range, Hodgkin-Huxely (HH) [9] model is the most accurate model which describes the behavior of a single neuron in a great detail and it won the Nobel prize in 1963. This model is made of four coupled differential equations and it requires a high volume of mathematical calculations. Due to high volume of required computation, this model is not appropriate for hardware implementation. On the other hand, Integrate and Fire (IF) [10] is the simplest spiking neuron model which is made of only one differential equation. Consequently, this model is not able to reproduce a wide range of spiking behavior at the output. However, it is a good candidate for a large scale SNN implementation due to its very low hardware cost. There are other models [11]-[14] which make a trade-off between the biological accuracy and mathematical complexity. Most of these models have been made of two or three coupled differential equations such as Adaptive-Exponential model (AdEx) [15], and Izhikevich (Izh) [16]. There are also other two and three dimensional models which have been presented to simulate the activity of the neurons in brain. These models are fairly ideal candidates for small to medium size SNN implementation such as CPGs and stimulators for medical studying due to the trade-off they provide between the accuracy and mathematical complexity.

In contrast with the single spiking neuron models [9]-[16], population-based models represent the behavior of groups of spiking neurons in the brain. For example, Wilson-Cowan (WC) [17] model presents two coupled differential equations representing the behavior of groups of excitatory and inhibitory neurons in the neocortex part of the brain.

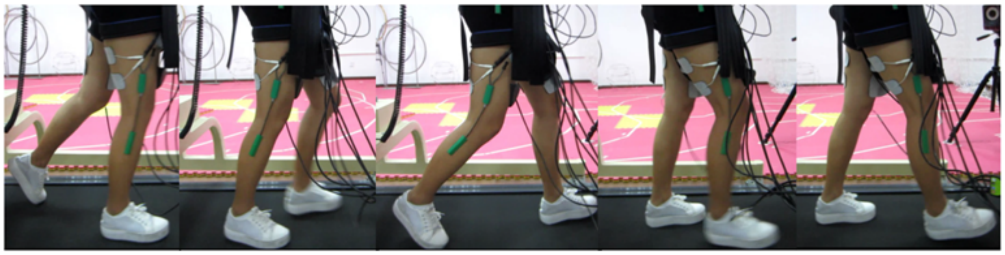


Figure 1.3: A rehabilitation system using CPG [18]

Since, all tasks in our brain are performed by groups of spiking neurons, having a population-based model is essential to study the brain behavior. Neocortex part of the brain is responsible for controlling the locomotion part of the animals including both vertebrate and invertebrates. Therefore, hardware implementation of such a model can be employed in developing bio-robotic and CPGs which can be used in FES. Fig. 3 [18] shows an example of a CPG which has been employed in a rehabilitation system to stimulate muscles for recovering movement.

1.4 Neuromorphic Engineering

Neuromorphic Engineering (NE) is a concept which was introduced by carver Mead [19] in 1980s for the first time, and it states the use of Very Large Scale Integration (VLSI) electronic systems containing analog components to map the functionality of the brain, including the neurons, into analog VLSI. Nowadays, NE is used for digital, analog and mixed signal VLSI implementation of SNN. NE is an interdisciplinary field of study which takes inspiration from many fields including biology, physics, neurology, mathematics, computer science and electronic engineering. There are lots of funded projects around the world which are working in this area including Blue Brain project [20] in EPFL Switzerland and Human Brain [21] project which has over 100 university partner around the world.

1.5 Hardware and Software based Implementations

Different methods can be used for a neural system implementation. Computational cost, speed, and configurability are the main concerns for the implementations. Although CPU-based simulations offer a relatively high speed simulations they are designed to be used for general purpose and every day applications. Besides, they offer a serial implementation which limits the number of neurons which can be implemented at the same time. Hardware implementations instead can provide a platform to have parallel implementations. Although analog implementation is relatively efficient it suffers from the long process of fabrication. FPGAs, on the other hand offer a configurable platform which offers parallel processing which makes it a suitable candidate for spiking neural network implementations.

1.6 Digital Hardware Implementation Methodologies

There are different kinds of methodologies for digital hardware implementation which is discussed briefly in this section. In this thesis, depending on the application, either one or combination of these methods can be used for digital implementation.

- **Look Up Table based implementation (LUT)**: In this method, the discrete points for a range of input are stored in a memory. This method can be very accurate but it is not efficient for large range of inputs and outputs. Furthermore, by adding each bit to the input range the size of the LUT will be doubled.

- **Piecewise Line approximation (PWL)**: In this method, the non-linear behavior of functions is approximated using several first-order lines. The accuracy of the approximation is highly dependent to the number of lines which are used. With the higher number of lines, higher accuracy is provided. This method is more efficient than LUT based approximation, but it usually provides less accuracy.

- **Hybrid method**: A hybrid method can be used for a bivariate function. In this case, by considering one fixed input, PWL method can be employed to approximate the function and these values can be stored in a memory.

- **Base-2 approximation**: In this method, a nonlinear function can be approximated using a base-2 function. Base-2 function can be easily implemented in digital by shift and add/sub operations. Therefore, with an affordable hardware cost, an acceptable accuracy can be provided with this method. Although, this method is efficient, it's more appropriate for non-linear functions with a large input and output range. Because for small input and output range LUT approximation provides better accuracy with an affordable hardware cost.

1.7 Design Flow and Prototype Used in the Thesis

First of all the equations must be discretized for a digital hardware implementation. Choosing an appropriate digital approximation methodology is required for nonlinear functions. The proposed approximation must be tested through a numerical simulation. In this thesis, MATLAB simulation is used to verify the validity of the approximation.

The second step is designing the architecture and employing RTL design which is performed by writing a behavioral Verilog code to describe the model in hardware. The design is simulated by ModelSim software and the functionality is checked in this phase.

In the third phase the design must be synthesized for a target FPGA. This step can be performed using FPGA compilers such as Vivado (Xilinx FPGAs) and Quartus Prime (Altera FPGAs). In this thesis, Quartus Prime compiler has been used to synthesize the design and map the design to the FPGA resources.

The last phase of the design is the measurement which is performed using PXI national instrument device and LabVIEW software. The GPIO port of the Altera FPGA is linked with the input port of the PXI device using a cable and the waveform is captured by

LabVIEW software.

1.8 Outline of the Dissertation and List of the Contributions

In this thesis several digital blocks are designed for SNN.

- In chapter 2, a novel digital neuromorphic design is introduced for neural-glial interaction. The design is made of pre and post synaptic AdEx spiking neuron, synapse, and astrocyte cell model. Synthesis of the designed circuits shows that the designed astrocyte circuit is able to mimic its biological counterpart and modify the synapse connection, successfully. In addition, synthesis results indicate that the proposed design uses less than 1% of available resources of a VIRTEX II FPGA which saves up to 4.4% of FPGA resources in comparison to other designs.

- In chapter 3, a low cost, accurate and configurable digital circuits for pair-based and triplet-based STDP learning models are introduced. The proposed architectures are compared with the state of the art methodologies such as Lookup table (LUT), and Piecewise Linear approximation (PWL). The results show a maximum error of 0.0088 between the proposed digital circuits and the original learning models. The circuits have been synthesized and physically implemented on Altera FPGA board. The Quartus prime synthesis tool verify that the proposed circuits take maximum 1% of the FPGA resources. The circuits can be employed in a large scale Spiking Neural Network (SNN) implementation due to the efficient hardware cost.

- In chapter 4, Gaussian Wilson–Cowan model as a biological model for neocortex part of the brain is investigated in terms of its digital implementation feasibility. Gaussian Wilson–Cowan model as one of the well-known population-based models represents neuronal functionality in neocortex. Hardware implementation of biological neural models are helpful in understanding of the brain performance, cognitive tasks, and learning about the brain diseases. Digital model is proposed for the Gaussian Wilson–Cowan and examined from dynamical and timing behavior point of view. The evaluations show that the proposed model is capable of mimicking the biological model. An efficient digital hardware system is presented for the digitized model with minimum required resources using Verilog Hardware Description Language. Digital architectures are implemented on an Altera FPGA board. Experimental results show that the proposed circuits take maximum 2% of the available resources of a Stratix Altera board. A maximum frequency of 244 MHz has been achieved by employing Static Timing Analysis (STA).

- In chapter 5, A digital architecture is proposed for a CPG composed of coupled Digital Hindmarsh-Rose (DHR) neurons to generate the anti-phase patterns in the output of the network. Hardware implementation of bio-inspired Central Pattern Generators (CPG) is employed in many robotic and control applications. Aiming an efficient implementation, a fast and compact FPGA implementation is proposed for HR neuron by employing on-chip memories and a combined methodology which reduces the computational complexity appropriate for digital hardware implementation. The model is verified in terms of accuracy and system dynamic behavior. The proposed architecture for CPG is implemented on an FPGA device which is suitable in robot control applications. Experimental results indicate that the generated waveforms from the RTL design follows the simulations successfully and the CPG parameters can be adjusted for an adaptive behavior. Static Timing Analysis (STA) indicates that the circuit can work in a maximum frequency of 139.53 MHz which is a 70% improvement in comparison with the fastest work reported in the literature.

References

[1] S. Ghosh-Dastidar, and H. Adeli, “Spiking neural networks,” International journal of neural systems 19, no. 04, pp. 295-308, 2009.

[2] X. Zhang, Z. Xu, C. Henriquez, and S. Ferrari, “Spike-based indirect training of a spiking neural network-controlled virtual insect”, 52nd IEEE Annual Conference on in Decision and Control (CDC), pp. 6798-6805, 2013.

[3] D. Zhang, Q. Zhang, and X. Zhu, “Exploring a type of central pattern generator based on Hindmarsh–Rose model: from theory to application”, International journal of neural systems, vol. 25, no. 01, p. 1450028, 2015.

[4] http://www.ine-web.org/ [Accessed 11 Mar. 2019].

[5] J. Dethier, V. Gilja, P. Nuyujukian, S.A. Elassaad, K.V. Shenoy, and K. Boahen, “Spiking neural network decoder for brain-machine interfaces,” In International IEEE/EMBS Conference on Neural Engineering:[proceedings], 2011.

[6] E. M. Izhikevich, Dynamical systems in neuroscience, MIT press, 2007.

[7] D.E. Postnov, R. N. Koreshkov, N. A. Brazhe, A. R. Brazhe, and O. V. Sosnovtseva, “Dynamical patterns of calcium signaling in a functional model of neuron–astrocyte networks,” Journal of biological physics, vol. 35, no. 4, pp. 425-445, 2009.

[8] S. Song, K. D. Miller, and L. F. Abbott, “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity,” Nature neuroscience, vol. 3, no. 9, p. 919, 2000.

[9] A. L. Hodgkin, and A.F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” The Journal of physiology, vol. 117, no. 4, pp. 500-544, 1952.

[10] E. M. Izhikevich, “Which model to use for cortical spiking neurons?,” IEEE Trans. Neural Networks, vol. 15, no. 5, pp. 1063–1070, 2004.

[11] C. Morris and H. Lecar, “Voltage oscillations in the barnacle giant muscle fiber,” Biophys. J., vol. 35, pp. 193–213, 1981.

[12] R. Jolivet, A. Rauch, H. R. L. Lüscher, and W. Gerstner, “Integrateand-Fire models with adaptation are good enough: Predicting spike times under random current injection,” Adv. Neural Inf. Proces. Syst., pp. 595–602, 2005.

[13] M. G. F. Fuortes and F. Mantegazzini, “Interpretation of the repetitive firing of nerve cells,” J. Gen. Physiol., vol. 45, no. 6, pp. 1163–1179, Jul. 1962.

[14] J. L. Hindmarsh, and R. M. Rose, “A model of neuronal bursting using three coupled first order differential equations,” Proc. R. Soc. Lond. B, vol. 221, no. 1222, pp. 87-102, 1984.

[15] W. Gerstner and R. Brette, “Adaptive exponential integrate-and-fire model,” Scholarpedia, vol. 4, no. 6, p. 8427, 2009.

[16] E. M. Izhikevich, “Which model to use for cortical spiking neurons?,” IEEE Trans. Neural Networks, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.

[17] H. R. Wilson and J. D. Cowan, “Excitatory and inhibitory interactions in localized populations of model neurons,” Biophys. J., vol. 12, no. 1, 645 pp. 1–24, 1972.

[18] D. Zhang, Q. Zhang, and X. Zhu, “Exploring a type of central pattern generator based on Hindmarsh–Rose model: from theory to application,” International journal of neural systems, vol. 25, no. 01, p. 1450028, 2015.

[19] C. Mead, C. “Neuromorphic electronic systems,” Proceedings of the IEEE, vol. 78, no. 10, pp. 1629-1636,1990.

[20] https://bluebrain.epfl.ch/

[21] https://www.humanbrainproject.eu/en/

[22] S. Gomar, M.Ahmadi “Digital Realization of PSTDP and TSTDP Learning” Proc. Of 2018 World Congress on Computational Intelligence, 1-5, July 2018.

Chapter 2

A Digital Neuromorphic Circuit for Neural-Glial Interaction

2.1 Introduction

In the pathway toward understanding the brain activity and its capabilities for learning, glial cell along with neurons is needed to be considered in order to explain brain behavior [1], [2]. Without astrocytes, description of the brain behavior is incomplete. Therefore, in order to have a comprehensive representation of brain, glial cells are required to be incorporated in the modelling of the brain activity.

Astrocyte is a subtype of glial cells and plays an important role in communication between neurons by controlling synaptic transmission through receptors, transporters, and glia transmitters release [3]. In fact, astrocyte reacts to synapse activity and modifies synaptic process [4].

Understanding the functionality of the astrocyte and its relation with neurons affects how the network activities such as information processing, synapse formation, modulation of synaptic transmission, and memory formation are modelled. A precise neural network should include the astrocyte model in its hardware implementation. Accurate implementation of the astrocyte which regulates the synaptic activity improves the reliability of the systems implemented in modern CMOS technologies where fabrication process limits the reliability of circuit operations.

In this paper, an efficient digital implementation of the network composed of neurons, synapse, and astrocyte is presented. This network serves as a building block for implementation of a large scale Spiking Neural Network (SNN).

In the proposed network interaction model between the astrocyte and neuron is implemented. These interactions although have been reviewed in literature [3], [4], but only a few have covered hardware implementation of astrocytes and neural-glial interaction [7]-[9]. It should be mentioned that even in these works, a simplified version of astrocyte interaction is considered.

The proposed digital architecture is based on the Postnov mathematical network [10], except the neuron model has been replaced by the biological AdEx model instead of Fitzhugh-Nagumo (FH) model. The proposed model is more comprehensive and is composed of astrocyte and AdEx neurons. Considering large scale nature of applications to reduce the hardware cost, approximations have been used in the design of the system.

In the following section, the interaction of neurons and astrocyte is briefly presented. The model of the network and its elements are introduced in section 2.3. In section 2.4 the proposed network is introduced. Design and hardware synthesis are discussed in section 2.5, following with the conclusion in section 2.6 .

2.2 Neuron and Astrocyte Interaction

A neural-glial network is composed of at least two neurons which are called presynaptic and postsynaptic neuron, an astrocyte, and a synapse which interacts with the two neurons and the astrocyte. This type of synapse provides three paths to neurons and astrocyte and is called tripartite synapse. In such a network, there is a mechanism in which each unit affects the other cells. This mechanism is initiated by presynaptic neuron. When presynaptic neuron fires, two activations are initiated. First glutamate receptors is activated on the postsynaptic membrane and then glutamate metabotropic receptors is activated on astrocyte which results an increment in the amount of calcium ion (Ca^{+2}). The amount of increment is affected by the level of synaptic activity. Increasing Ca^{+2} results in triggers in the glial transmitters Glutamate release, adenosine triphosphate (ATP), or D-serine. Finally Glial Glu interacts with i-GluRs which results in additional depolarization of the postsynaptic neuron [10].

The astrocyte is controlled by neurons using two routes. Fast route (α parameter) in which astrocyte is sensitive to potassium ion (K^{+}) release when a postsynaptic neuron fires. The second route is slow route (β parameter) in which astrocyte is sensitive to presynaptic neuron firing. On the other hand, astrocyte affects postsynaptic neuron excitation and inhibition as well. The excitation is controlled by γ parameter and inhibition is controlled by δ parameter.

Therefore, Ca^{+2} has oscillations due to Ca^{+2} exchange between neurons and astrocyte cells. This oscillation is transformed into the network and even passes through neighboring astrocytes as well. Experiments indicate that astrocytes can form a network similar to neurons and an astrocyte network can even be connected to another one only through a common astrocyte [11], [12]. Therefore, a large scale network can be built based on the smaller implemented networks of astrocytes and neurons suitable for different processing applications.

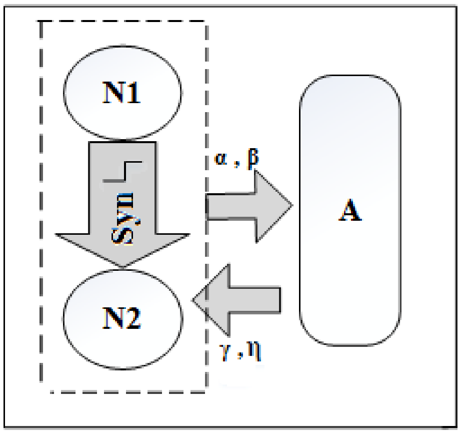


Figure 2.1: A Neuron-Synapse-Astrocyte network: N_1 is presynaptic neuron, N_2 is postsynaptic neuron, S is synapse unit and A is astrocyte unit.

2.3 Network Model

In this section the network model is discussed. The network is composed of two neurons (N_1, N_2), a tripartite synapse (S), and an astrocyte (A) which regulates the synapse transmission between the two neurons. Fig. 2.1 shows the main elements of the network. As it is shown in this figure, both neurons and astrocyte affect each other. The effects of neurons on astrocyte is displayed by α and β parameters and astrocyte controls the postsynaptic neuron through γ and η parameters.

In the following section background information and the mathematical models used for each component are briefly reviewed.

2.3.1 Neuron (N)

There are two neurons in the network, a presynaptic, and a postsynaptic neuron. In this section mathematical representation of their activities is briefly described.

Presynaptic Neuron (N1)

Presynaptic neuron is activated by an external current and if a spike occurs, it will activate the synapse unit.

In this work, neurons are based on the biological AdEx model as follows [11]:

$$C\frac{dV}{dt} = -g_l(V - E_l) + g_l\Delta T \cdot \exp\frac{V - V_T}{\Delta T} + I - \omega \tag{2.1}$$

$$\tau_\omega\frac{d\omega}{dt} = a(V - E_l) - \omega \tag{2.2}$$

where V is the membrane potential, ω is the adaptive variable, C is the total membrane capacitance (pF), g_l is the total leak conductance (nS), E_l is effective rest potential (mV), ΔT is threshold slop factor (mV), V_T is effective threshold potential (mV), V_r is resting potential (mV), τ_ω is adaptation time constant (ms), a is subthreshold adaptation (ns), b is spike trigged adaptation (pA) and I is total current.

When the membrane voltage crosses its apex (0), the voltage and adaptation variables change replaced according to auxiliary (reset) equations, as follows:

$$\text{if } V > 0\; then \quad \begin{cases} V \rightarrow V_r \\ \omega \rightarrow \omega_r = \omega + b \end{cases} \tag{2.3}$$

By choosing different parameters, the model can generate various types of spiking activities [11]. According to the model used in [9] at $I = 500mA$, the tonic neuron can be activated.

In the proposed design, the tonic type is used for both the presynaptic and postsynaptic neurons.

Postsynaptic Neuron (N2)

The same model is used for postsynaptic neuron. However, input current is different for postsynaptic neuron. For presynaptic neuron $I = I_{stimuli}$, while for postsynaptic neuron input current is equal to:

$$I_{post} = (I_2 - I_{syn} - I_{Gm} + I_{Ga}) \tag{2.4}$$

where I_2 is current for N_2, I_{syn} is synapse current, I_{Gm} and I_{Ga} are feedback currents of the astrocyte.

$I_{Gm} = \gamma G_m$ denotes the glutamate flow released by the glial unit and $I_{Ga} = \eta G_a$ is representing hyperpolarizing action of variable G_a. Therefore, when presynaptic neuron spikes, synapse will be activated.

2.3.2 Synapse (S)

There are two main parameters which have to be considered for synaptic transmission: delay and threshold of synapse activation [10].

These two parameters are incorporated in the Kopel model [8] as follows:

$$\tau_c \frac{dz}{dt} = [1 + \tanh(S_s(V_1 - h_s))](1 - z) - \frac{z}{d_s} \tag{2.5}$$

$$I_{syn} = (k_s - \delta G_m)(z - z_0) \tag{2.6}$$

where z is the activation variable, τ_c determines time delay, parameters h_s, S_s, and d_s are constants determining activation and relaxation of z variable, k_s is conductance, and δ is a controller parameter which controls G_m effects. Also, z_0 is a reference level and when synapse is active it is almost 0, otherwise it is 1. When V_1 is less than h_s synapse is inactive, while when it goes further than h_s synpase is activated and z is approximately equal to 1. After each presynaptic neuron spike, V_1 drops below the h_s which inactivates the synapse variable (z). Therefore, when z is active, it provides a current called I_{syn} which is applied to the postsynaptic neuron [10].

