

# A Digital Neuromorphic Realization of the 2-D Wilson Neuron Model

Moslem Nouri<sup>1</sup>, Mohsen Hayati<sup>1</sup>, Teresa Serrano-Gotarredona<sup>2</sup>, *Member, IEEE*,  
and Derek Abbott<sup>3</sup>, *Fellow, IEEE*

**Abstract**—This brief presents a piecewise linear approximation of the nonlinear Wilson (NW) neuron model for the realization of an efficient digital circuit implementation. The accuracy of the proposed piecewise Wilson (PW) model is examined by calculating time domain signal shaping errors. Furthermore, bifurcation analyses demonstrate that the approximation follows the same bifurcation pattern as the NW model. As a proof of concept, both models are hardware synthesized and implemented on field programmable gate arrays, demonstrating that the PW model has a range of neuronal behaviors similar to the NW model with considerably higher computational performance and a lower hardware overhead. This approach can be used in hardware-based large scale biological neural network simulations and behavioral studies. The mean normalized root mean square error and maximum absolute error of the PW model are 6.32% and 0.31%, respectively, as compared to the NW model.

**Index Terms**—Nonlinear Wilson (NW) neuron model, piecewise Wilson (PW) neuron model, field programmable gate array (FPGA), neuromorphic circuits.

## I. INTRODUCTION

**T**O DATE, various models with diverse levels of biological detail have been developed to mimic biological neurons [1]–[12]. These models are typically presented in the form of differential equations that capture the dynamics of biological process in the neuron. Biologically detailed models describe cellular phenomena, using physiological parameters and properties of the individual components. Unfortunately, such models are complex and computationally intensive for large scale simulation and/or implementation. Abstract models, on the other hand, present higher level descriptions with fewer biological details and computationally are preferred for computational numerical and behavioral study in the areas of network dynamics, neural data coding, and memory.

The three-dimensional HR model was first introduced in 1989 by Rose and Hindmarsh [7]. The model can be seen either as an extension of the FitzHugh-Nagumo equations or as a simplification of the physiologically practical model proposed by Hodgkin and Huxley (HH) model [8]. The FitzHugh-Nagumo neuron model cannot reproduce significant biological signal behavior such as: tonic bursting, spike frequency adaptation, integration, and rebound bursting in comparison to the Wilson neuron model. The nonlinear Wilson (NW) neuron model has been proposed, which can reproduce many neuronal behaviors [9]. Moreover, spiking-based models such as the Integrate and Fire (IF), Izhikevich, and the Leaky-Integrate and Fire (LIF) models, do not accurately mimic neurons in the living nervous system [11]. The NW model is considered here because of its ability to mimic nearly all the behaviors presented by real biological neurons with greater simplicity compared with the HH model.

In general, reconfigurable digital platforms, i.e., Field Programmable Gate Arrays (FPGA), are low-cost, and widely enable the possibility of implementing stable and reliable large scale networks [11]. To take full advantage of this approach, it is necessary to develop a rich library of different neural network building blocks and primitives for use in high level synthesizers for large scale SNNs. This brief can be considered as an effort to assist the development of such building blocks. It is important to have simple and accurate models when using FPGAs for implementation and simulation of large networks of spiking neurons because of their limited resources. Several studies have already been reported regarding simulation of networks of spiking neurons on FPGA [1]–[6], [10]–[12]. In a hardware approach the number of available or synthesizable functional units is one of the major limitations of implementing large scale networks on FPGAs. For instance, 32 Leaky-Integrate and Fire (LIF) neurons have been implemented because only 32 multipliers were available in the utilized FPGA platform [10]. Note that in the case of [10], the neuron model is much simpler than the NW neuron model and unable to reproduce all the required biological behaviors. Since the NW neuron model has a large hardware overhead due to its use of nonlinear functions, optimization of the model for the realization of a large number of neurons is significant. Hence, the high-speed multiplierless design of the NW model is significant. Motivated by high-speed design, reduced hardware overhead, and an efficient implementation of a NW neuron model as a principal component for investigations with larger neuron populations, our work

M. Nouri and M. Hayati are with the Department of Electrical Engineering, Faculty of Engineering, Razi University, Kermanshah 67149, Iran (e-mail: moslemnouri70@gmail.com; mohsen\_hayati@yahoo.com).

T. Serrano-Gotarredona is with the Instituto de Microelectrónica de Sevilla, IMSE-CNM (CSIC and Universidad de Sevilla), 41092 Sevilla, Spain (e-mail: terese@imse-cnm.csic.es).