2.3.3 Astrocyte Unit (A)

The astrocyte model used in this paper is the model which has been discussed in [10]. This model provides appropriate controlling parameters which can activate or deactivate different paths between the units thus generating specific features of different neural–glial patterns. This model is composed of six equations as follows:

$$\tau_c \frac{dc}{dt} = -c - c_4 \cdot f(c, c_e) + (r + \alpha\omega_2 + \beta S_m) \tag{2.7}$$

$$\varepsilon_c \tau_c \frac{dc_e}{dt} = f(c, c_e) \tag{2.8}$$

$$f(c, c_e) = c_1 \frac{c^2}{1 + c^2} - \frac{c_e^2}{1 + c_e^2} \cdot \frac{c^4}{c_2^4 + c^4} - c_3 c_e \tag{2.9}$$

$$\tau_{Sm} \frac{dS_m}{dt} = [1 + \tanh(s_{Sm}(z - h_{Sm}))](1 - S_m) - \frac{S_m}{dS_m} \tag{2.10}$$

$$\tau_{Gm} \frac{dG_m}{dt} = [1 + \tanh(s_{Gm}(c - h_{Gm}))](1 - G_m) - \frac{G_m}{dG_m} \tag{2.11}$$

$$\tau_{Ga} \frac{dG_a}{dt} = [1 + \tanh(s_{Ga}(c - h_{Ga}))](1 - S_m) - \frac{G_a}{dG_a} \tag{2.12}$$

where variable c represents the calcium dynamics in the astrocyte, c_e presents the calcium concentration in the internal storage, ω_2 denotes the adaptive variable of the second neuron. The nonlinear function f indicates the C_a^{+2} exchange between the cytoplasm and the endoplasmic reticulum. The feedback current from synapse to astrocyte is described by S_m and G_m is responsible for describing the glial glutamate production [10].

In this model, τ_c, τ_s, τ_{Sm} and τ_{Gm} denote time constants delays, and h_S, h_{Sm}, h_{Gm}, and

h_{Ga} are used for discerning of activation and inactivation states of V_1, z, c, $c_1 - c_4$, d_{Sm}, d_{Gm}, d_{Ga}, s_{sm}, s_{Gm}, s_{Ga} also are some other constants which can be provided using the model in [10].

To implement the network in hardware, modifications are required to develop an efficient digital system representing such a network. It should be noted that τ_{Sm} is much larger than τ_c, therefore c_e displays a faster dynamic behavior [8], [9]. However, its dynamic affects the astrocyte outputs therefore, in order to provide more accuracy, implementation of c_e and f is also required.

2.4 Proposed Network

In this section, a modified model of the network is introduced. In the original model, there are two main functions which are expensive from hardware implementation point of view. These functions are Exponential function (Ex) used in the neuron model and the Tanh function (Tanh) which is the main function of synapse and astrocyte as represented by (5), (10)-(12).

2.4.1 Exponential Function

Implementation of the exponential function generally consumes a lot of resources, and takes up a large area. Therefore, to implement this function efficiently, approximations have to be applied.

The first equation of the neuron model can be modified using a base-2 term [14]. A base-2 value can be implemented in digital using shift and add operations.

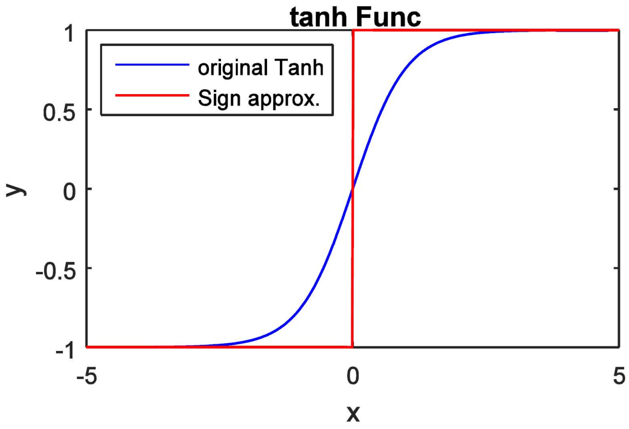


Figure 2.2: Tanh function and its approximations

2.4.2 Tanh Function

The Tanh function can be approximated using a sign function approximation as follows:

$$\tanh(x) \cong \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{2.13}$$

Fig. 2.2 shows the Tanh and its approximation based on (13).

The network has been simulated and the results are shown in Fig. 2.3 and Fig. 2.4.

Fig. 2.3 shows the astrocyte effect on the postsynaptic neuron. As can be seen from this figure, astrocyte is able to control the postsynaptic pattern by tuning the γ and δ parameter. Therefore, even after stopping the stimuli current, the postsynaptic neuron is spiking due to the feedback currents from the astrocyte. The postsynaptic neuron firing continues during the G_m activity.

On the other hand, Fig. 4.4 represents the results for slow and fast activation routes of the astrocyte. The fast activation is controlled by α parameter. Fig. 2.4(a) shows the postsynaptic neuron effect on the calcium release (variable c) for two different values of α parameter (0.0033 and 0.0004). Small values of α results in damped oscillations in the variable c. By increasing α the effect of postsynaptic neuron on astrocyte is strong enough

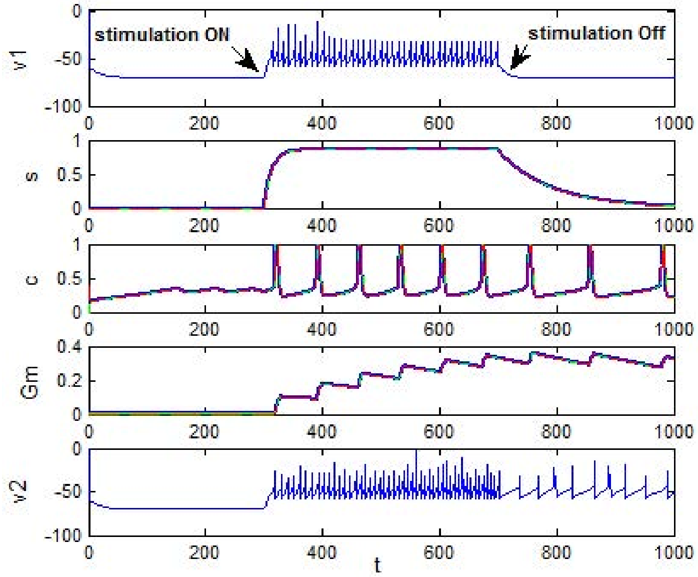


Figure 2.3: Astrocyte effect on the postsynaptic neuron, V_1 is membrane potential of the first neuron. Astrocyte variables have been simulated using three approximations for Tanh function: original (blue), base-2 (green), sign (red). V_2 is representing the second neuron output.

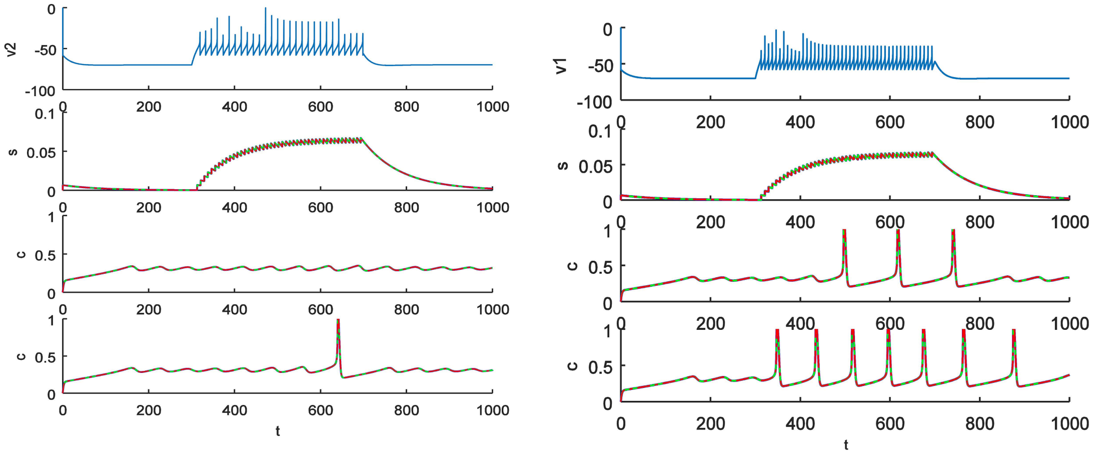


Figure 2.4: Simulation results (a) fast route activation (b) slow route activation

to initiate spike in variable c during the postsynaptic neuron firing.

The slow activation route is controlled by β parameter. Fig. 2.4(b) presents the presynaptic neuron effect on c for two different values of β parameter (0.05 and 0.09). As it

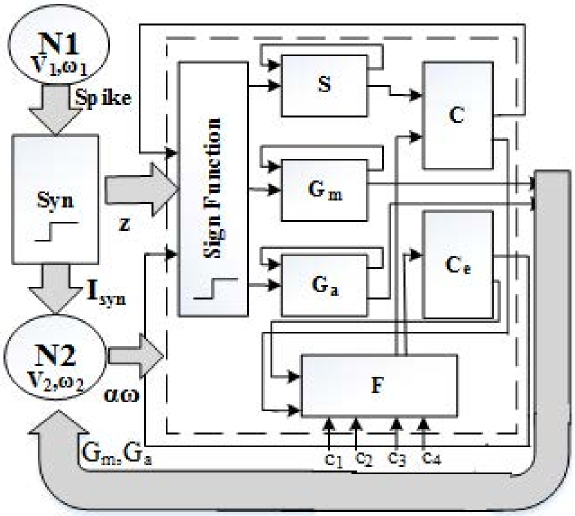


Figure 2.5: Top level architecture of the neural network system

can be seen from this figure, β controls the presynaptic neuron effect on variable c which describes the calcium release. At small values of β the calcium amplitude has some oscillations and it slowly increases during the presynaptic firing period. When the presynaptic neuron stops spiking, there is still some fluctuations in Ca^{+2} release and then it gradually disappears. At higher values of β, more activity is observed in Ca^{+2} fluctuations which is in match with biological experiments.

In these simulations, base-2 has been used for exponential function and (13) has been used for Tanh and the results are close to the case in which original tanh function has been used.

The results of approximation output are very similar to original one. According to these results, sign function (red line) is able to produce original Tanh (green line). Therefore it is an appropriate option for digital implementation due to its low-cost and accuracy.

2.5 Digital Design and Synthesis

This section presents the hardware architecture of the system. As the first step, it is required to discretize (1) and (2) for the neurons, (5) and (6) for the synapse , and (7)-(12) for the astrocyte is required. This step is performed using the Euler method.

The second step is digital optimization and bit width determination. Fixed point representation has been used for defining a variable with fraction part. Variables range calculation is needed to determine the bit width of the variables. Since the integer range of the astrocyte variables is small, two bits can be used to represent integer part of each variable. However, to avoid overflow when the digits are shifted, 5 bits has been considered for the integer part of the variables. The fraction part digit number determines the precision of the computation. For the fraction part, 25 digits has been considered which provides a precision of 2.98×10^{-8} for digital calculations.

Base-2 model has been used for the exponential function implementation of the neurons and sign function has been used instead of Tanh function to have an efficient implementation.

Fig. 2.5 shows the top level view of the system architecture. In this network architecture, the presynaptic neuron, N_1 is initially stimulated by input current. The synapse unit is activated by each presynaptic neuron spike. When V_1 is in under threshold, z variable of synapse unit is inactive. By increasing V_1 above threshold variable z is activated. Moreover, c is activated through slow activation route which is controlled by β parameter. The synapse unit sends a current signal to postsynaptic neuron and also activates the block of the astrocyte unit which is described by Eq. (10). Then N_2 is activated by synapse which affects the astrocyte by α parameter (fast route activation). Finally astrocyte provides the feedback currents to postsynaptic neuron which are controlled by γ and δ parameter.

Scheduling details of the astrocyte unit and synapse have been shown in Fig. 2.6 and Fig. 2.7, respectively. In order to reduce hardware cost, constant multiplication and division are implemented using add and shift. Three dividers are required for implementation of the

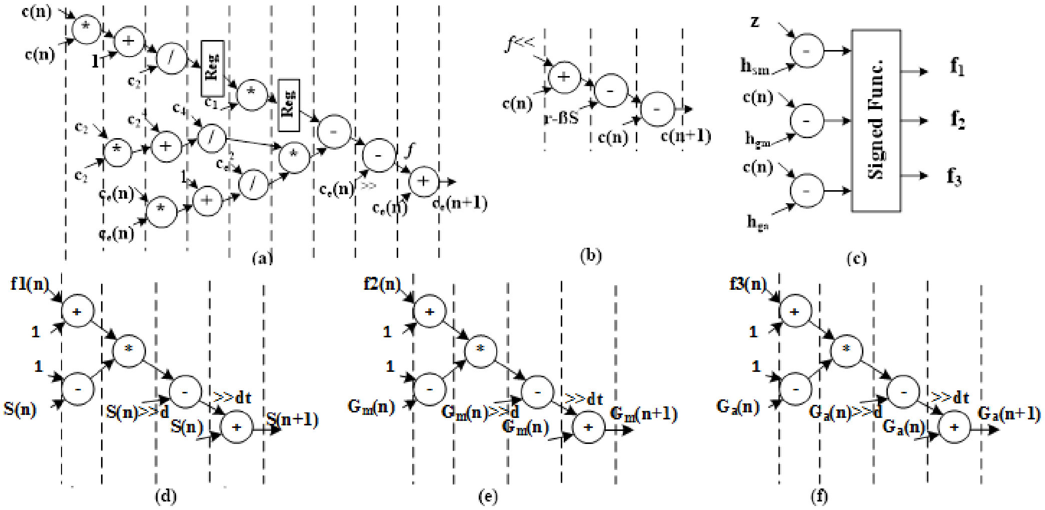


Figure 2.6: Scheduling diagram for astrocyte unit: a) f and c_e computation, b) c variable of astrocyte computation c) Sign function block d) Variable S computation e) Variable G_m computation and f) variable G_a computation

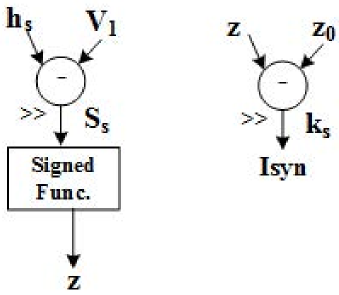


Figure 2.7: Synapse unit computation

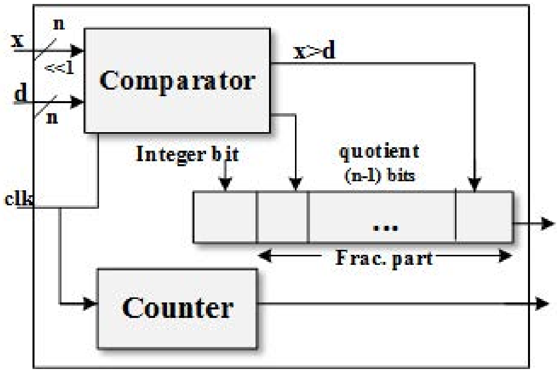


Figure 2.8: Divider building block

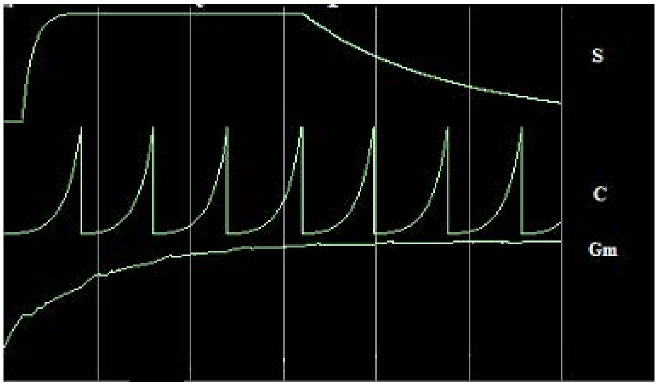


Figure 2.9: Output results of the astrocyte

block which is described by Eq. (9). However, only two multiplication/division are used in each cycle of the scheduling design. In the diagrams, dash lines separate different cycles. For instance, based on Fig. 4.6(a), nine cycles are needed to have c_e variable ready.

The synapse unit has been presented in Fig. 5.6. The synapse has been implemented by a sign function. In this scheduling, shift and add have also been used for constant multiplication or division. If presynaptic neuron spikes, the input of sign function is positive which activates the z and synapse sends I_{syn} to the postsynaptic neuron, otherwise z is inactive and I_{syn} is zero.

Since the results of the division terms are positive and less than 1, therefore it definitely meets the sequential division condition [15]. Fig. 2.8 displays the divider block. In this block, x, the dividend and d, the divisor, are inputs of the division block. With each clock pulse, x is shifted to right and is compared with the dividend and the result would be a bit of quotient. After $(n-1)$ clock pulses the output is ready.

Fig. 2.9 displays a typical output of the proposed digital architecture of astrocyte. In this figure, c presents the calcium release of the astrocyte which is activated by the presynaptic and postsynaptic neurons, successfully.

The hardware design has been synthesized using ISE tools for a VIRTEX II FPGA. Table. I shows a comparison between this work and previous works from network feature and hardware cost points of view. The proposed network is different from the previous works

Table 2.1: Network feature and hardware cost comparison

	Neuron Type	No. Neurons	f and c_e eq.	No. FF	LUT	Mult/Div
[7]	Izh	1	included	923	1349	1(20*20)
[9]	FH	4	not included	315	233	2
proposed work	AdEx	2	included	205	143	2

in the neuron model and network complexity point of view. A neuron model with more similarity with biological neuron (AdEx) has been used in the proposed network. In pervious works, the dynamical function of astrocyte model which describes the Ca^{+2} exchange either has not been considered or a linear approximation has been used. However, because its dynamic affects the astrocyte output, the nonlinear function has been considered and implemented using sequential division which can be implemented in hardware efficiently. Implementation of nonlinear function provides a more accurate functionality of astrocyte and neuron interaction. The proposed system utilizes less than 1% of the FPGA resources (27392 LUT and FF) that saves up to 4.4% of FPGA resources when compared with similar works. This feature allows for scaling the proposed network for implementation of large scale spiking networks.

2.6 Conclusion

A network of two neurons and an astrocyte has been developed based on Postnov astrocyte and AdEx neuron model. The equations have been modified for an efficient hardware implementation. Simulations have shown that Tanh function can be approximated using sign function with acceptable accuracy and similarity with the original function, therefore sign function has been used for equation approximation. Although, c_e dynamic is faster it can affect the output, so to have more accuracy f and c_e equations also have been incorporated in the design. The proposed architecture has been simulated and synthesized and the results show that it is successfully able to reproduce the interaction signals. A digital architecture has been presented for the network. In order to reduce the hardware cost, shift and add have been used instead of multiplication and division in some cases where constant multiplication or division is needed. Synthesis results verifies that the proposed design uses less than 1% of available resources of a VIRTEX II FPGA, so it can be used as a module in a large scale SNN implementation. On the other hand, since astrocyte is able to regulate the synaptic weight, it can increase the systems reliability in neural network applications.

Acknowledgement

We would like to acknowledge CMC Microsystems for the provision of products and services that facilitated this research for providing CAD tools.

References

[1] L. E. Clarke, B.A. Barres, “Emerging roles of astrocytes in neural circuit development.” Nature Reviews Neuroscience 14.5 (2013): 311-321.

[2] A. M. Benediktsson, G. S. Marrs, J. C. Tu, P. F. Worley, P. F., J. D. Rothstein, D. E. Bergles, M. E. Dailey, “Neuronal activity regulates glutamate transporter dynamics in developing astrocytes,” Glia, vol. 60, pp. 175-188, 2012.

[3] M. Papa, C. De Luca ,F. Petta, L. Alberghina, G. Cirillo , “Astrocyte–neuron interplay in maladaptive plasticity” *Neuroscience and Biobehavioral Reviews*, vol. 42, pp. 35-54, 2014.

[4] S. Nazari, K. Faez, M. Amiri, E. Karami, “A digital implementation of neuron–astrocyte interaction for neuromorphic applications,” *Neural Networks*, vol. 66, pp. 79-90, 2015.

[5] M. López-Hidalgo, J. Schummers, “Cortical maps: a role for astrocytes,” *Current opinion in neurobiology*, vol. 24, pp. 176-189, 2014.

[6] A. Volterra, and J. Meldolesi, “Astrocytes, from brain glue to communication elements: the revolution continues,” *Nature Reviews Neuroscience*, vol. 6, no. 8, pp. 626-640, 2005.

[7] Y. Irizarry-Valle, A. Parker, “An astrocyte neuromorphic circuit that influences neuronal phase synchrony,” *IEEE Trans on what*, vol.9, pp. 175–187, 2015.

[8] H. Soleimani, M. Bavandpour, A. Ahmadi, D. Abbott, “Digital Implementation of a Biological Astrocyte Model and Its Application.” *Neural Networks and Learning Systems, IEEE Trans.*, Vol. 26, pp.

[9] M. Hayati, M. Nouri, M. Haghiri and D. Abbott, “A Digital Realization of Astrocyte and Neural Glial Interactions,”*Trans. on Biomedical Circuits and Systems*, in press(Researchgate).

[10] D. E. Postnov, R. N. Koreshkov, N. A. Brazhe, A. R. Brazhe, O. V. Sosnovtseva, "Dynamical patterns of calcium signaling in a functional model of neuron–astrocyte networks," Journal of biological physics vol. 35, pp. 425-445, 2009.

[11] D. E. PosDmitry, L. S. Ryazanova, and V. Sosnovtseva. "Functional modeling of neural–glial interaction." BioSystems 89.1 (2007): 84-91.

[12] L. Corentin, A. Fournel and E. Reynaud, "A neurons-astrocyte network model: from synaptic boosting to epilepsy." Cinquième conférence plénière française de Neurosciences Computationnelles," Neurocomp'10", 2010.

[13] R. Naud, N. Marcille, C. Clopath and W. gerstner, "Firing patterns in the adaptive exponential integrate-and-fire model,"*Biol. Cybern.,*vol. 99, no. 4-5, pp. 335-347, 2008.

[14] S. Gomar, A. Ahmadi, "Digital Multiplierless Implementation of Biological Adaptive-Exponential Neuron Model, "*Trans. on Circuit and Sytem*, vol. 61, pp. 1204-1219, 2014.

[15] Koren, Israel. Computer arithmetic algorithms. Universities Press, 2002.

Chapter 3

Digital Realization of PSTDP and TSTDP Learning

3.1 Introduction

Synaptic plasticity plays an important role in learning and memory process in the brain [1]. Biological experiments shows that the relative timing of the spike events is very important to form a phenomena in the brain called Spike Timing Dependent Plasticity (STDP) learning. STDP is a process which is responsible for strengthening and weakening the connection between pairs of neurons in the brain [2]. The classical model of STDP which updates the weights only based on the timing difference between pairs of pre-post neuron spikes is called pair-based STDP (PSTDP). PSTDP is not able to generate some of the biological observations in higher-order spike trains [3]. Triplet based STDP (TSTDP) which was introduced by Pfister and Gerstner in 2006 [1], on the other hand, is able to take the effect of triplets of spikes into account [2], and it is able to reproduce the biological counterpart behavior with higher accuracy.

In this paper, a couple of efficient digital architectures are proposed for PSTDP and TSTDP learning models which are able to replicate the biological models, accurately. The proposed digital models have been employed for a coupled modular spiking neurons and it is indicated that the proposed learning circuit is able to successfully adjust the post-synaptic neuron firing pattern. Furthermore, the proposed circuits are compared with the state of the art solutions for implementing mathematical learning models, and the results confirm that the proposed architectures are more efficient and accurate. Digital Hardware Description Language (HDL) has been employed to design the circuits and the design has been physically implemented on Altera FPGA board. The experimental results verify that the digital circuits are successfully able to reproduce the learning curves for both PSTDP and TSTDP models and modify the postsynaptic neuron firing pattern.

The rest of this paper has been organized as below. In section 3.2, a brief is presented on learning rules: PSTDP and TSTDP. In section 3.3, the proposed models are introduced and compared with the current solutions. Comparison and error measurement are given in section 3.4. In section 3.5, digital hardware design is discussed. Experimental results are presented in section 3.6, and the paper is concluded in section 3.7.

3.2 Material And Method

Learning is defined as synaptic weight updating by a rule. This process has been modeled by different types of mathematical equations. Among them, PSTDP and TSTDP highly match with the biological process which are briefly discussed in the next two subsections.

3.2.1 Pair Based STDP Rule

PSTDP which is based on a pair of spikes timing difference is defined as [2]:

$$\Delta w = \begin{cases} \Delta w^{+} = A^{+} e^{\frac{-\Delta t}{\tau^{+}}}, & \text{if } \Delta t \geq 0 \\ \Delta w^{-} = -A^{-} e^{\frac{\Delta t}{\tau^{-}}}, & \text{if } \Delta t \leq 0 \end{cases} \tag{3.1}$$

Where $\Delta t = t_{post} - t_{pre}$, represents the time difference between the pre and postsynaptic neurons spikes. In this model, Δt^{+} denotes potentiation and Δw^{-} denotes depression. Based on this model potentiation occurs when postsynaptic neuron fires after presynaptic neuron ($\Delta t \geq 0$) and depression occurs when the postsynaptic neuron fires before presynaptic one ($\Delta t < 0$).

3.2.2 Triplet Based STDP Rule

TSTDP modifies the weights of synapses based on three consecutive spikes timing [2]-[3]. TSTDP equations are given as below

$$\Delta w = \begin{cases} \Delta w^{+} = e^{\frac{-\Delta t_1}{\tau^{+}}} (A_2^{+} + A_3^{+} e^{\frac{-\Delta t_2}{\tau_y}}) \\ \Delta w^{-} = -e^{\frac{\Delta t_1}{\tau^{-}}} (A_2^{-} + A_3^{-} e^{\frac{-\Delta t_3}{\tau_x}}) \end{cases} \tag{3.2}$$

similar to PSTDP, Δw^{+} and Δw^{-} represents potentiation and depression, respectively. Potentiation occurs when the postsynaptic spike arrives while depression occurs by a presynaptic spike arrival. A_2^{+}, A_3^{+}, A_2^{-}, A_3^{-} are the potentiation and depression strength parameters and τ^{+}, τ^{-}, τ_x and τ_y are time constants which control the width of STDP curve window. Depending on the neuron type, and timescale, the width of the learning curve can be adjusted. Also, $\Delta t_i s$ are defined as below:

- Δt_1: time difference between pre and postsynaptic neuron spikes
- Δt_2: time difference between current post and previous postsynaptic neuron spikes

- Δt_3: time difference between current pre and previous presynaptic neuron spikes

Fig. 3.1 demonstrates the timing representations for PSTDP and TSTDP learning protocols.

3.3 State of the Art and Proposed Model

3.3.1 LUT Based Approximation

Discrete points are stored in a memory in LUT method [5]. The accuracy is highly dependent on the number of the points which is chosen for the hardware implementation. In fact, by adding each bit accuracy, the size of the memory will be doubled. Moreover, the LUT-based methods require a high volume of memory to store the values which are not efficient when targeting a large scale system implementation or a function with a wide range of input and output.

3.3.2 PWL Approximation

In PWL model, the nonlinear behavior of the equations is linearized by several first order lines. This method is efficient but its accuracy depends on the number of lines used in approximation [4],[6]-[7]. In this paper, the proposed model is compared with LUT method, and linear approximation by using one line (PWL), and two lines (PWL2) for approximating each curve.

PSTDP Rule

PWL: Each exponential term can be approximated by one line:

$$\begin{cases} \Delta w^{+} = m_0 \Delta t + b_0 \\ \Delta w^{-} = m_1 \Delta t + b_1 \end{cases} \tag{3.3}$$

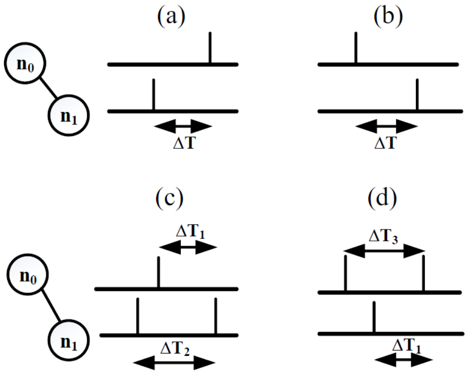


Figure 3.1: PSTDP, and TSTDP timing representations. n_0 represents the presynaptic neuron, and n_1 represents the postsynaptic neuron. a) PSTDP depression, b) PSTDP potentiation, c) TSTDP, post-pre-post protocol, and d) TSTDP, pre-post-pre protocol. Vertical bars show the spike events. [2]

PWL2: Each exponential term can be approximated by 2 lines:

$$\Delta w^{+} = \begin{cases} m_0 \Delta t + b_0, & \text{if } \Delta t \geq p_0 \\ m_1 \Delta t + b_1, & \text{if } \Delta t \leq p_1 \end{cases} \tag{3.4}$$

$$\Delta w^{-} = \begin{cases} m_2 \Delta t + b_2, & \text{if } \Delta t \geq p_3 \\ m_3 \Delta t + b_3, & \text{if } \Delta t \leq p_4 \end{cases} \tag{3.5}$$

where m_i, and b_i represent the parameters of the lines.

Table 3.1 shows the line coefficients for PWL and PWL2 models.

TSTDP Rule

Since TSTDP is a bivariate function of $dt1$ and $dt2$, a combination of PWL and LUT methods can be used as a hybrid method for TSTDP implementation. TSTDP equations can be converted into a function with one variable ($dt2$). Therefore, one-variable functions can be approximated by lines and stored in memory for discrete values of $dt1$. The details about the size of the memory are discussed in the following section.

Table 3.1: PSTDP coefficients for PWL

parameters	m0	m1	b0	b1
$A^{+} = 1, A^{-} = 1$	-0.016	-0.016	1	-1
$A^{+} = 0.5, A^{-} = 0.5$	-0.008	-0.008	0.5	-0.5
$A^{+} = 1, A^{-} = 1$	-2^{-6}	-2^{-6}	1	-1
$A^{+} = 0.5, A^{-} = 0.5$	-2^{-7}	-2^{-7}	0.5	-0.5

Table 3.2: PSTDP coefficients for PWL2

parameters	m0	m1	m2	m3	b0	b1	b2	b3
$A^{+} = 1, A^{-} = 1$	-0.0224	-0.0096	-0.0224	-0.0096	1	0.68	-1	-0.68
$A^{+} = 0.5, A^{-} = 0.5$	-0.0112	-0.0048	-0.0112	-0.0048	0.5	0.34	-0.5	-0.34
$A^{+} = 1, A^{-} = 1$	-0.0234	-0.0098	-0.0234	-.0098	1	-0.6875	-1	-0.6875
$A^{+} = 0.5, A^{-} = 0.5$	-0.0117	-0.0049	-0.0117	-0.0049	0.5	0.34375	-0.5	-0.34375

Table 3.3: Error measurement for PSTDP models for time step =1. Values in the brackets shows the data bit-width.

Err	PWL[8]	PWL[16]	PWL2[8]	PWL2[16]	LUT[8]
RMSE	0.0121	0.0121	0.0042	0.0042	$2.86 * 10^{-4}$
ME	0.2846	0.2769	0.0703	0.0726	0.0078
Err	LUT[16]	$2^{1.5x}$[8]	$2^{1.4375x}$[8]	$2^{1.5x}$[16]	$2^{1.4375x}$[16]
RMSE	$2.35 * 10^{-6}$	$5.85 * 10^{-4}$	$2.83 * 10^{-4}$	$4.78 * 10^{-4}$	$4.53 * 10^{-5}$
ME	10^{-4}	0.021	0.0088	0.0143	0.0014

Table 3.4: Error measurement for TSTDP models for time step=1. Values in the brackets shows the data bit-width.

Err	hybrid [16]	$2^{1.5x}$[8]	$2^{1.5x}$[16]	$2^{1.4375x}$[8]	$2^{1.4375x}$[16]
RMSE	$4.19 * 10^{-5}$	$2.03 * 10^{-5}$	$1.30 * 10^{-6}$	$2.03 * 10^{-5}$	$1.81 * 10^{-7}$
ME	$3.2 * 10^{-3}$	$9.3 * 10^{-3}$	$1.7 * 10^{-3}$	$7.8 * 10^{-3}$	$1.73 * 10^{-4}$

3.3.3 Base-2 Approximation

STDP equations are made from exponential functions. Although LUT method can work accurately and linear approximations are efficient for implementing the exponential function, the accuracy and hardware cost are the main concerns simultaneously when a large-scale implementation is targeted. The exponential function can be converted to a base-2 function (Ex2) which has been employed in [4] for the first time:

$$exp(x) \cong 2^{1.44x} \cong 2^{1.5x} \tag{3.6}$$

In [4], the coefficient 1.44, has been modified to 1.5 to make the digital implementation straightforward. In order to increase the accuracy the approximation can be modified as:

$$exp(x) \cong 2^{x+2^{-1}x-2^{-4}x} \cong 2^{1.4375x} \tag{3.7}$$

This modification only requires one more addition operation, however, because of the exponential nature of the function it improves the accuracy, significantly. As shown in Table 3.2 and 3.3 the accuracy of equation (7) is up to 10 times of the (6).

$PSTDP\ approximation$: By applying (7) to PSTDP model, approximated equations are extracted as below:

$PSTDP_{Ex2}$:

$$\Delta w = \begin{cases} \Delta w^{+} = A^{+}.2^{-1.4375(\frac{\Delta t}{\tau^{+}})}, & \text{if } \Delta t \geq 0 \\ \Delta w^{-} = -A^{-}.2^{1.4375(\frac{\Delta t}{\tau^{-}})}, & \text{if } \Delta t \leq 0 \end{cases} \tag{3.8}$$

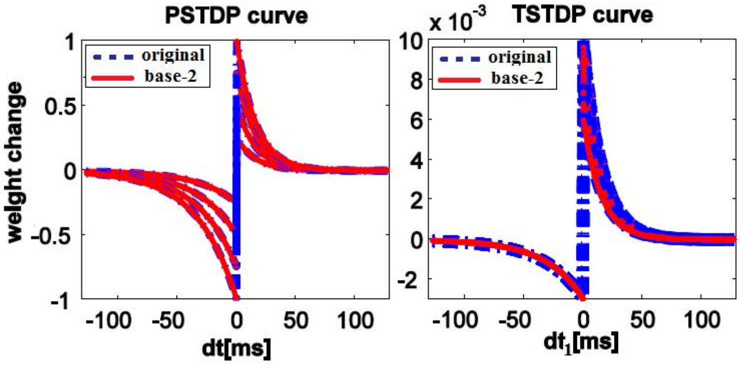


Figure 3.2: (a) PSTDP, (b) TSTDP curves

$TSTDP\ approximation$: Similar to PSTDP, the equations can be extracted for TSTDP model as below:

$TSTDP_{Ex2}$:

$$\Delta w = \begin{cases} \Delta w^{+} = A_2^{+}.2^{-1.4375\frac{\Delta t_1}{\tau^{+}}} + A_3^{+}.2^{-1.4375(\frac{\Delta t_2}{\tau_y}+\frac{\Delta t_1}{\tau^{+}})} \\ \Delta w^{-} = -A_2^{-}.2^{1.4375\frac{\Delta t_1}{\tau^{-}}} + A_3^{-}.2^{-1.4375(\frac{\Delta t_3}{\tau_x}-\frac{\Delta t_1}{\tau^{-}})} \end{cases} \tag{3.9}$$

Fig. 3.2 shows PSTDP, and TSTDP curves and the base-2 models for different values of parameters. As shown, PSTDP and TSTDP curves have been reproduced by proposed models accurately. In the next section, the proposed model is compared with the original model in terms of accuracy.

3.4 Comparison and Error Measurement

In order to compare the original function and the results two criteria have been used to evaluate the circuits accuracy which are introduced as below:

- $Root\ Mean\ Square\ Error\ (RMSE)$: which is defined as:

$$RMSE = \sqrt{\frac{\Sigma(func_{origin} - func_{approx})^2}{n}} \tag{3.10}$$

where, $func$ represents either TSTDP or PSTDP functions, and n represents the number of discrete points which have been used in the design.

- $Max\ Error$ (ME): The max absolute difference between the original function and approximated models has been considered as a criteria to evaluate and compare the accuracy of the models.

Table II and III show the error measurement and a comparison between the proposed models and the original function simulation with two different bit-width of 8 and 16 bits. For PSTDP learning, 8 bits have been considered for Δt (one more bit is required for the sign bit) therefore a range of (-127,127) can be provided to the input of the PSTDP unit. The errors have been measured with time-step equal to $1ms$. It should be noted for a smaller time step equal to $0.5ms$, one additional bit has to be considered for the bit-width of the learning input (Δt). This additional bit does not increase the hardware cost of the PWL and base-2 models, significantly, however, a double-sized memory is required for the LUT approach. For the TSTDP learning, a range of $(-127, 127)$ and $(0 - 64)$ have been considered for Δt_1 and Δt_2 or Δt_3, respectively. Based on the neuron models used in the network, parameters can be adjusted for the network and neuron models. For error measurement, parameter values in [2] have been used.

Based on these results, the LUT approach has the highest accuracy, but it requires a larger amount of hardware resources. It should also be noted that by adding each bit to the inputs the hardware cost is doubled which is not efficient in a large-scale neural networks implementation. PWL method require the least hardware cost but it suffers from lower accuracy. Among these models, the base-2 model provides accuracy with affordable hardware cost. The accuracy has also been compared based on bit-width for the models. As it is expected by increasing the bit-width higher accuracy is provided by digital models.

The parameters can be adjusted any time in the base-2 and PWL models without modifying the whole design, while this is impossible when LUT approach is employed for hardware implementation. When parameters are modified, the stored values in the memory needs to be changed based on the new parameters for LUT method. When input range changes the characteristics of the lines require to be recalculated and modified, in the base-2 method, learning parameters can be modified directly. Therefore, base-2 model provides the highest reconfigurability, and it is going to be discussed for the rest of the paper.

3.5 Digital Hardware Design

The top level design is shown in Fig. 3.3(a), and Fig. 3.3(b) shows the Finite State Machine (FSM) diagram designed for the system. The system is made of three main parts: neuron unit, time measurement unit (Timer), and Learning Unit (LU). In the FSM diagram, S_0 represents the neuron calculation state, S_1 represents the timing measurement state, depending on the results of state S_1, either potentiation, S_2, or depression, S_3 is activated, and then the process is repeated.

3.5.1 Neuron Unit

The neuron unit is made of two neurons: pre and postsynaptic neurons. PLE [4] model which is a digital version of Adaptive Exponential neuron model [8] has been used in the system. Synaptic weight between pre and the postsynaptic neuron is modified using PSTDP and TSTDP rule. Spike trains are passed to timing measurement unit.

3.5.2 Timer

This unit consists of two counters which count the number of clocks between each pair of pre-post (or post-pre) spikes. If one of the neurons fires, it initiates the counter. The counter continues counting until it receives $spike$ signal from the other neuron. Once it receives the $spike$ signal, it stops counting and resets the counter. Depending on the sign of the counter output either potentiation or depression will be activated. If the time difference is positive potentiation occurs. If the time difference is negative depression occurs. Based on the sign of the output, the related parameters (A, τ) for potentiation and depression are chosen using two multiplexers, and then the results are given to the learning unit to compute either potentiation or depression weight change.

3.5.3 Learning Unit: Base-2 Model

Scheduling diagram

Fig. 4.4 shows the scheduling diagram designs of the circuits for PSTDP, TSTDP, and modified Ex block. PSTDP and TSTDP circuit implementations are given in Fig. 3.4 a and c, respectively. As shown, the first step for both learning is timing measurement. Since TSTDP has three spike trains as inputs, it's timing measurement is more complicated than the timing measurement of PSTDP. This block measures two timing difference of Δt_1 and Δt_2 for the exponential blocks and then based on the sign of Δt_1, the required parameters are chosen from the register bank and then the results are passed to the Ex blocks. Two Ex blocks are required to compute potentiation and depression based on (6). Finally, Ex blocks results are added together to compute either Δw^+ or Δw^-.

Exponential unit

The Ex block has been shown in Fig.3.4(b). Since the inputs are always negative, checking the sign bit is not needed. Therefore, a modified Ex block has been designed for negative numbers which reduces the hardware cost to half of the original Ex block reported in [4]. After preparing the input, calculating $1.4375X$, integer and fraction parts are stored in a register. The integer part of the input is checked to find 'true' bits. At the same time, the fraction part is added to $'1'$. Based on the 'true' bits, the result of the adder is shifted to the right.

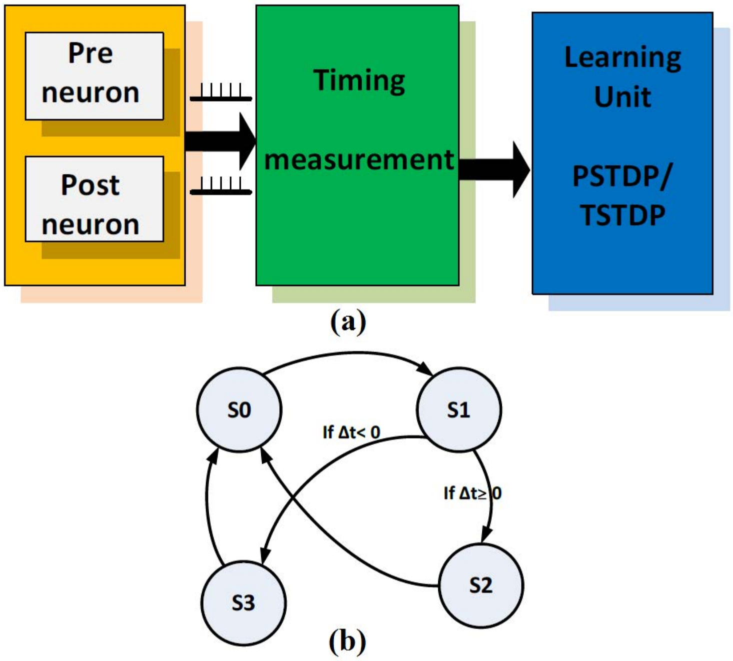


Figure 3.3: (a) Top level design of the system, (b) Finite State Machine (FSM) designed for the system. neuron membrane calculation is performed in state S_0. S_1 represents the timing measurement state. Depending on the sign of the timing measurement result, either S_2 (potentiation), or S_3 (depression) will occur.

Bit-width Determination

The minimum and maximum range of the variables are required to be examined to choose the optimum bit-width range. To reduce the hardware cost, varied bit-width data path has been considered for the registers ranging from 8 to 16 bits. Since the weight change range is small (less than 1), accuracy and consequently fraction part is more important. Higher number of fraction bits provides more precision. For example, with the choice of 10 bits for the bit-width of the fraction part of the Ex block an accuracy about 0.001 (2^{-10} equals to $9.7656 * 10^{-4}$) is provided in the output.