D. Abbott is with the School of Electrical and Electronic Engineering, University of Adelaide, Adelaide, SA 5005, Australia (e-mail: derek.abbott@adelaide.edu.au).

demonstrates a significantly simplified realization of the NW neuron model. To circumvent hardware-intensive functional units, we present a set of piecewise linear models that can be efficiently implemented in large numbers on different digital platforms including FPGAs. Our proposed Wilson (PW) model shows similar behavior to the NW model with lower implementation overhead. The procedure used in this brief can also be used for implementation of other information processing units in the brain such as astrocytes and synapse models [6]. In this brief the NW neuron model is used to present an efficient building block for the simulation and physical realization of SNNs. The remainder of this article is organized as follows. Section II presents the background of NW neuron model. The dynamical analysis is explained in Section III. Section IV presents the implementation results. Section V concludes this brief.

## II. BACKGROUND

This model has been developed by polynomial curve fitting, based on the Hodgkin-Huxley neuron model [9]. The most straightforward form of the NW model is a two-dimensional dynamical model, where the state is determined by two variables: (i) the membrane potential and (ii) the recovery variable. The model comprises two differential equations [9]:

$$\begin{cases} c \frac{dV}{dt} = I_{\text{apply}} - (f(V) + g(V, R)) \\ \frac{dR}{dt} = \frac{1}{\tau}(-R + 1.35V + 1.03) \end{cases} \quad (1)$$

where

$$\begin{cases} f(V) = (17.81 + 47.71V + 32.63V^2)(V - 0.55) \\ g(V, R) = 26R(V + 0.92). \end{cases} \quad (2)$$

Here,  $V$  in mV denotes the membrane potential and  $R$  determines the recovery variable, which is dimensionless. Via use of scaling parameters and for different levels of the stimulus current, which is in  $\mu\text{A}$ , the model can reproduce different types of spiking patterns [9].

### A. Piecewise Wilson (PW) Model

In order to achieve improved computational speed of the model and low hardware overhead, we modify the NW model. The main motivation of this modification is to significantly reduce the number of digital gates required for an implementation.

The original equations of the 2D NW model can be rewritten as

$$c \frac{dV}{dt} = I_{\text{apply}} - (q(V) + h(V, R)) \quad (3)$$

$$\frac{dR}{dt} = \frac{1}{\tau}(-R + 1.35V + 1.03) \quad (4)$$

where

$$q(V) = -9.8 - 8.43V + 30p(V) \quad (5)$$

$$h(V, R) = 13 \cos(R - (V + 0.92)) - 13 \cos(R + (V + 0.92)) \quad (6)$$

where

$$p(V) = \begin{cases} a_0|V| + b_0; & V_1 < V < V_2 \\ a_1|V| + b_1; & V_2 < V < V_3 \\ a_2|V| + b_2; & V_3 < V < V_4 \\ a_3|V| + b_3; & V_4 < V < V_5 \\ a_4|V| + b_4; & \text{else} \end{cases} \quad (7)$$

and where  $a_i$ ,  $0 < i < 4$  are the slopes of piecewise linear segments in the PWL5 approximation, which are listed in Table I. Consequently, the nonlinear polynomial term is replaced by a piecewise linear approximation, which facilitates a significance increase in computational efficiency. As depicted in Fig. 1, this PWL5 function is well-matched with the nonlinear function.

An exhaustive search algorithm is applied to the parameters to find their values with an improved precision. This algorithm searches for the best parameters among a set of solutions and determines the closest answer with minimum error. For a given function  $f(x)$  defined on the interval  $[a, b]$  the goal is finding the approximation  $p(x)$  in the form of  $p(x) = \alpha x + \beta$  such that  $|f(x) - p(x)| \leq \epsilon$  for every  $x$  in  $a \leq x \leq b$ . The algorithm begins by selecting the longest possible single line, which satisfies the error restriction and repeats the process by selecting lines of the maximum length from the left-hand side towards remaining of the function to be approximated. It is assumed that discontinuities are allowed at the boundaries between the linear segments. The number of linear segments depends on the threshold error. For example, in the case of 5-5PWL, if the linear segments go over 5 lines, the threshold error will be increased so that number of lines decreases and finally it fits to 5 lines. Since the aim is the implementation on a digital computer, the function  $f(x)$  should be defined only on a discrete set of points. Therefore, the problem can be restated as:

For a given set of real numbers  $x(0), x(1), x(2), \alpha, \beta \in \mathbb{R}$  and  $n \in \mathbb{N}$ , such that  $|f(j) - (\alpha x(j) + \beta)| \leq \epsilon$  for  $0 \leq j \leq n$  and  $n$  is maximal. The algorithm is shown in Algorithm 1. In line 2 and line 3, initial conditions for the parameters and domain of the function to be approximated  $f(x)$  and range of the coefficients to be searched  $[a, b]$  are assigned. The loop in line 5 for  $L$  point checks the error between original and approximation function and if it goes over the threshold error, breaks the loop and compares maximum  $x$  with maximum  $x$  of the previous combinations of  $\alpha$  and  $\beta$  (in order to find the longest line) and if it is the largest yet,  $\alpha$  and  $\beta$  will be stored. The search is restarted by choosing  $x_m$ , the maximum  $x$  which can imply the relation  $|f(x) - p(x)| \leq \epsilon$ , of longest line as start of new line. This process is continued until it reaches the end of the function domain. In the line 17, the number of linear segments are compared with a preconditioned number of segments and if it goes over, the function cannot be approximated with this threshold and number of lines and the threshold error will be increased and the loop continues until the desired results are obtained.

The mean normalized root mean square error (NRMSE) and maximum absolute error (MAE) [6] of the PW model are 6.32% and 0.31%, respectively, as compared to the NW model.

**Algorithm 1** The Pseudo Code of the Used Exhaustive Search Algorithm in Finding the Coefficients

```

1: //assign initial values for  $\alpha$ ,  $\beta$ , and other variables
2: nlines  $\leftarrow$  0;  $x_m \leftarrow$  0;  $dx(j) \leftarrow$  0.001;
3:  $\Delta \leftarrow$  0.001;  $x_{\text{start}} \leftarrow$   $x_1$ ;
4: s;
5: for L point do
6:    $P \leftarrow \alpha x(j) + \beta$ ;  $F \leftarrow f(x(j))$ ;  $x(j) \leftarrow x(j) + dx(j)$ ;
7:   if  $\text{abs}(F - P) >$  predefined maximum error then,
8:     goto line 10;
9:   //check that it is the longest line or not
10:  if  $x(j) >$   $x_m$  then
11:     $x_m \leftarrow x(j)$ ;
12:    store  $\alpha$ ,  $\beta$ ;
13:  //check other combinations
14:   $\alpha \leftarrow \alpha + \Delta$ ;  $x(j) \leftarrow x_{\text{start}}$ ;
15:   $\beta \leftarrow \beta + \Delta$ ;
16:   $x_{\text{start}} \leftarrow x_m$ ; nlines++;
17:  if nlines  $<$  5 then
18:    goto s;
19:  else
20:    reset all variables;

```

TABLE I  
MODIFIED COEFFICIENTS FOR THE PW MODEL

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$
-0.50	0.02	0.30	0.25	1.00
$b_0$	$b_1$	$b_2$	$b_3$	$b_4$
0.50	0.14	-0.02	0.00	-0.15
$V_1$	$V_2$	$V_3$	$V_4$	$V_5$
-1.00	-0.70	-0.60	0.00	0.20

The multiplication operation of the membrane potential variable  $V$  and of the recovery variable  $R$ , in our PW model, has been eliminated and substituted by a cosine operation plus a small number of additions and subtractions as can be observed in the formulation of  $h(V, R)$  in Equation (6). The cosine function can be implemented in a FPGA using a simple LUT module, resulting in a low cost model as compared to the NW model due to our multiplierless formulation. A cosine function since it is periodic, can be easily implemented with lookup tables by storing one cycle of the function into the memory and mapping the input as an address for fetching the corresponding output.

### III. DYNAMICAL ANALYSIS

#### A. Dynamics of a Single NW Neuron

The role of the two nullclines are of significance in describing the transition between resting state and spiking state (bifurcation) [11]. It will be convenient to write the NW model as

$$\begin{cases} \frac{dV}{dt} = x(V, R) \\ \frac{dR}{dt} = y(V, R). \end{cases} \quad (8)$$

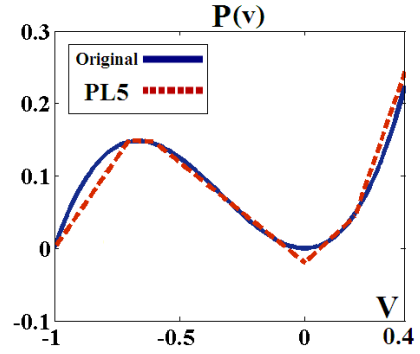


Fig. 1. The difference between the original nonlinear function,  $p(V)=(1+V)V^2$  and the PWL5 approximation.

The nullclines of the NW model are given as

$$\frac{dV}{dt} = \frac{dR}{dt} = 0 \Rightarrow x(V, R) = y(V, R) = 0 \quad (9)$$

$$R = \frac{-9.8 - 8.43V + 29.77V^2 + 32.63V^3 - I_{\text{app}}}{-26(V + 0.92)} \quad (10)$$

$$R = 1.35V + 1.03. \quad (11)$$

Equilibrium points occur at the intersection of two nullclines [11].