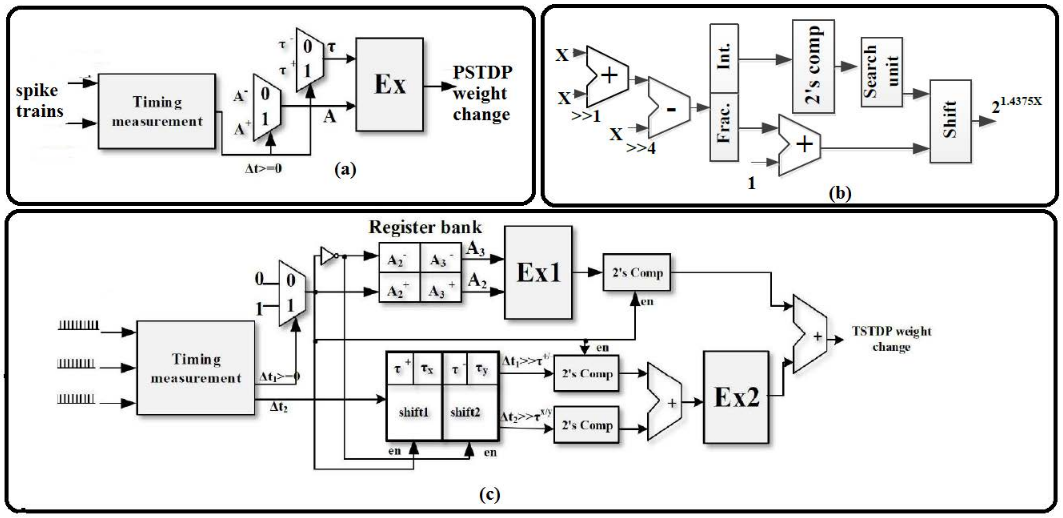


Figure 3.4: Digital circuits: (a) PSTDP implementation, (b) Ex Unit [4], and (c) TSTDP implementation

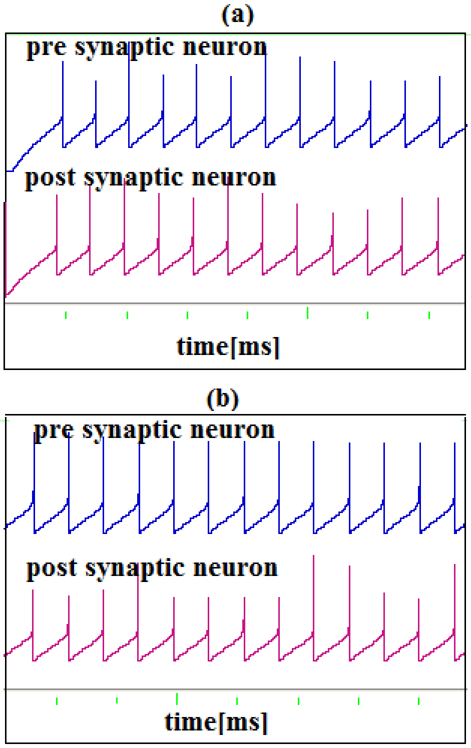


Figure 3.5: PSTDP demonstration for a pair of pre and postsynaptic neurons

Table 3.5: Digital synthesis results

model	Total reg	ALUT	IO pins	Max Freq (MHz)
PSTDP	16	49	26	250
TSTDP	12	125	26	250

3.6 Experimental Results

The circuits were developed in verilog Hardware Description Language (vHDL) and they have been physically implemented on FPGA. System setup is made of three parts: hardware, data transfer, and user interface. Circuits are implemented on an Altera Stratix VI FPGA (EP4SGX530KH40C2N) board. The results of hardware utilization are given in Table 3.3. Fig. 3.5 shows a pair of pre and postsynaptic neurons which are coupled and PSTDP rule has been applied for regulating the strength between the two neurons. Fig. 3.5(a) shows the neurons spike patterns in the first 100 ms, and Fig. 3.5(b) shows the result when PSTDP is completed and depression causes both neurons to fire, simultaneously.

3.7 Conclusion

Digital hardware circuits have been proposed and compared for STDP learning rules. The modified base-2 method with an efficient hardware provides a tradeoff between accuracy and configurability for the system. The maximum error was reported 0.0088 for 8 bit data-path and 0.0014 for 16 bit data-path. In addition, the hardware cost is not highly dependent to the data-path bit-width. Digital circuits have been synthesized and implemented on FPGA, and the experimental results showed that the models are capable of producing the learning window, successfully.

The models were compared with the LUT and linear approximations. Although LUT model provides the highest accuracy the hardware requirement is high and highly dependent on the accuracy and it does not provide configurability. PWL method has lower hardware cost, but it provides the least accuracy. A modular pair of pre-postsynaptic neuron have been implemented and STDP could successfully be applied to the coupled neurons. The hardware cost for the proposed learning model architectures was obtained less than 1% of the FPGA resources, and the maximum frequency was gained as $250MHz$. The model developed in this work can be employed as a module for a larger spiking neural network.

References

[1] J. Pfister and W. Gerstner,“Triplets of spikes in a model of spike timing dependent plasticity” *The Journal of Neuroscience*, vol. 26, no. 38, pp.9673–9682, 2006.

[2] M. R. Azghadi, et al. “Efficient design of triplet based spike-timing dependent plasticity,”*IEEE International Joint Conference on Neural Networks (IJCNN)*, 2012.

[3] M. R. Azghadi, et al. “Novel VLSI implementation for triplet-based spike-timing dependent plasticity,” *Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2011.

[4] S. Gomar, and A. Ahmadi, “Digital multiplierless implementation of biological adaptive-exponential neuron model,” *I*EEE Transactions on Circuits and Systems I: Regular Papers vol. 61, no 4, 2014.

[5] K. Leboeuf, A. H. Namin, R. Muscedere, H. Wu, and M. Ahmadi, “High-speed VLSI implementation of the hyperbolic tangent sigmoid function,”*Third IEEE International Conference on In Convergence and Hybrid Information Technology, ICCIT'08*, Vol. 1, pp. 1070-1073 , 2008.

[6] H. Soleimani, A. Ahmadi, and M. Bavandpour, “Biologically inspired spiking neurons: Piecewise linear models and digital implementation,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 12, 2012.

[7] E. Jokar, and H. Soleimani, “Digital Multiplierless Realization of a Calcium-Based Plasticity Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, 2017.

[8] R. Naud, et al,“Firing patterns in the adaptive exponential integrate-and-fire model,” *Biological cybernetics 99*, no .4-5, 2008.

Chapter 4

Digital Hardware Implementation of Gaussian Wilson–Cowan Neocortex Model

4.1 Introduction

Brain is made of different regions, each region is made of a large number of neuron cells. Although in the recent decades a wide range of mathematical models [1]-[7] and electronic circuits implementations, either in digital or analog [8]-[13], have been developed for describing a single neuron behavior but since the number of neurons and synapses even in a small region of the brain is enormous, a model which is able to represent the activity of a population of neurons is more suitable to represent neuron activities of different regions of the brain [14]. The first attempts go back to 1950 and 1963 when Beurle [15] and later Griffith [16] who introduced a model by focusing on a population of neurons which are activated at a specific time frame in a given volume of model brain tissue. However, their model was unable to describe inhibitory neurons and also it was not able to describe excitatory neurons with refractory or recovery variable [14]. After two decades, Wilson and Cowan [17] modified the Beurle's [15] model to include both excitatory and inhibitory neurons and also refractories in the form of a set of coupled differential equations in which each represents one population activity. Beurle and Wilson & Cowan are considered as the pioneers who introduced a population-based mathematical model for the neocortex region. Furthermore, this work was followed by Amari [18] to study pattern formation in continuum models of neural activity.

The activation function used in Wilson-Cowan for representing the nonlinear behavior of the input and output of the neural populations is a sigmoid function which is not in complete match with experimental observations [15]. Recently, a model has been introduced [19] which is more thorough. In this model the sigmoid function has been replaced by a Gaussian function which is able to show the upper threshold phenomenon, and it is more realistic as there is experimental evidence which supports the Gaussian activation function better than the sigmoid one [19]. In fact, both vivo and vitro observations indicate that the relationship between the input and output of the network is not like a sigmoid function, but also it supports more accurate activation function such as a Gaussian curve instead of the sigmoid function [19].

Cognitive features of brain can be inspired for designing future computing machines, and this is possible by mapping the brain functionality into an electrical hardware system which is the subject of a newborn interdisciplinary field of study named Neuromorphic Engineering (NE) [20]. Hardware implementation of the neural models not only can help to imitate brain features for different kinds of engineering applications [21] but also it can help studying biological neural network functionality and brain diseases in order to find the appropriate treatments for the brain diseases. Although a wide range of works have been presented in the literature for a single neuron implementation [8]-[13], literature reviews

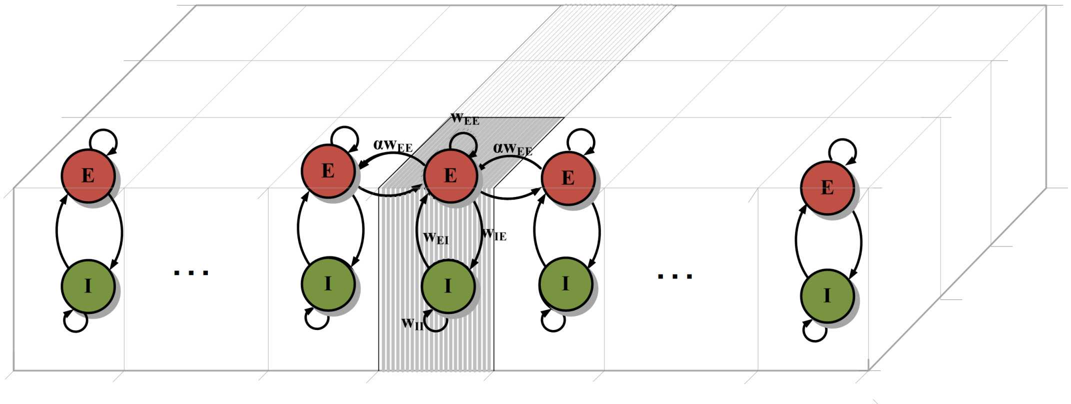


Figure 4.1: Cortical sheet schematic. Each column is represented by a pair of Excitatory (E) and Inhibitory (I) neuron population [19]

lack a hardware implementation for a popular-based model such as Wilson- Cowan Model.

When a hardware implementation is targeted, computational cost, speed, biological accuracy, and configurability are the main concerns. Different approaches can be employed for hardware implementation of such a network including digital/analog ASIC design and Field Programmable Gate Arrays (FPGAs) [9]-[10]. Although analog implementation is more efficient in terms of power and area, the process of its design and fabrication is very time-consuming. In addition, analog circuits are very sensitive to thermal noise or device mismatch [9]- [10]. FPGA, on the other hand provides a configurable platform for digital implementation of the biological neural network.

In this paper, hardware implementation of Gaussian Wilson-Cowan model is investigated. The digital model is examined in terms of accuracy, and dynamical behavior. The analysis demonstrates that the proposed model is successfully able to reproduce the original model patterns and dynamical bifurcations. In the next step, a low-cost digital hardware design is presented and implemented on an Altera FPGA board. The hardware is designed with the minimum required resources to keep the design suitable for large-scale implementation. Timing analysis is also performed to obtain the maximum possible frequency for the proposed design.

The rest of the paper is organized as follows. In section 4.2, Gaussian Wilson-Cowan is briefly presented. In section 4.3, the digital model is proposed and compared in terms of dynamic and timing analysis. Digital circuit design is presented in Section 4.4. Experimental results and conclusion are given in sections 4.5 and 4.6, respectively.

4.2 Gaussian Wilson-Cowan Model

Local microcircuits in the neocortex can be modelled by an excitatory and an inhibitory populations with weights for the connection strengths [19] which was presented for the first time by Wilson & Cowan (WC) [17]. These populations can be coupled to the neighboring pairs through long range excitatory connections projecting to the excitatory population. Such a configuration can be demonstrated in Fig. 4.1. Studies indicate that Gaussian function is more reflective of biological evidence in comparison with a sigmoid function [17], [19], [22]. The later modified model was introduced by Meijer et al, [19] as a Gaussian Firing Rate Function (Gaussian FRF). The model can be given as:

$$\tau_X X_k' = -X_k + (1 - X_k)F_x(J_{X_k}) \tag{4.1}$$

$$F_X(J_{X_k}) = exp(-(\frac{J_{X_k} - X_\theta}{X_{sd}})^2) - exp(-(\frac{-X_\theta}{X_{sd}})^2) \tag{4.2}$$

$$J_{E_k} = w_{EE}E_k - w_{IE}I_k + B + \alpha w_{EE}(E_{k+1} + E_{k-1}) \tag{4.3}$$

$$J_{I_k} = w_{EI}E_k - w_{II}I_k \tag{4.4}$$

where, in these equations X is either E or I, in which E represents the excitatory neuron population while I represents the inhibitory neuron population. K is the index of the populations, $K = 1, 2, \ldots, N$. The excitatory populations of E_1, and E_N only receive input from E_2 and E_{N-1} , respectively [19]. τ_E and τ_I are the time constants of (1) which are taken equal to $1ms$ in all our simulations and implementations. According to [19], the output patterns of the model can be controlled mainly by two bifurcation parameters of B, and w_{EI} . The rest of the parameters are given in Table 4.1. These parameters have been

Table 4.1: Equation parameters

parametr	value	parameter	value
w_{EE}	16	E_θ	7
w_{IE}	12	I_θ	5
w_{II}	3	E_{sd}	2.1
τ_x	1	I_{sd}	1.5

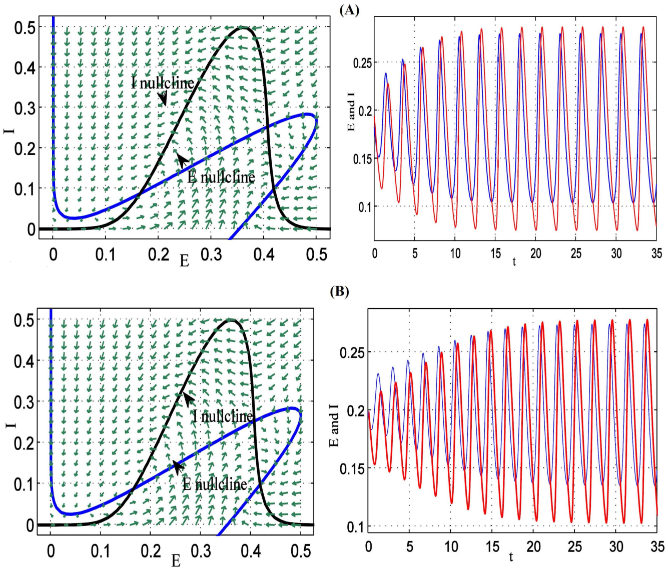


Figure 4.2: Phase plane and timing plots for (A) original GWC model, and (B) DWC model. Green arrows show the direction for the system, the E and I nullclines have been shown in blue and black in the phase plane graphs, respectively. In the timing plots, variable E has been demonstrated in blue, and I has been indicated in red. In both models, $B = 3$, and $w_{EI} = 18$, while $E_{sd} = 2$, $I_{sd} = 1.6$ for DWC, and $E_{sd} = 2.1$, and $I_{sd} = 1.5$ for original model, respectively.

used in the former studies as well [19].

Table 4.2: Bifurcation parameters

Parameter	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19
B	0.25	3	8	1	10	2.65	2.7	2.7	2.25	2.5	2	1	7.75	7.5	5.5	3.8	6.7	5	3.5
w_{EI}	12	30	8	2	27	23	21	16	14	11	10	8	25	20	20	19	16	16	13

4.3 Digitized Gaussian Wilson-Cowan Model

In this section, different approaches for digital implementation of GWC model are discussed, and a digital model is proposed. Equations of the model are highly nonlinear which makes the implementation challenging. However, some approximations can be taken without a significant variation in output pattern in comparison with the original model. The main challenge is implementing (2) in which the first term is a combination of exponential and squaring function which can be approximated using techniques proposed in [9], and [10]. Based on the proposed method introduced in [9], which has been used in several works such as [13] the exponential function can be approximated to a digitally implementable function as below:

$$exp(x) \cong 2^{1.5x} \tag{4.5}$$

The range of the outputs of GWC model (i.e E, and I) is limited to $(0-1)$, which means these equations are very sensitive to the accuracy of the calculations. Therefore, in order to increase the accuracy of the digital model, a modified, and more accurate approximation is proposed in this paper:

$$exp(x) \cong 2^{x} + 2^{-x} + 2^{-4x} \cong 2^{1.4375x} \tag{4.6}$$

Another option for a digital implementation of an exponential block is using Look Up Tables (LUTs) [23]. Although LUTs provide enough accuracy but they usually require a large volume of memory which is not efficient especially in large scale implementation. More details are given about this approach in the digital implementation section.

Secondly, based on (1) and (2), two multipliers are required for implementing each population (E, I). One for the second term of the (1), $(X_k F_X(J_(X_k)))$, and another one for square function for the first term of (2), $(((J_{(X_k)} - X_\theta)/X_{sd})^2)$.

Since the multipliers are very expensive in terms of hardware implementation cost, Piecewise Linear approximation (PWL) technique as a well-known method [11] in digital hardware design can be utilized for the square function which results in reducing one multiplier with the cost of reducing the accuracy. Although, PWL method can reduce the number of multipliers, but it is unable to provide required accuracy in most cases, because the output of the square function is the input of the exponential function which results in increasing the error.

Furthermore, the second term of (2) can be calculated using the values given in Table. I:

$$exp(-(\frac{-E_\theta}{E_{sd}})^2) \cong exp(-(\frac{-7}{2.1})^2) = 1.4979 * 10^{-5} \tag{4.7}$$

$$exp(-(\frac{-I_\theta}{I_{sd}})^2) \cong exp(-(\frac{-5}{1.5})^2) = 1.4979 * 10^{-5} \tag{4.8}$$

As can be seen from the above calculations the second term can be ignored since its value is very small. It is shown later on in simulations that this does not affect the outputs of the equations, significantly. The rest of the calculations can be performed using simple mathematic operations with a small coefficient modifications which are discussed in the digital circuit design section.

Accordingly, Digital Gaussian Wilson-Cowan (DWC) model can be introduced as below:

$$\tau_X X_k^{'} = -X_k + (1 - X_k)F_x(J_{X_k}) \tag{4.9}$$

$$F_X(J_{X_k}) = 2^{K(-G_X)} \tag{4.10}$$

$$G_X = (\frac{J_{X_k - X_\theta}}{X_{sd}})^2 \tag{4.11}$$

where $K = 1.4375$, and $J(E_k)$, and $J(I_k)$ are the same as the original model. Function G_X can be implemented either with a multiplier for a higher accuracy, or using a PWL method for an efficient implementation. Because the square function is the input for the

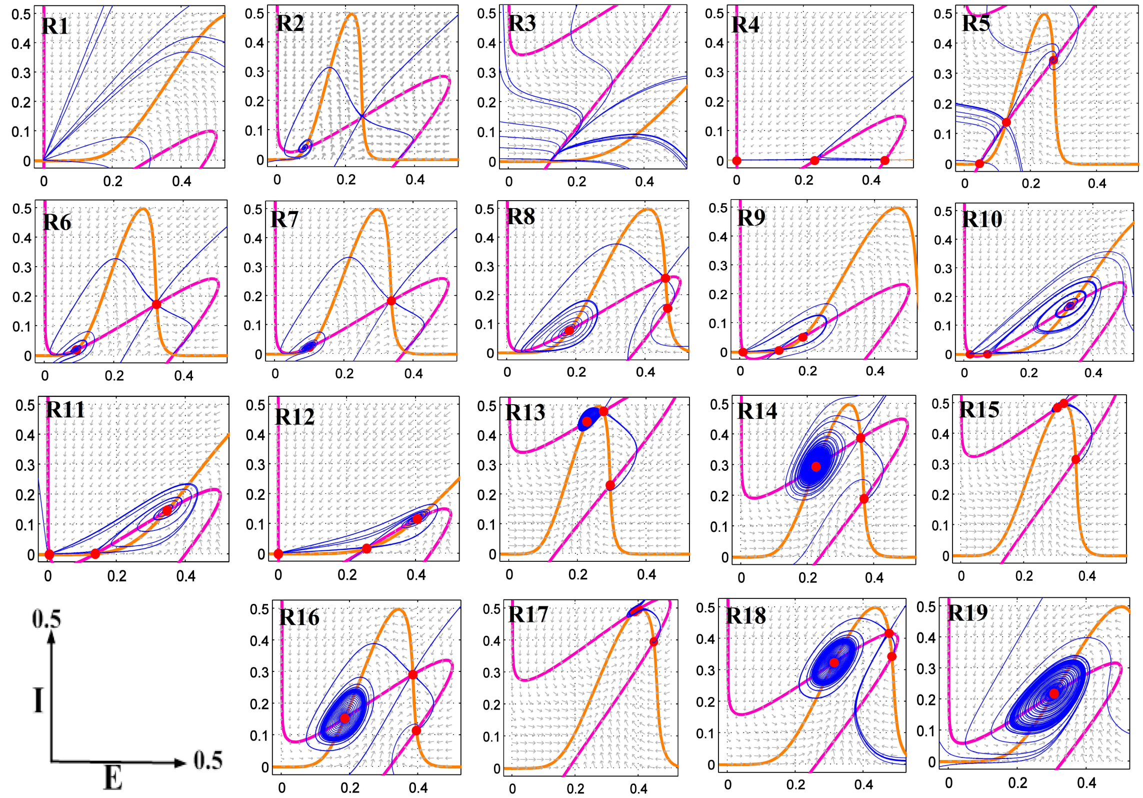


Figure 4.3: Bifurcation analysis for original GWC model with varying the control parameters: B and wEI. Orange and pink color show the I and E null cline respectively. The arrows and the blue lines show the direction of the system dynamic. Red dots are the equilibrium of the system

exponential function, it is important to have an accurate implementation for the square function, otherwise the error raises exponentially when the output of the square function is passed into the exponential block. Therefore, DWC with 4 multiplier, DWC-4M, is focused in the next sections.

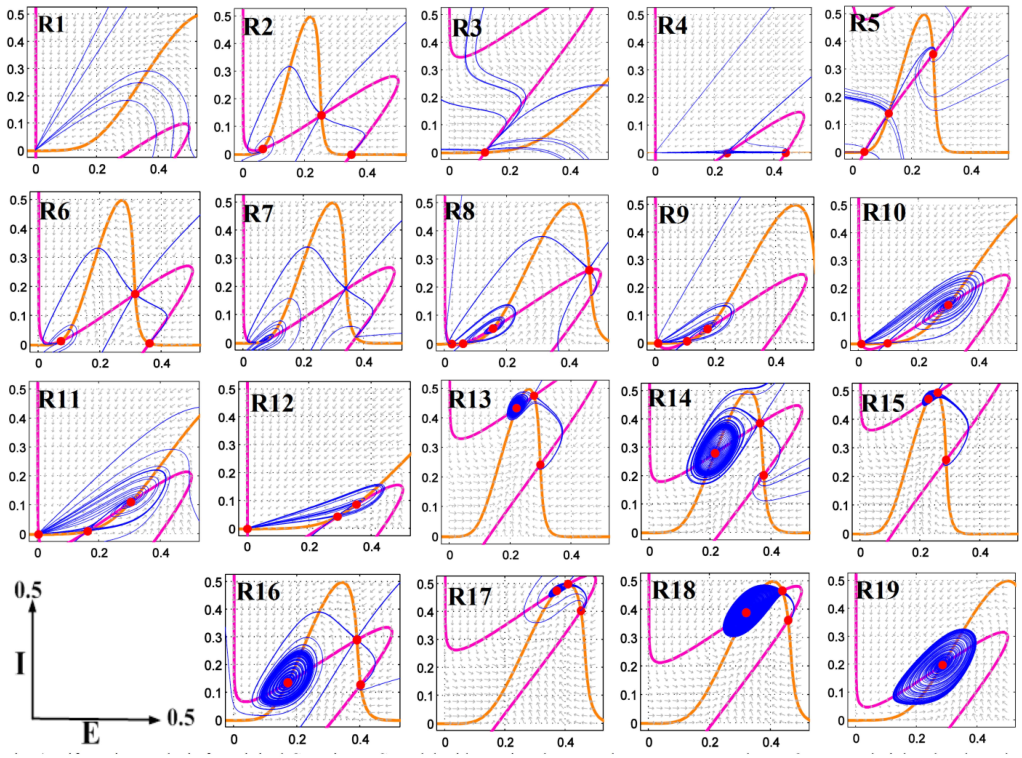


Figure 4.4: Bifurcation analysis for DWC model with varying the control parameters: B and w_{EI}. Orange and pink color show the I and E null cline respectively. The arrows and the blue lines show the direction of the system dynamic. Red dots are the equilibrium of the system.

4.4 Evaluation and Comparison: Dynamical and Timing Analysis

In this section, the original and proposed models are compared in terms of dynamical and timing analysis.

4.4.1 Dynamic Analysis

As it was mentioned in the previous section, GWC model supports biological experiments, more than sigmoid based WC model. Only one intersection is observed in the phase plane of the sigmoid based WC model [20], while the Gaussian model has two more intersections (saddle, and stable node) which results in two more steady states. Therefore, Gaussian model is more powerful in comparison to WC model in terms of the potential of generating more possible dynamics in the system.

In this section, GWC and DWC models are compared in terms of dynamical behavior. Fig. 4.2 shows a typical phase plane and timing plot with a set of parameters for original model (Fig. 4.2(A)) and DWC model (Fig. 4.2(B)). E_{sd} and I_{sd} parameters have been slightly modified for the DWC model. As can be seen in Fig. 4.2 the digital model phase plane and timing plot are very similar to the original one. As shown in Fig. 4.2(B), DWC model also has three intersections in its phase plane which means it is able to follow the original model dynamic, successfully.

The dynamic of the system is mainly controlled by two bifurcation parameters: B, and w_{EI}. In fact, by varying these two parameters, different types of bifurcations and oscillations are shown by the system [20]. For a thorough investigation of the system the phase portrait and timing plots have been obtained for different values of B and w_{EI} . Fig. 4.3, and 4.4 show different shapes of the system dynamic for GWC and DWC models, with varying control parameters (B and w_{EI}). Also the parameters used for each region have been given in Table. 4.2.

To have a dynamic analysis for the models, the null-cline of the equations can be obtained as below:

$$\frac{dE}{dt} = 0 \Rightarrow F_E(E, I) = 0 \tag{4.12}$$

$$\frac{dI}{dt} = 0 \Rightarrow F_I(E, I) = 0 \tag{4.13}$$

The intersections of the null-clines are called equilibrium points. The equilibrium points have been calculated using MATLAB. For a bifurcation analysis, Jacobean Matrix (JM) and eigenvalues are required to be calculated as below:

$$JM = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{4.14}$$

where

$$\begin{aligned} A &= \frac{\partial F_E(E,I)}{\partial E}, B = \frac{\partial F_E(E,I)}{\partial I} \\ C &= \frac{\partial F_I(E,I)}{\partial E}, D = \frac{\partial F_I(E,I)}{\partial I} \end{aligned} \tag{4.15}$$

Jacobean coefficients for original model (A, B, C, D), and DWC model (A', B', C', D') have been given in the appendix section.

In the next step, the eigenvalues can be calculated using the following equations:

$$\begin{aligned} T &= A + B \\ Z &= AD - BC \\ P(\lambda) &= \lambda^2 - T\lambda + Z \end{aligned} \tag{4.16}$$

The roots of $P(\lambda)$ are defined as the eigenvalues of the system. Four types of fixed points can be observed in the system: nodal sink, spiral sink, saddle point, and spiral source. Nodal sink fixed point has eigenvalues with negative real part. In spiral sink fixed point, eigenvalues are positive real numbers. In saddle fixed point, corresponding eigenvalues are real with different signs $\lambda_1 > 0, \lambda_2 < 0$ ($R2$, $R4$-$R18$). Spiral sink and spiral source both have complex eigenvalues with negative and positive real part, respectively. The type of fixed points of the system have been recognized for each region by calculating Jacobean matrix and eigenvalues which have been shown in Table. III. The DWC model is able to reproduce the same types of fixed points and dynamic which have been reported in [19] for GWC model. This table also shows how many fixed points each region has. For example, $R7$ has 4 fixed points, two nodal sink, one spiral sink and one saddle point. In addition, each point's coordinators are given in the table as well. For the same example, the saddle point is located at $(E, I) = (0.34, 0.19)$.

Table 4.3: Fixed points types of different regions for DWC model

R	Nodal sink	Spiral Sink	Spiral source	Saddle point
R1	(0,0)	-	-	-
R2	(0.35,0)	(0.06,0.02)	-	(0.25,0.14)
R3	(0.12, 0.0016)	-	-	-
R4	(0,0) (0.44, 0.001)	- -	- -	(0.24,0.0003)
R5	(0.39,0)	(0.27,0.350	-	(0.12,0.14)
R6	(0.036,0.006)	(0.07,0.014)	-	(0.34,0.19)
R7	(0.032,0.0007)	(0.077, 0.012)	-	(0.34,0.19)
R8	(0.017,0.0001)	(0.47,0.16)	(0.15,0.048)	(0.46,0.26) (0.07,0.003)
R9	(0.009,0)	-	(0.17,0.05)	(0.1,0.007)
R10	(0.009,0)	-	(0.297,0.14)	(0.096,0.002)
R11	(0.002,0)	-	(0.3,0.11)	(0.16,0.01)
R12	(0,00	-	(0.39,0.12)	(0.25,0.026)
R13	-	(0.29,0.24)	-	-
R14	-	(0.37,0.2)	-	-
R15	-	(0.29,0.26)	(0.23,0.48)	(0.26,0.5)
R16	-	(0.4,0.128)	(0.17,0.14)	(0.39,0.29)
R17	-	(0.45,0.4)	(0.37,0.48)	(0.41,0.5)
R18	-	(0.46,0.36)	(0.32,0.39)	(0.44,0.47)
R19	-	-	(0.28,0.2)	-

Stable and unstable resting states: If the corresponding eigenvalues of the intersection have negative real part, the equilibrium is called nodal sink which is seen in $R1$, $R3$. This type of fixed point attracts nearby points in the phase plane graph. This dynamic behavior is called stable resting state [27]. If both eigenvalues of the system have positive real part, the system has an unstable dynamic due to having a nodal source fixed point which repels the nearby area [27] (e.g. $R19$).

Saddle node bifurcation: To have a saddle node bifurcation, a stable resting and a saddle fixed point are required. A saddle fixed point is observed in all areas for GWC and DWC model, which means the system has the potential to have a saddle-node bifurcation (e.g. $R2$, $R4$).

Hopf bifurcation: The appearance or the disappearance of a periodic orbit by changing the parameters of a system is called Hopf bifurcation [25]. During a Hopf bifurcation, by changing the parameters of the system, the corresponding complex conjugate eigenvalues of the system lose their stability, and become absolute imaginary, and a periodic solution ascends in the system [27]. Hopf bifurcation can be classified in two categories: supercritical or subcritical. Supercritical bifurcation results in a stable orbit while subcritical Hopf results in an unstable bifurcation in the system. A supercritical bifurcation has been observed in $R7$, $R10$.

Homoclinic on saddle node: Homoclinic bifurcation occurs when a periodic orbit strikes with a saddle point. Such a behavior can be shown in the $R10$, $R16$, in which there are saddle points and spiral source fixed points. At the bifurcation point, the period of the orbit grows toward infinity, and after the bifurcation the periodic orbit disappears [27].

Bogdanov–Takens (BT) bifurcations: This bifurcation is a mix of other bifurcations and can occur in the systems with dimension of $n = 2$. Two bifurcation parameters are required for this bifurcation to occur. It requires three pairs of eigenvalues [25]:

- A pair of zero eigenvalues

- Eigenvalues with Re $\lambda_j < 0$

- Eigenvalues with Re $\lambda_j > 0$

The system requires one saddle fixed point and one non saddle fixed point equilibrium. These two equilibriums strike and vanish through a saddle-node bifurcation. The non saddle equilibrium generates a limit cycle via a Hopf bifurcation. The limit cycle deteriorate into an orbit homoclinic to the saddle and vanishes by a saddle homoclinic bifurcation [25]. This behavior can be shown in the GWC as well as DWC in most of the regions by varying the bifurcation parameters (B, w_{EI}), such as $R7$, $R8$, and $R16$. A transition from supercritical Hopf to subcritical Hopf is also observed in the system in some regions ($R7$, $R10$). The limit point of cycle (LPC) curve ($R16$) ends with a point in which there is a neutral saddle (NH).

Furthermore, some of the dynamic features are shared among the phase portraits which are presented in Fig. 4.4, similar to GWC, however they have different levels of inhibitory activities.

4.4.2 Time Domain Error Measurements

Fig. 4.5 shows the behavior of GWC and DWC models at 4 types of fixed points in time domain. As shown in these figures, the system is unstable in a source, while it shows a resting state in time domain at a nodal sink fixed point. At a spiral sink, a stable behavior is observed in the system although some fluctuations are displayed. Similar fluctuation is also shown at a saddle point. To compare the original model and the proposed model in terms of timing behavior three metrics have been introduced to measure the similarity of the proposed model with the original model in a time domain period. For the error measurements, initial values for E, and I variables have been considered the same for all regions to be consistent.

Root Mean Square Error (RMSE): which is defined as:

$$RMSE_{func} = \sqrt{\frac{\Sigma(func_{GWC} - func_{DWC})^2}{n}} \tag{4.17}$$

Where, $func$ is X_k, and n represents the number of discrete points which have been used in the design. The overall RMSE error has been defined as the average of calculated RMSE for the two equations of E, and I functions:

$$RMSE = \frac{(RMSE_E + RMSE_I)}{2} \tag{4.18}$$

Err_{max}: Maximum error is defined as the absolute values of the maximum difference between a set of discrete points calculated for the original and proposed model.

$$Err_{max} = |func_{GWC} - func_{DWC}| \tag{4.19}$$

Where $func$ could be E, or I, and GWC refers to original model and DWC refers to digital model. An average of Err_{max} of E and I functions has been considered for the overall error similar to RMSE error.

Correlation (corr): In order to measure overall similarity between the proposed model and original model in timing domain, correlation with a range of (-1,1) can be used:

$$\begin{aligned} & corr(func_{GWC}, func_{DWC}) = \\ \rho_{GWC,DWC} = & \frac{cov(func_G WC, func_D WC)}{\sigma_{GWC,DWC}} \end{aligned} \tag{4.20}$$

where,

$$\rho_{GWC,DWC} = \frac{\sum_{i=1}^{n}(func_{GWC} - func'_{GWC})(func_{DWC} - func'_{DWC})}{(\sum_{i=1}^{n}(func_{GWC} - func_{GWC})^2)^{1/2}((\sum_{i=1}^{n}(func_{DWC} - func_{DWC})^2))^{1/2}} \tag{4.21}$$

Table 4.4: Timing error analysis for different regions of parameters. Correlation has been reported in percentage.time step is considered t = 0.1ms, and simulation has been performed for 100 ms which requires 1000 discrete points

Region	RMSE	Err_{max}	corr%	Region	RMSE	Err_{max}	corr%
R1	0.143	0.3578	12.88	R11	0.0003	0.0011	100
R2	0.016	0.044	95.79	R12	0.0001	0.0008	100
R3	0.0025	0.0034	100	R13	0.0064	0.0127	99.64
R4	0	0	100	R14	0.0081	0.0154	99.87
R5	0.0055	0.0356	97.52	R15	0.0066	0.0126	99.65
R6	0.0025	0.0128	99.98	R16	0.0045	0.044	99.42
R7	0.0031	0.0131	99.97	R17	0.0087	0.0128	99.28
R8	0.0029	0.0117	99.98	R18	0.0818	0.2391	72.67
R9	0.0008	0.0079	100	R19	0.005	0.0371	98.03
R10	0.0010	0.0028	100	Average	0.0085	0.0268	97.98

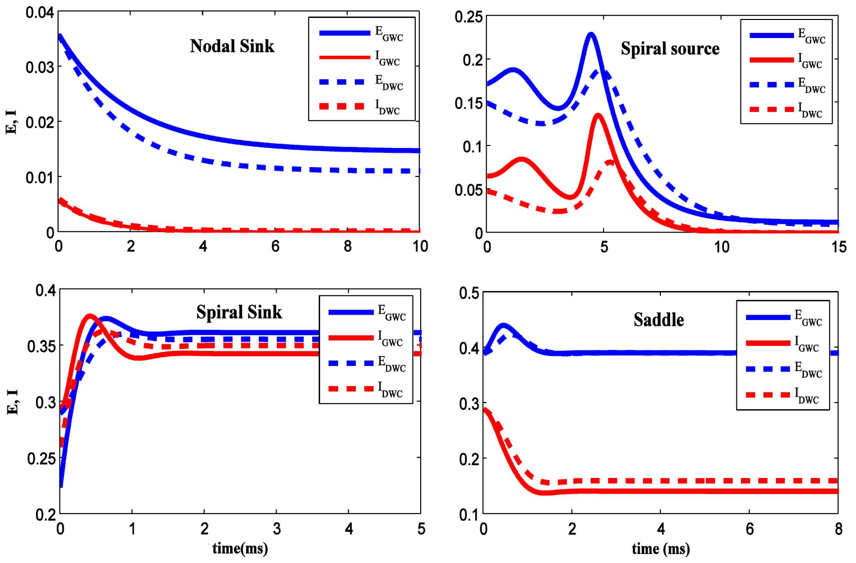


Figure 4.5: Timing behavior of GWC and DWC models at 4 types of fixed points observed in the system

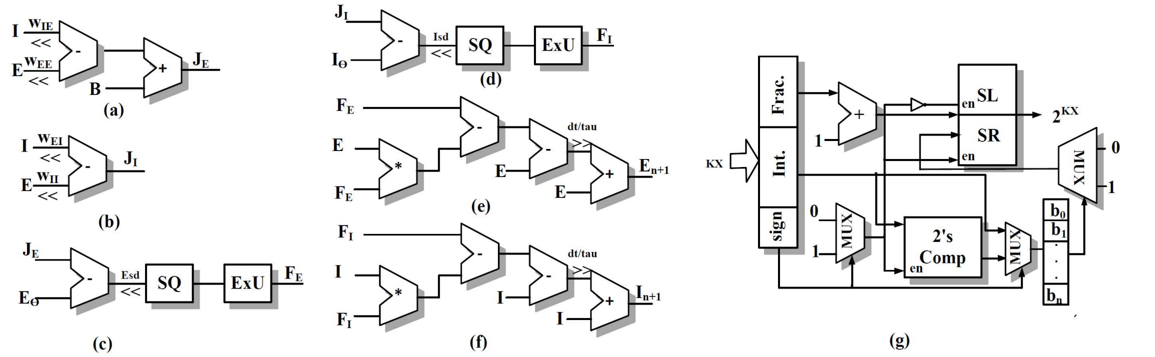


Figure 4.6: Proposed digital architecture for a single $E - I$ pair DWC. (a), and (b) show the J_E and J_I calculation. J_E, and J_I are passed through circuits shown in (c), and (d) in which initially square function and then exponential function are calculated, and then final calculation is performed through architecture shown in (e), and (f). The exponential block has been shown in (g).

An average of correlations for E, and I have been considered as the overall correlation, similarly. The results of the error measurements have been presented in Table. IV.

4.5 Digital Hardware Design

The proposed digital architectures for a pair $E - I$, and a network of coupled $E - I$ are presented in this section. Fig. 6 shows the digital arithmetic diagram for a pair of coupled $E - I$ neurons.

4.5.1 Equation Discretization

As the first step, (1) is required to be discretized for a digital circuit design. This can be performed by Euler method. The main differential equation for E and I can be discretized as below:

$$X_k[n+1] = X_k[n] + \frac{dt}{\tau_x}(-X_k[n] + (1 - X_k[n])F_X(J_{X_k})) \tag{4.22}$$

Where X_k can either be variable E or I. Based on Fig. 4.6, first of all the inputs of function G (equation (10)) have to be calculated. Fig. 4.6(a) and (b) show the required calculations for providing the inputs of G_E , and G_I functions, and the results are named J_E, and J_I. J_E, and J_I are passed into two blocks, Square (SQ) and Exponential (ExU) units, to calculate F_E and F_I functions as shown in Fig. 4.6(c) and (d), respectively.

In the next two scheduling diagrams, F_E and F_I are used for final calculation of the next state of the variables E, and I. Fixed point representation has been used to represent the variables in the digital design of the system. In order to have minimum hardware cost, bit-width minimization is required in the process of the digital design. The integer part is determined based on the (min, max) range of the variables, while the fraction part is determined based on the required accuracy. Therefore, the range of the variables in the data path of the digital system has been calculated for each variable. There is always a tradeoff between the number of bits which is considered for the fractional part and the accuracy. A Higher bit-width provides more accuracy in the system. In order to save more in the hardware cost, a variable data-path (12-30 bits, including 20 bits for fraction) has been utilized in the different parts of the system based on the required accuracy. For example, the exponential unit (ExU) needs a higher bit-width due to having a big output result.

4.5.2 Square Function

According to (9-10), the input of the exponential function is a quadratic function, itself. Therefore, a Square function (SQ) is needed before the exponential unit. Two possible approaches can be used for this unit. The SQ can be implemented using a multiplier which is accurate but expensive in terms of hardware cost. The second approach is using Piecewise Linear (PWL) approximation [10] in which the curve can be approximated using several first order lines. Depends on the required accuracy the number of lines can be varied. PWL is less expensive and faster than a multiplier but its accuracy is lower in comparison with a multiplier. However, it is not suitable for this model, because the output of the square block

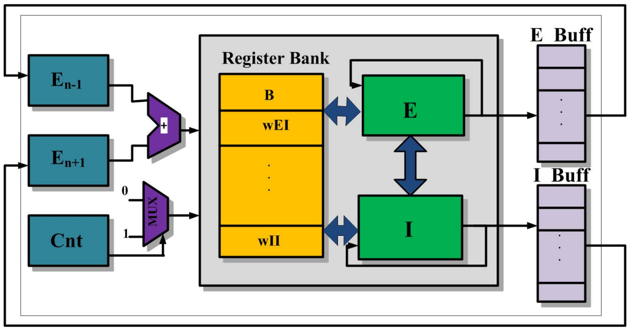


Figure 4.7: General view of time-multiplexing system

is passed into the exponential block. Therefore, the square function must be calculated, accurately. In this work both approaches are employed and compared in terms of hardware cost, speed and accuracy.

4.5.3 Exponential Unit

For implementing the exponential function the technique used in [9] has been used. However, since the range of the outputs of the WC model is limited to $(0, 1)$ which is a small range, a higher accuracy is required for the system. Therefore, the approach used in the [9] has been modified to have a more accurate approximation for the exponential function. In this modification the coefficient of the ExU unit (X) has been replaced by $K = 1.4375$ instead of 1.5. Therefore in order to calculate $exp(X)$, first of all, $KX = 1.4375X = X + 2^{(-1)}X - 2^{(-4)}X$ is calculated and then KX passed to the exponential block. Fig. 4.6(g) shows the exponential unit which has been designed for calculating the exponential functions (F_E, F_I).

Another approach is using LUTs for implementing the exponential function [23]. In order to design an LUT for this block, input and output range of the exponential block is required to be calculated. The input of the exponential function is the output of the square function.

Table 4.5: Synthesis results

Model	ALUT	Logic Reg	Total pins	DSP blocks	$Freq_max$
single DWC_{4M}	1006	535	26	16	238.32 MHz
single DWC_{2M}	436	513	26	8	244.38 MHz
DWC_{4M} coupled	2037	1164	26	32	234.58 MHz
DWC_{2M} coupled	862	1004	26	16	240.79 MHz
Multiplexed 10 DWC	1106	735	26	16	240.5 MHz

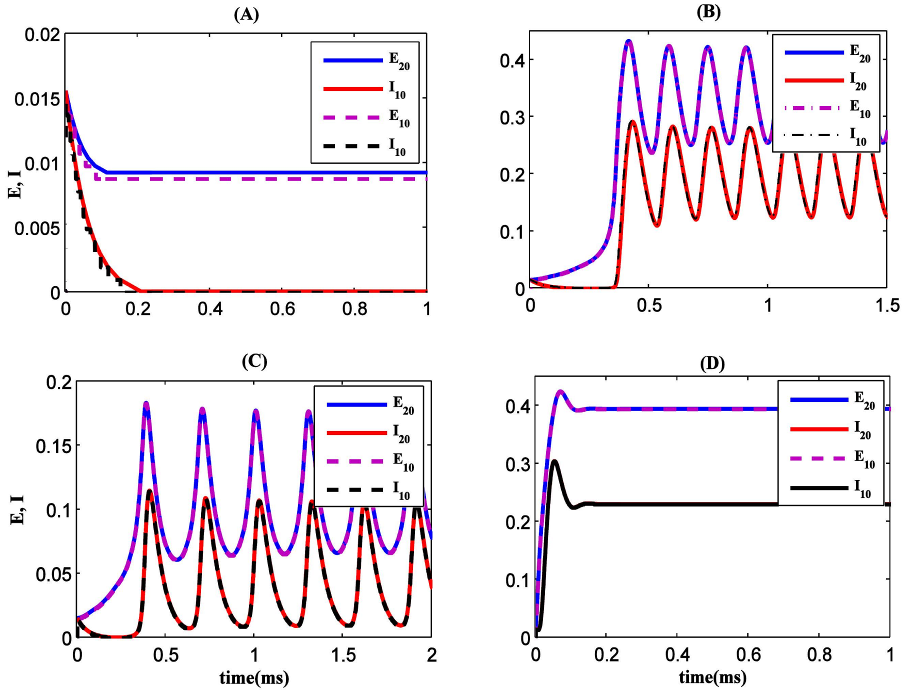


Figure 4.8: Typical output waveforms of the digital circuits implemented on Stratix IV (EP4SGX530KH) for for proposed DWC model

The range of the input of the exponential block are given as below:

$$[Min(|F_X(J_{X_k})|), Max(|F_X(J_{X_k})|)] = [0, 12] \tag{4.23}$$

Because the range of the final outputs of the model is small, $(0, 1)$, and the system is highly non linear, the number of bits which is considered for fraction part is important, and play an important role in the accuracy of the system. In some regions, F_X is as minimum as 0.0001. Therefore, by considering 15 bits for the fraction parts and 5 bits for integer part for input of the exponential block, a total number of 20 bits are required.

The range of the output of the exponential block has been calculated as below:

$$\begin{aligned}&[Min(exp(F_X(J_{X_k}))), Max(exp(F_X(J_{X_k})))]\\&\qquad = [1.495 * 10^{-5}, 0.9999]\end{aligned} \tag{4.24}$$

By considering 16 bits for fraction part to represent the values as small as $1.495 * 10^{-5}$, and 2 bits for the integer part, at least 18 bits are required for the outputs. Accordingly, the size of the look up table for implementing each equation (E, and I) can be calculated as below:

$$Size_{LUT} = 2^{BW_{input}} * BW_{output} = 2^{20} * 18 \tag{4.25}$$

where BW represents the bit width of the input and output of the block. In this case, two look up tables with the size of $2^{20} * 18$ are required to implement both equations which is a high volume of memory, and its not practical for large scale implementation of neural networks.

4.5.4 Population Implementation

In this section, an array of DWC similar to Fig. 4.1 is implemented. For a high accuracy implementation of DWC, 4 multipliers are required and since multipliers are very expensive in terms of hardware implementation, the number of the neurons which can be implemented on a specific FPGA platform is limited. In order to save hardware resources, multiplexing technique and resource sharing can be employed for a large number of neuron implementation. Time multiplexing and resource sharing can be used for a high accuracy and efficient implementation of the model. In this method, a physical DWC hardware is designed initially, and an index generator is employed to control the virtual neurons [26]. In this case, with each clock pulse one neuron output is calculated and stored in the output buffer. Fig. 4.7 shows the overall view of such a system. The main block is made of three parts: E, I, and register bank blocks. Blocks E, and I are responsible to calculate the E and I equations. They also share the data through a bus between E, and I blocks. The register bank stores the required parameters, and it also shares the required data among the E, and I blocks. With the first clock E, and I are calculated using the initial values. With each clock pulse one neuron output is stored in the output buffers until all neurons are calculated and outputs are put in the buffers. The buffers work in a First in$-$First out (FIFO) manner. Therefore, if there are N virtual neurons, N clock pulses are required for all neurons to be calculated. After N clock pulses, they need to be used for the next values calculation. Based on Fig. 1, for a one dimensional neural network implementation, each E_n population needs to be coupled with its neighbor populations of E_{n-1}, and E_{n+1} with coupling factor of α. The coupling factor is equal to 0.1 for the implementations. Therefore, an index recognizer is required in the main block to recognise the previous and next neighbours.

4.6 Experimental Results

The circuits are developed in Verilog Hardware Description Language (VHDL) and have been physically implemented on Altera Stratix IV (EP4SGX530KH) FPGA board. The output is written into GPIO port on the FPGA board, and then the results are captured and demonstrated in a PC. The hardware codes have been synthesized using the Altera compiler, Quartus II. The results of the hardware utilization are given in Table. V. In this table, DWC-4M refers to DWC with 4 multipliers (SQ function is implemented using multiplier), DWC-2M refers to DWC implementation using PWL method for SQ function which reduces the required multipliers to two. DWC coupled indicates a pair of DWC neurons which are coupled with the coupling factor of α. Multiplexed DWC shows the hardware cost for $N = 10$ time-multiplexed DWCs. As shown in Table. V, the hardware cost for the time multiplexed design is almost the same as a single DWC model. The only difference is the required time for each neuron output to be updated. In the time multiplexed design, the required time for each DWC output calculation is N times of a single DWC with dedicated hardware circuit.

Static Timing Analysis (STA) has been performed using TimeQuest timing analysis tool in Quartus II and Synapse Design Constraints (SDC) in Altera Compiler. Based on the timing analysis, the available frequency for a single DWC model with 4 multiplier is 238.32 MHz, and with 2 multiplier is 244.38 MHz. These frequencies are a little reduced when implementing two $E - I$ pairs. For N virtual DWC with a single physical digital hardware, each DWC is updated with the frequency of F_{max}/N. For instance for $N = 10$, each virtual DWC output is updated with the frequency of $F_{max}/10$. Based on Table.V the required Adaptive Look Up Tables (ALUTs) and logic registers for a single DWC is 1% of the available resources on the FPGA board, and the DSP blocks usage is 2 %. Fig. 4.8 shows some of the outputs of the neural network captured from the digital hardware implementation. In this figure, E_{20}, and I_{20} represent the outputs with considering 20 bits for fraction, and, E_{10}, and I_{10}, represent the outputs with 10 bits fraction. Except for Fig. 4.8(A), in which there is a slight variation between the two outputs (because the output range is very small), there is not a high difference between the outputs with 10 bits and 20 bits fraction.

4.7 Conclusion

In this paper, Gaussian Wilson & Cowan model as one of the accurate, and well known models which represents cortex behavior has been implemented for the first time. A digital model for hardware implementation of the model has been proposed and compared with the original model. The model has been proven to be able to follow the original model in both dynamic behavior and timing analysis, accurately. The mathematical analysis show that the digitized models are able to reproduce the same bifurcations and dynamic behavior. Time domain analysis indicates that the model show up to 97% similarity. RMSE and maximum error have been measured as minimum as 0.0085, 0.0268 in average, respectively. In the next step, digital architecture was designed for the proposed model and compared with PWL method in terms of the amount of resources they require and the accuracy they provide. There is always a trade off between the accuracy and the hardware cost. DWC-4M implements the model using 4 multipliers, and provides higher accuracy with a lower speed, while the DWC-2M model represents the model using two multipliers, lower accuracy and higher speed.

In the final step, the digital model was compiled and synthesized in Quartus II Altera compiler and physically implemented on the DE IV Stratix FPGA board. Maximum frequency was obtained as 238.32 and 244.38 for the DWC-4M, and DWC-2M, respectively. The hardware cost reports show that DWC-4M takes maximum 2% of the FPGA board for a single pair of $E-I$ implementation, which means a number of 50 pairs can be implemented on the board. With keeping the IO pins to the minimum, this number for DWC-2M can be up to 100. In order to implement a higher number of neurons on the board, one possibility is implementing virtual neurons using time-multiplexing, and resource sharing techniques. A time-multiplexing design has been performed for the model. A linear coupled population of the DWC was performed in this work which can be extended into a two-dimensional implementation for representing a complete DWC-based model of neocortex.

4.8 Jacobean Coefficient for GWC and DWC Models

$$A = \frac{2w_{EE}exp(-\frac{(B-E_\theta+w_{EE}E-w_{IE}I)^2}{E_\theta^2}) * (E-1) * (B-E_\theta+w_{EE}E-w_{IE}I)}{E_\theta^2} +$$

$$exp(-(\frac{E_\theta}{E_{sd}})^2) - exp(-\frac{(B-E_\theta+w_{EE}E-w_{IE}I)}{E_\theta^2} - 1$$

$$B = \frac{2w_{IE}exp(-\frac{(B-E_\theta+w_{EE}E-w_{IE}I)^2}{E_\theta^2}) * (E-1) * (B-E_\theta+w_{EE}E-w_{IE}I)}{E_\theta^2}$$

$$C = \frac{-2w_{EI}exp(-\frac{(I_\theta-w_{EI}E+w_{II}I)^2}{I_{sd}^2}) * (I-1) * (I_\theta-w_{EI}E-w_{II}I)}{I_s d^2}$$

$$D = \frac{2w_{II}exp(-\frac{(I_\theta-w_{EI}E-w_{II}I)^2}{I_{sd}^2}) * (I-1) * (I_\theta-w_{EI}E+w_{II}I)}{I_{sd}^2} - exp(-\frac{(I_\theta-w_{EI}E+w_{II}I)^2}{I_{sd}^2} - 1$$

$$A' = \frac{2}{2^{\frac{(K(B-E_\theta+Ew_{EE}-Iw_{II})^2)Kw_{EE}log(2)(E-1)(B-E_\theta+Ew_{EE}-Iw_{IE})}{E_{sd}^2}}} - \frac{1}{2^{(\frac{(K(B-E_\theta+Ew_{EE}-Iw_{IE})^2)}{E_{sd}^2})}} - 1$$

$$B' = -\frac{2}{2^{\frac{(K(B-E_\theta+Ew_{EE}-Iw_{IE})^2)Kw_{IE}log(2)(E-1)(B-E_\theta+Ew_{EE}-Iw_{IE})}{E_{sd}^2}}}$$

$$C' = -\frac{2}{2^{\frac{(K(I_\theta-Ew_{EI}-Iw_{II})^2)Kw_{EI}log(2)(I-1)(I_\theta+Ew_{EI}-Iw_{II})}{I_{sd}^2}}}$$

$$D' = \frac{2}{2^{\frac{(K(I_\theta-Ew_{EI}+Iw_{II})^2)Kw_{II}log(2)(I-1)(I_\theta-Ew_{EI}+Iw_{II})}{I_{sd}^2}}} - \frac{1}{2^{(\frac{(K(I_\theta-Ew_{EI}-Iw_{II})^2)}{I_{sd}^2})}} - 1$$

References

[1] A. L. Hodgkin, and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *Bulletin of mathematical biology* , vol(52), pp. 25-71, 1990.

[2] C. Morris and H. Lecar, “Voltage oscillations in the barnacle giant muscle fiber,” *Biophysical journa*, vol. 35, no. 1, pp. 193-213, 1981.

[3] R. Jolivet, A. Rauch, H.R. Lüscher, and W. Gerstner, “Integrate-and-Fire models with adaptation are good enough,” *In Advances in neural information processing systems*, pp. 595-602, 2006.

[4] M. G. .F. Fuortes, and F. Mantegazzini, “Interpretation of the repetitive firing of nerve cells,” *The Journal of general physiology*, vol. 45, no. 6, pp. 1163-1179, 1962.

[5] M. E. Izhikevich,”Which model to use for cortical spiking neurons?,”*IEEE transactions on neural networks*, vol(15), no.5, pp. 1063-1070, 2004.

[6] L. J. Hindmarsh, and R. M. Rose, “A model of neuronal bursting using three coupled first order differential equations,” *Proceedings of the royal society of London B: biological sciences*, vol. 221, no. 1222, pp.87-102, 1984.

[7] R. FitzHugh, ”Mathematical models of threshold phenomena in the nerve membrane,” *Bulletin of Mathematical Biology*, vol. 17, no. 4, pp. 257-278, 1955.

[8] S. Yang, J. Wang, S. Li, H. Li, X. Wei, H. Yu, and B. Deng, “Digital implementations of thalamocortical neuron models and its application in thalamocortical control using FPGA for Parkinson? s disease,” *Neurocomputing*, vol. 177, pp. 274-289, 2016.

[9] S. Gomar, and A. Ahmadi, “Digital multiplierless implementation of biological adaptive-exponential neuron model,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1206-1219, 2014.

[10] H. Soleimani, A. Ahmadi, and M. Bavandpour, “Biologically inspired spiking neurons: Piecewise linear models and digital implementation,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 12, pp. 2991-3004, 2012.

[11] R. M. Wang, T. J. Hamilton, J. C. Tapson, and A. van Schaik, “A mixed-signal implementation of a polychronous spiking neural network with delay adaptation,” *Frontiers in neuroscience*, 2014.

[12] H. Soleimani, A. Ahmadi, M. Bavandpour, and O. Sharifipoor, “A generalized analog implementation of piecewise linear neuron models using CCII building blocks,” *Neural Networks*, vol. 51, pp. 26-38, 2014.

[13] M. Hayati, M. Nouri, M., S. Haghiri, S., D. Abbott, “Digital multiplierless realization of two coupled biological Morris-Lecar neuron model,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no.7, pp. 1805-1814, 2015.

[14] http://www.scholarpedia.org/article/Neuralfields

[15] R. L. Beurle, “Properties of a mass of cells capable of regenerating pulses,” *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 240, no. 669, pp. 55-94, 1956.

[16] J. S. Griffith, “A field theory of neural nets: II. Properties of the field equations,” *Bulletin of Mathematical Biology*, vol. 27, no. 2, pp. 187-195, 1965.

[17] H. R. Wilson, and J. D. Cowan, “A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue,” *Biological Cybernetics*, vol. 13, no. 2, pp. 55-80, 1973.

[18] S. I. Amari, “Homogeneous nets of neuron-like elements,” *Biological cybernetics*, vol. 17, no. 4, pp. 211-220, 1975.

[19] H. G. Meijer, T. L. Eissa, B. Kiewiet, J. F. Neuman, C. A. Schevon, R. G. Emerson, R. Goodman, “Modeling focal epileptic activity in the Wilson–Cowan model with depolarization block,” *The Journal of Mathematical Neuroscience (JMN)*, vol. 5, no. 1,p. 7, 2015.

[20] D. Monroe, “Neuromorphic computing gets ready for the (really) big time,” *Communications of the ACM*, vol. 57, no. 6, pp. 13-15, 2014.

[21] S. K. Boddhu, and J. C. Gallagher, “Qualitative functional decomposition analysis of evolved neuromorphic flight controllers,” *Applied Computational Intelligence and Soft Computing*, p. 2, 2012.

[22] H. R. Wilson, and J. D. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophysical journal*, vol. 12, no.1, pp. 1-24, 1972.

[23] K. Leboeuf, A. Hosseinzadeh Namin, R. Muscedere, H. Wu, and M. Ahmadi, "High speed VLSI implementation of the hyperbolic tangent sigmoid function," *Third IEEE International Conference on In Convergence and Hybrid Information Technology (ICCIT)*, vol. 1, pp. 1070-1073, 2008.

[24] J. M. T. Thompson, and H. B. Stewart, "Nonlinear dynamics and chaos," *John Wiley & Sons*, 2002.

[25] http://www.scholarpedia.org/article/Bogdanov-Takens-bifurcation

[26] R.M. Wang, T. J. Hamilton, J. C. Tapson, and A. van Schaik, "A neuromorphic implementation of multiple spike-timing synaptic plasticity rules for large-scale neural networks," *Frontiers in neuroscience*, 2015.

[27] E. M. Izhikevich, "Dynamical systems in neuroscience: the geometry of excitability and bursting," *The MIT press*, 2007.

Chapter 5

A Digital Central Pattern Generator Based on Hindmarsh-Rose Spiking Neuron Model

5.1 Introduction

Central Pattern Generators (CPGs) are autonomous groups of biological neurons which produce rhythmic patterns to control the locomotion part of the animals without receiving a rhythmic input [1]. Accordingly, a CPG can be constructed by coupling bio-inspired neurons known as spiking neurons [1]- [2]. CPGs structures are known as efficient and stable construct for controlling movement despite their simplicity. In addition, CPGs are event-based system which means that the signals are transferred in the form of events and only when data are present. Moreover, it offers a parallel computation due to multiple units, i.e. spiking neurons. These characteristics make CPGs promising computational primitives for controlling motor outputs in biomimetic robotic applications. In this study, a Spiking Neural Network (SNN) is used to reproduce the behavior of a CPG [3].

A spiking neuron model is a dynamical system made of mathematical coupled differential equations which produce output patterns similar to what is observed in biology [4]. Many spiking neurons have been presented throughout the years which vary in terms of complexity and biological accuracy [4]-[9]. Among them, Hodgkin Huxley (HH) [5] provides the highest accuracy by describing the action potential generation with four detailed differential equations, six functions and tens of parameters while the Integrate and Fire (IF) model [4] presents the least biological accuracy with only one differential equation. Depending on the application, either a detailed model or a simple model can be employed. For instance, for a deep learning application a simple model like IF is helpful due to its low computation complexity, while for developing a prosthesis an accurate biological model is essential. Two and three dimensional models have been also presented in the literature which are simplified versions of the classical HH formulation. Among these models, Hindmarsh-Rose (HR) introduced by Hindmarsh and Rose [9] in 1984 is able to reproduce a wide range of behaviors shown in biology such as tonic, bursting, and spike bursting adaptation with a three-dimensional equation system [2]. An HR-based CPG and its feasibility for employing in engineering applications have been studied in [2]. HR has some advantages which makes it appropriate to develop a CPG [2]:

- The duty cycle of HR output pattern can be easily regulated for a given frequency which is desired in a CPG system.
- The diversity of frequency can be satisfied by altering two parameters.
- The accessibility to be entrained can be easily provided by adding a sensory feedback to the model.
- HR is immune to the possible sources of noise coming from the sensory feedback.
- Normal behavior can be recovered under the influence of a perturbation signal.

Efficient implementations of CPGs can be employed as embedded controllers in practical engineering applications like robotic systems including bipedal robots [10] or it can be used in a rehabilitation system to generate patterns for walking to help people with disabilities [2]. Although a few works have presented hardware implementations for CPG network such as [11], and [12] but these implementations are not efficient enough for a large system as they require a high volume of digital resources, and offer low frequency. In this paper, a novel, efficient and high speed hardware implementation is proposed for a digital CPG based on HR.

Hardware implementation of a bio-inspired neural network is known as Neuromorphic Engineering (NE), a field of research which was coined for the first time by Carver Mead in 1990 [13]. Various methodologies can be employed to implement a bio-inspired neural network including digital/analog ASIC design, and programmable hardware such as Field Programmable Gate/Analog Arrays (FPGA or FPAAs) each of which has some advantages and disadvantages [14]. FPAAs solve the problem of analog approach by offering a reconfigurable platform for analog designs. However, this approach suffers the limited number of available resources on a typical platform. FPGAs on the other hand offer a large number of resources to implement a bio-inspired neural network in a system level dynamic and its widely used in several works such as [14]-[15].

Several hardware implementations have been proposed for CPG with different neuron models in recent years such as [16]-[18]. However, a CPG made of HR model has not been developed for hardware implementation. In this study, a digital hardware platform is developed for implementing a CPG based on HR spiking neuron.

In the following sections, a brief is given on HR model in section 5.2. In section 5.3, the state of the art methodologies and proposed model are presented for HR neuron. With the aim of an efficient hardware implementation, HR model is implemented by a reduced size Look Up Table (LUT) methodology to minimize the computation complexity while keeping the biological accuracy high enough. In addition, the method is compared with the state of the art methodologies and error measurements are performed in this section. In section 5.4, dynamical analysis is performed for the digital model and coupled HR. A CPG is made Subsequently, the proposed digital model results in a reduction in the FPGA resources utilization in the implementation phase. Section 5.4 presents the CPG architecture composed of coupled HR. Verilog Hardware Description Language (VHDL) is employed to develop the digital system throughout this work. The proposed platform for test is given in section 5.5, and the paper is concluded in section 5.6.

5.2 A Brief on HR Neuron

In this section the spiking model is introduced briefly, and the proposed model and timing, and dynamical analysis are given in the next subsections. HR model [2] can be described by three differential equations:

$$\begin{aligned} x^{'} &= y - aF(x) + bG(x) - z + I, \\ y^{'} &= c - dG(x) - y, \\ z^{'} &= r(s(x - x_0) - z). \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} F(x) &= x^3, \\ G(x) &= x^2. \end{aligned} \tag{5.2}$$

where x, y, and z represent the time dependent variables among which x represents the membrane potential, y represents the recovery variable, and z is the adaptation current. The parameter x_0 is a control parameter for altering adaptation current, and it shows x-coordinate of the stable sub-threshold equilibrium point if external current,I, is equal to zero. The other parameters are constants and can be regulated to generate different patterns including tonic, and bursting. Adaptation current, z, slowly fluctuates and it increases when the neuron fires. The dynamics of the model is changed by altering the parameters including

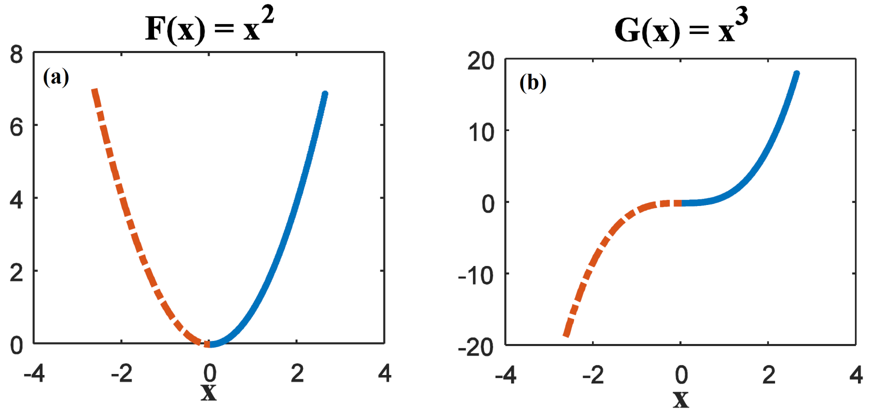


Figure 5.1: Symmetrical non-linear behavior of $F(x)$ (left), and $G(x)$ (right). The functions have been plotted with step size of $\Delta x = 2^(-6)$.

I (external current), b (intrinsic parameter), r (rate of activation for some current), and x_0. The current, i.e. I, has a significant role to shape diverse patterns in the output of the neuron. With small values of I no rhythmic pattern is observed in the output. With slightly increasing the current, the variable x, i.e. membrane potential, exceeds a certain threshold value which results in a rhythmic pattern of spike in the output. With further increasing I, the frequency of the spike is increased, accordingly. A transition from tonic spiking to bursting is also observable in the output of the neuron by increasing the current.

5.3 Methodologies and Proposed Model

5.3.1 State of the Art Methodologies and Proposed Model for HR

As seen from the equations (1) and (2), the neuron behavior is highly nonlinear due to two terms, i.e. F(x), and G(x) shown in Fig. 1(a), and (b), respectively. Nonlinear functions are expensive for implementation due to the required multiplication computation. In order to reduce the hardware implementation cost, approximation is an essential step.

Several methods can be used to approximate a nonlinear behavior such as Piecewise Linear (PWL) method in which the nonlinear behavior can be approximated using several first-order lines. This method has been widely used in NE such as [10], [14]. This method reduces complexity while provides an acceptable accuracy.

Base-2 method has also been used in several works especially for exponential function implementation [14], [18]. This method is appropriate for exponential functions with large input and output ranges. Although base-2 conversion is an efficient way for exponential function with large input ranges, it is not efficient for functions with lower level of complexity and small input and output ranges.

LUTs (memory) method on the other hand is used for storing discrete points and retrieving the values when the corresponding address is triggered. The accuracy of this method is dependent to the size of the LUTs. The larger LUTs, the higher accuracy is provided. LUT based method is not efficient for highly non-linear functions with large inputs, but it provides an accurate implementation with an affordable hardware cost for functions with small to medium input/output ranges.

Accordingly, depending on the function which is aimed to be implemented, one can find an appropriate, and efficient methodology for hardware implementation. PWL method provides an acceptable accuracy with lower resource utilization, while LUT provides a higher accuracy but it suffers higher hardware cost for functions with large input/output ranges. However, for a function with a small input/output range, the use of LUTs provides an acceptable trade-off between the accuracy and hardware cost.

As shown in Fig. 1, the required input range for $F(x)$, and $G(x)$ is small, $(0, 2.625)$, and it indicates that LUT method can be used without needing large LUTs for storing the values. The size of the LUTs can further be reduced thanks to the symmetrical feature which exists in $F(x)$, and $G(x)$ functions.

As can be seen in Fig. 1(a), the absolute values of $F(x)$ is the same for positive and negative values of x with the same absolute values which means that for the half negative of the graph, storing the values is not necessary as the output sign can be inverted after retrieving the value for its positive input.

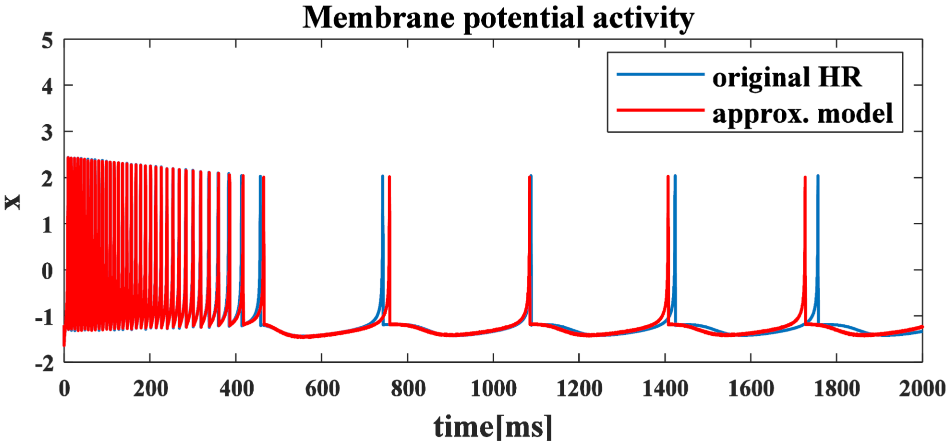


Figure 5.2: Error measurement for different sizes of LUTs. Blue represents MAE, and orange represents RMSE measurement. The errors have been measured with digitally feasible parameter values. For Bursting: $I = 2.5$, $r = 2^{-10}$, Tonic: $I = 1.5$, $r = 2^{-7}$, Fast spiking: $I = 3$, $r = 2^{-10}$, Quiescence: $I = 1$,$r = 2^{-10}$. Time step is equal to 0.0313 ($dt = 2^{-5}$) for all measurements.

On the other hand, the values of $G(x)$ is the same for identical absolute values of x (Fig. 1(b)), which means there is no need to store the outputs for negative values of x. In the other words, in the digital hardware, $G(x)$ can receive the absolute values of x and return the corresponding output for both positive and negative values of x with identical absolute values.

Therefore, thanks to the symmetrical feature of the nonlinear functions the size of the LUTs for $G(x)$, and $F(x)$ could be reduced to half, while the accuracy remains unchanged. In Fig. 5.1, $F(x)$, and $G(x)$ has been plotted with the step size of $dx = 2^(- 6)$, with considering 2 bits for the integer part, the size of the LUT can be calculated as $2^8 = 256$ which is affordable in hardware implementation and it provides a high biological accuracy at the same time. Fig. 2 shows a typical output pattern generated for the original model, and the approximated model.

5.3.2 Error Analysis

In this section, the proposed method is evaluated by measuring the error between the original model and proposed model. Two criteria which are widely used in previous works [10],[12], [16] have been considered to evaluate the model. Mean Absolute Error (MAE): which is defined as below:

$$MAE = \frac{1}{n}\sum_{1}^{n} e_i, \tag{5.3}$$

, where

$$e_i = |x_{origin}(i) - x_{approx.}(i)| \tag{5.4}$$

Root Mean Square Error (RMSE): which is calculated as below: **Root Mean Square Error (RMSE)**: which is defined as:

$$RMSE_{HR} = \sqrt{\frac{\Sigma_{i=1}^{n}(DHR(i) - HR(i))^2}{n}} \tag{5.5}$$

where, n is the number of discrete points calculated for a period of time. The index of the original values and proposed values is represented by i. When a digital approximation is proposed truncation error in implementation phase is unavoidable. In order to avoid underestimating the effect of truncation error in hardware design phase all values (including constant parameters and time step) have been converted to a base-2 which is convenient for digital implementation. For example, the time step of simulation is equal to $dt = 2^(- 5)$. Therefore, the parameter values do not need further modifications in the design phase, and the values in simulations will be exactly the same in the hardware. This error factor has not been considered in previous works. Fig. 5.3 shows the error analysis for different sizes of LUTs.

In this figure, the horizontal axis indicates the step sizes for stored values of x. For example for $dx = 2^(- 5)$, $F(x)$, and $G(x)$ have been calculated for x values in the range of $(0, 2.625)$ which are stored in the memory with a step size of dx. As shown in Fig. 5.3, both MAE, and RMSE generally drop with smaller step sizes (larger LUTs). For some of

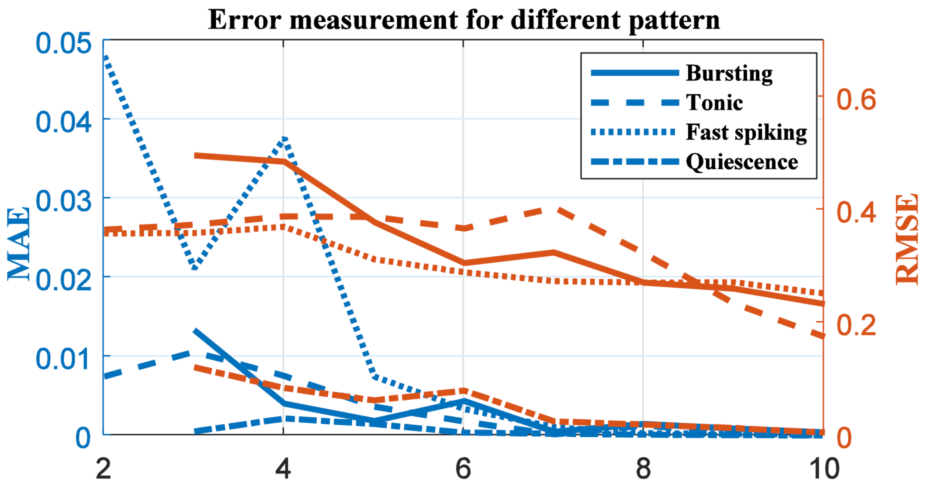


Figure 5.3: The simulated spiking neuron for the original model and the approximated model.

the patterns, the approximated model is not able to predict the output pattern accurately with large step sizes. For instance, for $dx = 2^(- 2)$ the model is not accurate enough to predict the output patterns for quiescence an bursting. However, one can understand that with a step size of $dx = 2^(- 5)$which requires a LUT as small size as 22, MAE drops to an amount as low as to 0.0156 which is far smaller than the one reported in [18] which has employed PWL and base-2 methodology. According to the error analysis performed in this section, it could be concluded that with $dx = 2^(- 5)$, the errors are small enough (MAE below 0.01, and RMSE below 0.4) to accurately predict the original model for all patterns.

5.4 Dynamical Analysis of the Digitized Model

5.4.1 Single Digitized HR Dynamic

In order to analyze the dynamical behavior of the proposed Digital HR (DHR), nullclines of the system must be obtained. The main purpose of the third equation in (1) is adding a slow current which repolarizes the membrane potential and results in a periodic spike pattern instead of shutting down the neuron after a few spikes [9]. Since z variable is slow

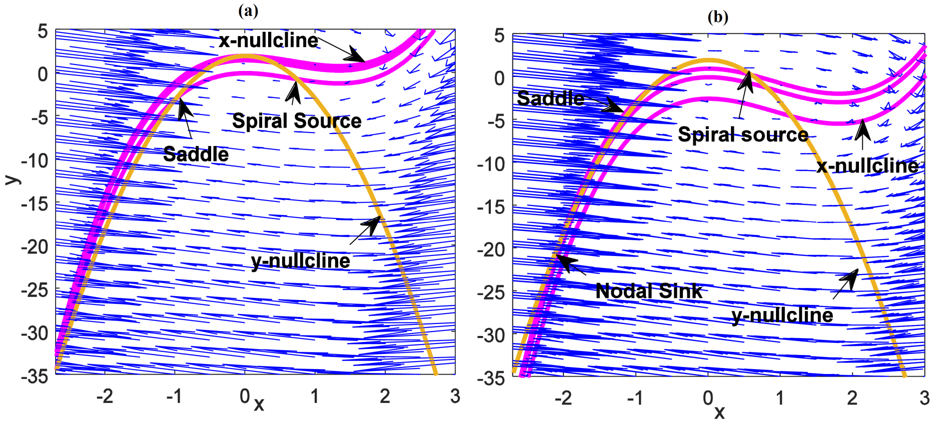


Figure 5.4: Dynamic behavior of DHR with 32 bit fixed-point representation for (a) tonic, and (b) bursting. Blue represent the trajectories. X-nullcline and y-nullcline are represented by pink and orange, respectively. The intersections of the two nullclines represent the EP of the system.

in comparison with y, x variables, the analysis of the 3-D equation is very close to the 2-D model [9]. Nullclines of the 2-D model can be calculated as below:

$$\begin{aligned} x^{'}[n] &= 0, \\ y^{'}[n] &= 0, \end{aligned} \tag{5.6}$$

$$\begin{aligned} y_x[n] &= aF(x[n]) + bG(x[n]) - z[n] + I, \\ y_y[n] &= c - dG(x[n]), \end{aligned} \tag{5.7}$$

where n represents nth value of discrete x and y variables. all variables in the calculations are represented by fixed-point representation. By solving the equations (5), the intersections of the two nullclines, i.e. Equilibrium Points (EPs) of the system, can be computed. The neuron output could shape different patterns including tonic spike or bursting depending on the input parameters r, and s; consequently, the type and the combination of the EPs that the system experience during the stimulation. The type of the EPs can be determined by calculating the Jacobean Matrix (JM) and finding the corresponding Eigen values at each EP. Jacobean matrix for DHR can be obtained as below:

$$JM_{DHR} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{5.8}$$

where,

$$A = \frac{\partial F(x)}{\partial x}, B = \frac{\partial F(x)}{\partial y}, \tag{5.9}$$

$$C = \frac{\partial G(x)}{\partial x}, D = \frac{\partial G(x)}{\partial y} \tag{5.10}$$

$$\begin{gathered} A = \frac{\partial F(x,y)}{\partial x}, B = \frac{\partial F((x,y)}{\partial y} \\ C = \frac{\partial G(x,y)}{\partial x}, D = \frac{\partial G(x,y)}{\partial y} \end{gathered} \tag{5.11}$$

$$JM_{DHR} = \begin{bmatrix} -3ax[n]^2 + 2bx[n] & -1 \\ -2dx[n] & 1 \end{bmatrix} \tag{5.12}$$

The eigenvalues can be calculated using the following equations:

$$\begin{aligned} T &= A + B, \\ Z &= AD - BC \\ P(\lambda) &= \lambda^2 - T\lambda + Z \end{aligned} \tag{5.13}$$

A typical set of Eigen values and the types of the EPs of the system for a set of given parameters are given in Table 5.1.

To perform the dynamic analysis for the digital model discrete points have been stored in a variable and fixed-point calculations with bit-width of 32 have been performed in MATLAB. When simulation is perfumed in MATLAB, a floating point representation is

Table 5.1: EPs of the system and the corresponding Eigen values and types given for $a = 1$, $b = 2.7$, $c = 2$, $d = 5$, $s = 4$, $I = 2.25$ when $z = 3.5$, with considering 32-bit Fixed-point values for representing the variables.

EP	(x,y)	T	Z	Eigen Values	Type
EP1	0.5 0.727	Positive	positive	0.475+1.68i, 0.475-1.68i	spiral source
EP2 0.283	-0.67	negative	negative	-6.190, 0.290	unstable focus
EP3	-2.13 -20.79	negative	positive	-25.96, -0.147	stable nodal sink

used for representing the numerical values by default which makes the dynamic analysis inaccurate for the digital model.

By performing fixed-point analysis for the digital model it is shown that similar to the original model proposed in [9], three types of EP are observed in the digital model system including spiral source, saddle point and nodal sink. Fig. 5.4 shows the x-y phase for two modes of tonic and bursting behavior. The arrows show the trajectory of the system. As shown there is a possible unstable focus orbit around the spiral source EP. X-nullcline (shown in pink) position changes during the stimulation and adaption current (z variable) adjustment to have one or three EPs which results in various patterns including tonic and bursting patterns in the output.

By studying the phase-plane and EPs of the system, it can be found that during a tonic operation, the x-nullcline moves upwards and the nodal sink EP moves downward. (Fig. 5.4(a)) Consequently, the saddle EP moves upward until the limit cycle crosses the saddle EP. At this point the firing stops and the phase point moves in to the narrow region below the saddle point EP, and gradually reaches to the nodal sink EP. This movement hyperpolarizes the wave and with the change of the adaption current (z) the x value goes above x_0 as the nodal sink goes back to its previous position.

In the second mode, i.e. bursting, the system transits between two states of dynamics

(Fig. 5.4(b)). One of which holds one unstable EP and the other one holds three EPs. This transition between the two dynamic states generates burst of spikes in the output of the neuron. Essentially, at the beginning of applying the stimulation current the system states changes from having three EPs to one unstable spiral source EP by moving the x-nullcline downwards (Fig. 5.4(b)). Burst of spikes is generated due to the existence of an unstable focus orbit around the spiral source EP. With the adaption current adjustment, i.e. z, x-nullcline moves upward which creates two intersections with y-nullcline (creating two EPs). X-nullcline continues going upward until the orbit reaches saddle EP and enters the narrow region below the saddle point.

5.4.2 Coupled DHR Dynamic

The dynamical behavior of coupled DHR has been investigated in Fig. 5. As shown in this figure, the $x1 - x2$ phase-plane and the corresponding output patterns are given for four states of: asynchronous bursting, synchronous bursting, synchronous tonic and asynchronous tonic patterns. Fixed-point representation with the bit-width of 32 has been used for implementing F(x), and G(x) to investigate the behavior of coupled DHR. As seen, the dynamical behavior of coupled DHR matches the behavior which is observed in the original model reported in [11]. The digital model is successfully able to reproduce coupled patterns in the output.

5.5 CPG Architecture and Bipedal Gait Generation

5.5.1 Architecture

Fig. 5.6 (a) shows the top level design for DHR implementation. As can be seen from this figure, the system is made of five main modules including three modules for implementing the three equations of x, y, z variables, and two LUTs for implementing $F(x)$, and

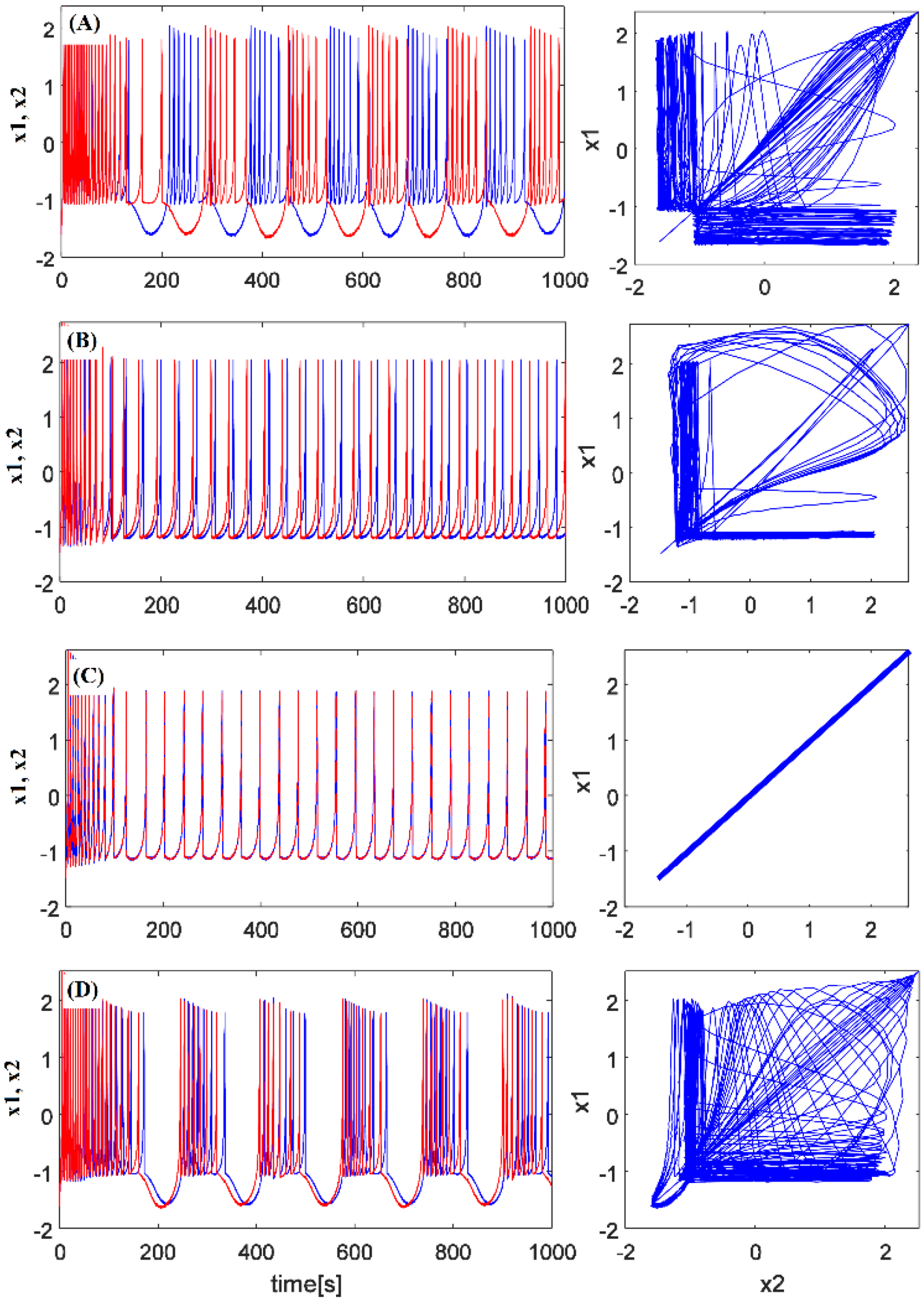
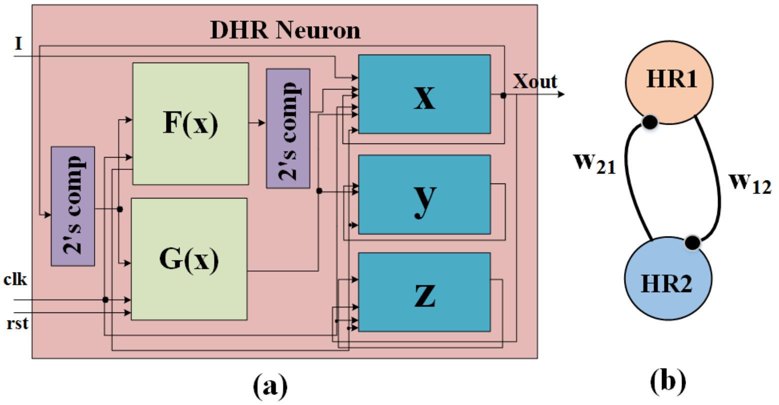


Figure 5.5: Coupled DHR. A) asynchronies bursting, B) synchronous tonic, C) synchronous tonic, and D) synchronies bursting.

$G(x)$. The system designed for DHR is synchronous, therefore the status of the registers is controlled by a global clock. A reset signal is also used for resetting the system. The



(a) (b)

Figure 5.6: (a) Top level design for a single DHR neuron model, (b) a CPG block made of two coupled DHR neurons.

variables are assigned to an initial value and then the next values are calculated with each clock arrival.

DHR neurons can be coupled through the following equations to build a CPG block [2]:

$$\begin{aligned} x[n+1] &= x[n] + dt[y[n] - aF(x[n]) + bG(x[n]) - z[n] \\ &\quad + I + \sum_{j=1, j\neq 1} \omega_{ij}\phi(x[n]_j) + ge], \\ y[n+1] &= c - dG(x[n]) - y[n], \\ z[n+1] &= r(s(x[n] - x_0) - z[n]). \end{aligned} \tag{5.14}$$

where ω represents the connection weight strength, $\varphi isthecouplingfunctionwhichisequalto\varphi = max(x_j, 0)$, e is the sensory feedback and g is the gain of sensory feedback. If $\omega_{ij} = 0$ it means that there is no connection between the neurons, if $\omega_{ij} > 0$, it means that there is an excitatory between the neurons, and $\omega_{ij} < 0$ indicates an inhibitory connection between the two neurons. A basic CPG building block can be built by coupling a pair of DHR neurons as shown in Fig. 5.7 (b). By adjusting the coupling weights and input parameters such as I, and r an anti-phase pattern can be obtained at the output of the neurons.

Fig. 5.7 shows the hardware implementation for a CPG composed of four coupled

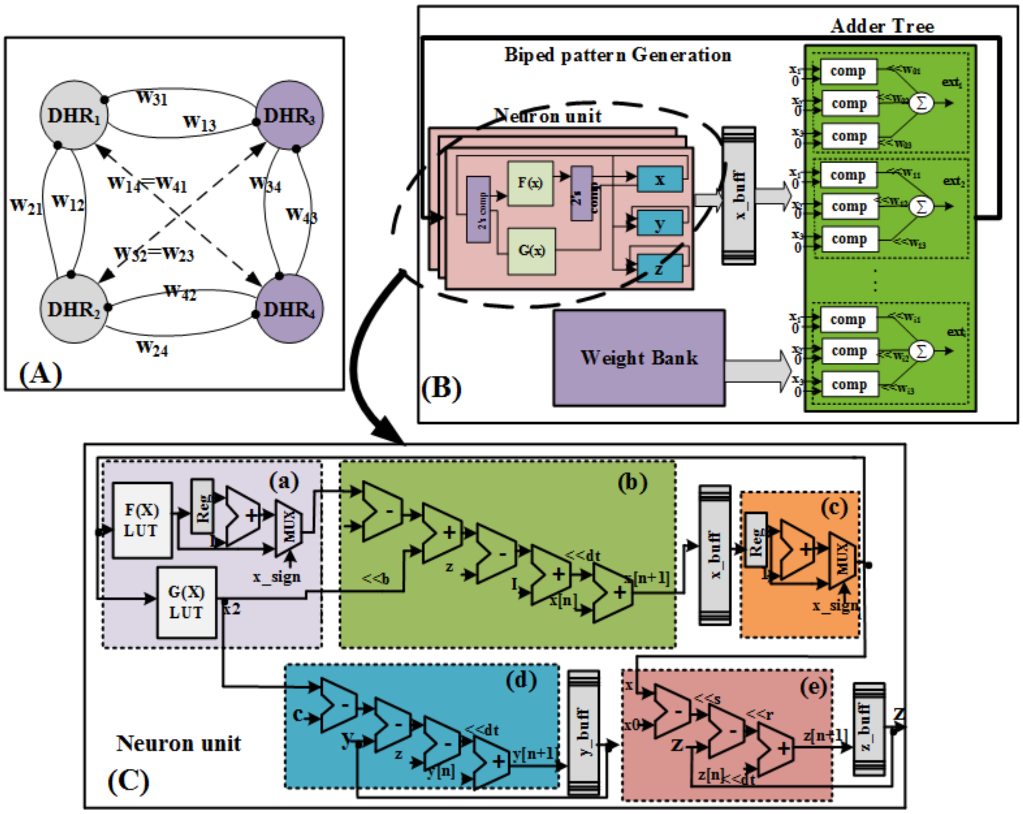


Figure 5.7: Hardware architecture of the proposed CPG. A) Neural connection configuration for the CPG composed of four DHR neurons for biped gait patterns generation. DHR1 and 2 represent the left leg and DHR3 and 4 show the right leg. B) Top-level architecture, C) Details of hardware implementation for DHR neuron model.

DHR neurons. Fig. 5.7(A) shows the schematic of neural connection for a bipedal pattern generation. Fig. 5.7 (B) displays the top level architecture designed for the CPG shown in Fig. 5.7(A).

As shown in Fig. 5.7(B), The neuron unit computes the DHR membrane potentials in a pipeline manner, the result is passed to a buffer and once the results of the neurons are ready, they are passed to the adder tree to calculate the external term for each neuron. Based on the neuron configuration shown in Fig. 5.7 (A), each neuron receives three external terms (ext_i) from other neurons, which is the summation of the weighted coupling functions. The adder tree reads the weigh values form the weight bank and computes the external terms.

Table 5.2: Relative phases for four bipedal gait generation[2]

Gait	Left leg	Right Leg
Two legged hop	$x1(t), x1(t)$	$x1(t), x1(t)$
Walk	$x1(t), x1(t+1/2)$	$x1(t+1/2), x1(t)$
Two legged jump	$x1(t), x1(t+1/2)$	$x1(t), x1(t+1/2)$
Run	$x1(t), x1(t)$	$x1(t+1/2), x1(t+1/2)$

Fig. 5.7(C) demonstrate the details of hardware implementation for the DHR neuron model. Two LUTs have been considered for $F(x)$, and $G(x)$. The values of square and quadratic functions for the required range of x have been stored in LUTs. Depending on the sign of the variable x, a 2's complement operation may be needed. If $x < 0$, the output of the $F(x)$ LUT must be inverted, otherwise the output is passed to the next part of the implementation which shown in Fig. 5.7 (b), and (d). Parts (b), (d), and (e) represent the digital circuits for equations x, y, and z, respectively. The output of each section is placed in a buffer in a FIFO manner. Once the result of the x circuit (part (b)) is ready, the result must be given to the LUT to restore the corresponding values. At this stage, the sign of x must be checked since the LUTs only store the values for positive x. Therefore, if $x < 0$, the sign must be inverted to retrieve the corresponding $F(x)$, $G(x)$ values.

Fig. 5.8 shows a typical anti-phase pattern obtained from the digital design simulation performed in ModelSim software for a CPG block composed of two couple DHR neurons. As seen by adjusting the weight connections anti-phase patterns can be produced in the output of the CPG block. With a similar approach additional neurons can be coupled to generate extra patterns for more complex behavior in the output of the CPG.

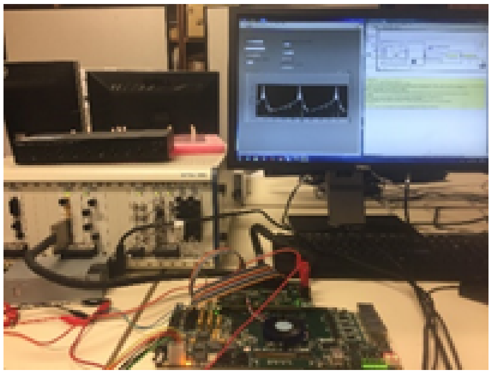


Figure 5.8: The proposed Digital Platform composed of the FPGA board, PXI national Instrument and LAbVIEW.

5.5.2 Bipedal Gait Generation

To generate four different gaits including two-legged hop, walk, two legged jump and run, the connection weights between the neurons must be tuned properly. The relative phase between the two legs of the bipedal CPG shown in Fig. 7(A) must follow the Table I given in [2]. Weight connections for the primary gaits generations are given in [2], and these values are used in this work. The coupling matrix is given as below [2]:

$$W = \begin{bmatrix} 0 & k_1 & k_1 & k_2 \\ k_1 & 0 & k_2 & k_1 \\ k_1 & k_2 & 0 & k_1 \\ k_2 & k_1 & k_1 & 0 \end{bmatrix} \tag{5.15}$$

Table 5.3: FPGA synthesis results and comparison for hardware resources utilization. (A)LUT: (Adaptive) Look Up Table, FFs: Flip Flops, Logic Reg: Logic Registers, LE: Logic Element, DSPB: DSP Block, Mult: Multiplier, Max Freq: Maximum Frequency.

Model	(A)LUT	LFFs	Logic Reg/slice	LE	Pins	DSPB/ Mult	Max Freq (MHz)
Single HR [18]	659	431	412	-	64	0	81
Coupled HR [12]	-	-	4851	-	-	104 Mult	14.19
Coupled HR [19]	-	-	8925	-	-	96 Mult	3.54
Coupled Komendantov–Kononenko [12]	-	-	533	27872	8	0	17.22
proposed DHR	5862	-	210	-	36	0	-
Coupled DHR							
Pipelined 4 DHR							

5.6 Proposed Platform for Test and Measurement

The platform is displayed in Fig. 5.8. As seen in this figure, it is composed of an FPGA board which is used for hardware implementation, a NI PXI device for measurement and LabVIEW program to demonstrate the output pattern. The circuits are developed in Verilog Hardware Description Language (VHDL) and have been implemented on Altera Stratix IV (EP4SGX530KH) FPGA development board. The RTL design have been synthesized using the Altera compiler, Quartus Prime. The results of the hardware utilization are given in Table. 5.3.

Static Timing Analysis (STA) has been performed using TimeQuest timing analysis tool in Quartus Prime and Synapse Design Constraints (SDC) in Altera Compiler. Based on the timing analysis, a maximum frequency of 98.43 MHz was achieved for the proposed architecture. A comparison is made between the proposed model and the available works in the literature in Table II. Ass seen in this table the proposed work offers affordable hardware cost while working in higher maximum frequency.

Fig. 5.9 demonstrates the anti-phase patterns for walking gate generations. As expected based on Table II $x2$, and $x3$ are in the same phase while in opposite phase with $x1$, and $x4$.

5.7 Conclusion

This paper presented digital hardware implementation for a CPG based on HR spiking neuron model on an FPGA-embedded system providing efficient and flexible platform to generate rhythmic patterns suitable for mobile robotic applications. A specialized digital circuit is proposed for HR spiking model which requires less resources for hardware implementation in comparison with similar works proposed in the literature. The proposed digital design is validated by a timing error measurement and dynamic analysis. For the first time, this analysis is performed with fixed-point representation to validate the digital model and the results indicate that the proposed digital model is successfully generates the original model pattern in the output. Furthermore, dynamic behavior of coupled neurons was verified for different coupling states including synchronous and asynchronous tonic and bursting. The proposed digital model is compared with the state of the art methodologies, and it indicates less resources while presenting higher accuracy. DHR neurons are coupled through a coupling function to form a CPG block. An architecture is proposed for the CPG composed of four DHR to generate biped pattern. An RTL design is performed to implement the architecture and the experimental results confirms that the proposed digital architecture is able to successfully reproduce the anti–phase pattern in the output of the CPG. The FPGA device provides an integrated platform for developing robot control applications. The proposed model takes maximum 5 percent of the available resources of Stratix IV Altera FPGA board (EP4SGX530KH). STA has been performed using Quartus Prime compiler and a maximum frequency of 98.43 MHz is provided. Extra topologies for CPG can be analyzed and different patterns can be generated in order to apply the system in various applications and it can be used for modeling different patterns such as walking, running, and jumping by adjusting the coupling weights. The system can be employed as a neuro core to couple higher number of neurons for different applications.

References

[1] S. L. Hooper, “Central pattern generators,” Current Biology, vol. 10, no. 5, R176-R179, 2000.

[2] D. Zhang, Q. Zhang, and X. Zhu, “Exploring a type of central pattern generator based on Hindmarsh–Rose model: from theory to application,” International journal of neural systems, vol. 25, no. 01, p. 1450028, 2015.

[3] E. Donati, G. Indiveri, and C. Stefanini, “A novel spiking CPG-based implementation system to control a lamprey robot,” 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp. 1364-1364, 2016.

[4] E. M. Izhikevich, “Simple model of spiking neurons,” IEEE Transactions on neural networks, vol. 14, no. 6, pp. 1569-1572, 2003.

[5] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” J. Physiol., vol. 117, no. 4, pp. 500–544, Aug. 1952.

[6] C. Morris and H. Lecar, “Voltage oscillations in the barnacle giant muscle fiber,” Biophys. J., vol. 35, pp. 193–213, 1981.

[7] R. Jolivet, A. Rauch, H. R. L. Lüscher, and W. Gerstner, “Integrateand-Fire models with adaptation are good enough: Predicting spike times under random current injection,” Advances in neural information processing systems, pp. 595–602, 2005.

[8] M. G. F. Fuortes and F. Mantegazzini, “Interpretation of the repetitive firing of nerve cells,”, The Journal of general physiology, vol. 45, no. 6, pp. 1163–1179, Jul. 1962.

[9] J. L. Hindmarsh, and R. M. Rose, “A model of neuronal bursting using three coupled first order differential equations,” Proc. R. Soc. Lond. B, vol. 221, no. 1222, pp. 87-102, 1984.

[10] N. Gomar, B. Moradi, M. Ahmadi, “Digital Hardware Implementation of a Biological Central Pattern Generator”, Midwest Symposium, 2018.

[11] N. Korkmaz, İ. Öztürk, and R. Kılıç, “The investigation of chemical coupling in a HR neuron model with reconfigurable implementations.” Nonlinear Dynamics, vol. 86, no, pp.1841-1854, 2016.

[12] Q. Chen, J. Wang, S. Yang, Y. Qin, B. Deng, and X. Wei, “A real-time FPGA implementation of a biologically inspired central pattern generator network,” Neurocomputing, vol. 244, pp. 63-80, 2017.

[13] C. Mead, “Neuromorphic electronic systems,” Proceedings of the IEEE, vol. 78, no. 10, pp. 1629-1636, 1990.

[14] S. Gomar, and A. Ahmadi, “Digital multiplierless implementation of biological adaptive-exponential neuron model,” IEEE Transactions on Circuits and Systems I Regular Papers, vol. 61, no. 4, pp.1206-1219, 2014.

[15] L. Camuas-Mesa, A. Acosta-Jimnez, C. Zamarreo-Ramos, T. Serrano- Gotarredona, and B. Linares-Barranco, “A 32x32 Pixel Convolution Processor Chip for Address Event Vision Sensors with 155ns Event Latency and 20Meps Throughput,” Transactions on Circuits and Systems I , vol. 58, No. 4, pp. 777-790, 2011.

[16] M. Ambroise, T. Levi, S. Joucla, B. Yvert, and S. Saıghi, “Real time biomimetic central pattern generators in an FPGA for hybrid experiments,” Frontiers in neuroscience, vol. 7, p. 215, 2013.

[17] J. H. Barron-Zambrano, and C. Torres-Huitzil, “FPGA implementation of a configurable neuromorphic CPG-based locomotion controller,” Neural Networks, vol. 45, pp. 50-61, 2013.

[18] M. Hayati, M. Nouri, D. Abbott, and S. Haghiri, “Digital multiplierless realization of two-coupled biological Hindmarsh–Rose neuron model,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 5, pp. 463-467, 2016.

[19] N. Dahasert, İ. Öztürk, and R. Kiliç, “Experimental realizations of the HR neuron model with programmable hardware and synchronization applications,” Nonlinear Dynamics, vol. 70, no. 4, pp. 2343-2358 , 2012.

Chapter 6

Conclusion and Future Works

6.1 Summary of Contributions

In this dissertation, it is mainly focused on developing configurable digital architectures on FPGA for several neuromorphic building blocks including astrocyte, STDP learning rule, Wilson-Cowan cortex model, and a CPG based on HR spiking neuron.

In chapter 2, a digital architecture was presented for astrocyte which regulates the activity of the coupled spiking neurons. A network of two neurons and an astrocyte has been developed based on Postnov astrocyte and AdEx neuron model. The equations have been modified for an efficient hardware implementation. Simulations have shown that Tanh function can be approximated using sign function with an acceptable accuracy and similarity with the original function, therefore sign function has been used for equation approximation. In order to reduce the hardware cost, shift and add have been used instead of multiplication and division in some cases where constant multiplication or division is needed. Synthesis results verify that the proposed design uses less than 1% of available resources of a VIRTEX II FPGA, so it can be used as a module in a large scale SNN implementation. On the other hand, since astrocyte is able to regulate the synaptic weight, it can increase the reliability of the system in neural network applications.

In chapter 3, digital hardware circuits were proposed and compared for STDP learning rules. The modified base-2 method with an efficient hardware provides a tradeoff between accuracy and configurability for the system. The maximum error was reported as 0.0088 for 8-bit data-path and 0.0014 for 16-bit data path. In addition, the hardware cost is not highly dependent on the data-path bit-width. Digital circuits have been synthesized and implemented on FPGA, and the experimental results showed that the models are capable of producing the learning window, successfully. The models were compared with the LUT and linear approximations. Although LUT model provides the highest accuracy the hardware requirement is high and highly dependent on the accuracy and it does not provide configurability. PWL method has lower hardware cost, but it provides the least accuracy. A modular pair of pre-postsynaptic neuron have been implemented and STDP could successfully be applied to the coupled neurons. The hardware cost for the proposed learning model architectures was obtained less than 1% of the FPGA resources, and the maximum frequency was gained as 250 MHz. The model developed in this work can be employed as a module for a larger spiking neural network.

In chapter 4, the Gaussian Wilson Cowan model as one of the accurate and well-known models which represents cortex behavior has been implemented for the first time. A digital model for hardware implementation of the model has been proposed and compared with the original model. The model has been proven to be able to follow the original model in both dynamic behavior and timing analysis, accurately. The mathematical analysis shows that the digitized models are able to reproduce the same bifurcations and dynamic behavior. Time domain analysis indicates that the model shows up to 97% similarity. RMSE and maximum error have been measured as minimum as 0.0085, 0.0268, on average respectively. In the next step, digital architecture was designed for the proposed model and compared with PWL method in terms of the number of resources they require and the accuracy they provide. There is always a tradeoff between the accuracy and the hardware cost. DWC-4M implements the model using 4 multipliers, and provides higher accuracy with a lower speed, while the DWC-2M model represents the model using two multipliers, lower accuracy, and higher speed. In the final step, the digital model was compiled and synthesized in Quartus II Altera compiler and physically implemented on the DE IV Stratix FPGA board. Maximum frequency was obtained as 238.32 and 244.38 for the DWC- 4M, and DWC-2M, respectively. The hardware cost reports show that DWC-4M takes maximum 2% of the FPGA board for a single pair of EI implementation, which means a number of 50 pairs can be implemented on the board. With keeping the IO pins to the minimum, this number for DWC-2M can be up to 100. To implement a higher number of neurons on the board, one possibility is implementing virtual neurons using time-multiplexing, and resource sharing techniques. A time-multiplexing design has been performed for the model. A linear coupled population of the DWC was performed in this work which can be extended into a two-dimensional implementation for representing a complete DWC-based model of neocortex.

Chapter 5 presents digital hardware implementation for a CPG based on HR spiking neuron model on an FPGA-embedded system providing an efficient and flexible platform to generate rhythmic patterns suitable for mobile robotic applications. A specialized digital circuit is proposed for HR spiking model which requires fewer resources for hardware implementation in comparison with similar works proposed in the literature. The proposed digital design is validated by a timing error measurement and dynamic analysis. For the first time, this analysis is performed with a fixed-point representation to validate the digital model and the results indicate that the proposed digital model successfully generates the original model pattern in the output. Furthermore, the dynamic behavior of coupled neurons was verified for different coupling states including synchronous and asynchronous tonic and bursting. The proposed digital model is compared with the state of the art methodologies, and it indicates fewer resources while presenting higher accuracy. DHR neurons are coupled through a coupling function to form a CPG block. An architecture is proposed for the CPG composed of four DHR to generate biped pattern. An RTL design is performed to implement the architecture and the experimental results confirm that the proposed digital architecture is able to successfully reproduce the anti-phase pattern in the output of the CPG. The FPGA device provides an integrated platform for developing robot control applications. The proposed model takes maximum 3 percent of the available resources of Stratix IV Altera FPGA board (EP4SGX530KH). STA has been performed using Quartus Prime compiler and a maximum frequency of 139.53 MHz is provided. Extra topologies for CPG can be analyzed and different patterns can be generated in order to apply the system in various applications and it can be used for modeling different patterns such as walking, running, and jumping by adjusting the coupling weights. The system can be employed as a neuro core to couple a higher number of neurons for different applications.

6.2 Suggested Future Works

This dissertation can be continued in two fields as below:

- Central Pattern Generators (CPGs) are neural circuits which generate rhythmic patterns to control the locomotion parts of both vertebrate and invertebrates. Accordingly, CPGs are made of coupled spiking neurons with adjustable weight connections between the neurons. In this dissertation, digital circuits were proposed for Wilson Cowan neuron model which can be used as a block of CPG. In addition, a bio-inspired digital CPG based on HR model was designed to produce rhythmic patterns. An embedded implementations including all these neuromorphic building blocks can be developed for real world applications as either a bio-robotic control system or a rehabilitation system to help people with injuries. The developed bio-inspired rehabilitation systems can be utilized to stimulate the nervous system of people with locomotor disabilities.

- The NE research is growing rapidly in a variety of fields including image processing, signal processing, and object tracking. An embedded system composed of Spiking neurons such as HR neuron which was designed in this thesis along with a learning algorithm such as STDP can be developed for a pattern recognition or image processing task. The aim of this project is to integrate all neuromorphic blocks to build a configurable FPGA based neuromorphic system composed of spiking neurons, learning block, and vision sensor for an image processing task for various applications such as vehicle navigation system.

VITA AUCTORIS

Shaghyegh (Rose) Gomar was born in 1988 in Kermanshah, Iran. She received her Bachelor and Master degrees in Electrical Engineering from Razi University in 2011 and 2013. She worked as lecturer in the Electrical Engineering Department of Razi University and Educational Institutions in 2013-2015. In 2015 she was awarded the International Ontario Trillium Scholarship (OTS) from the University of Windsor to continue her Ph.D. studies. Since 2015, she has been pursing her Ph.D in the field of Neuromorphic Engineering in Electrical and Computer Engineering department of the University of Windsor. Her research interests are Spiking Neural Network (SNN), FPGA design, Digital design, Computer Arithmetic, Neuromorphic Engineering, and parallel processing. She is also an IEEE member and she served as Vice-Chair and Chair of IEEE Women In Engineering (WIE) of Windsor section.