The Jacobian matrix and eigenvalues are required for analyzing the equilibrium points using a dynamical bifurcation analysis, [6] and the Jacobian matrix can be obtained as follows:

$$J(V, R) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (12)$$

where

$$\begin{cases} A = \frac{\partial x(V, R)}{\partial V}, & B = \frac{\partial x(V, R)}{\partial R} \\ C = \frac{\partial y(V, R)}{\partial V}, & D = \frac{\partial y(V, R)}{\partial R} \end{cases} \quad (13)$$

$$\begin{cases} A = \frac{8.43 - 59.54V - 97.89V^2 + 26R}{c} \\ B = \frac{-26(V + 0.92)}{c} \\ C = \frac{1.35}{\tau}, & D = \frac{-1}{\tau}. \end{cases} \quad (14)$$

In this case, equilibrium point stability is determined by the trace of  $J(V, R)$  [6]. Bifurcation theory explains how solutions vary as a function of the model parameters [6]. Using a bifurcation analysis, classify transition types that arise as a function of the parameters that we vary. Specifically, we are able to predict the magnitude of  $I_{\text{apply}}$  at which the equilibrium point loses its stability and oscillations appear.

In this way, for the PW model, the Jacobian matrix can be obtained as

$$J(V, R)_{\text{PW}} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (15)$$

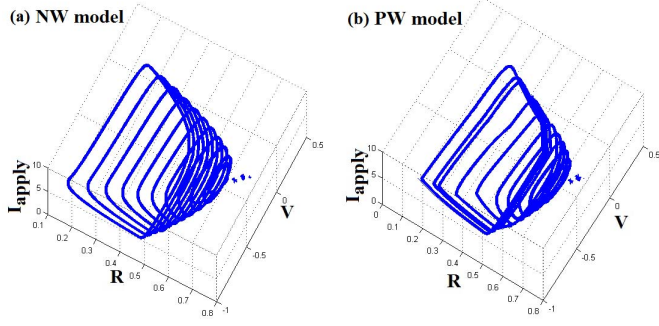


Fig. 2. The 3D phase portraits of the NW and PW models with many limit cycles with different fixed  $I_{\text{apply}}$  via  $0.5 \mu\text{A}$  step size, which represent supercritical Hopf bifurcations at  $\tau = 5.6$  and  $C = 1 \mu\text{F}/\text{cm}^2$ . Here,  $V$  represents the membrane potential,  $R$  denotes the recovery variable, and  $I_{\text{apply}}$  presents the input stimulus current. (a) The 3D phase portraits of the NW model with limit cycles for fourteen fixed input stimulus current samples. (b) The 3D phase portraits of the PW model with fourteen limit cycles for fixed input stimulus current samples.

$$\begin{cases} A = \frac{1}{c}(-\bar{f}(V) - \partial h(V, R)/\partial V) \\ B = 13 \sin(R + (V + 0.92)) - 13 \sin(R - (V + 0.92)) \\ C = \frac{1.35}{\tau}, \quad D = \frac{-1}{\tau} \end{cases} \quad (16)$$

where

$$\bar{f}(V) = -8.43 + 30\bar{g}(V) \quad (17)$$

$$\begin{aligned} \partial h(V, R)/\partial V &= 13 \sin(R - (V + 0.92)) \\ &+ 13 \sin(R + (V + 0.92)) \end{aligned} \quad (18)$$

where

$$\bar{g}(V) = \begin{cases} \frac{a_0}{V}, & V_1 < V < V_2 \\ \frac{a_1}{V}, & V_2 < V < V_3 \\ \frac{a_2}{V}, & V_3 < V < V_4 \\ \frac{a_3}{V}, & V_4 < V < V_5 \\ \frac{a_4}{V}, & \text{else.} \end{cases} \quad (19)$$

The Hopf bifurcation is the mechanism through which one can go from a stable equilibrium point to an oscillation [11]. For the Hopf regime, the bifurcation diagram can be described by a variation in  $I_{\text{apply}}$ , within the Wilson equations. The NW and PW models undergo some oscillations when the input stimulus current increased over  $0.34 \mu\text{A}$ . Therefore, both models have supercritical Hopf bifurcations for  $I_{\text{apply}}=0.34 \mu\text{A}$ . For further clarification of the effect of the stimulus current changes on the dynamical behaviors of the systems, phase portraits of the PW and NW models with many limit cycles are plotted versus the fixed input stimulus currents in Fig. 2, which shows a supercritical Hopf bifurcation [9], [13], [14] here for both Nonlinear Wilson (NW) and Proposed Wilson (PW) models.

The pipelined digital neuromorphic circuits to implement the NW and PW models are shown in Fig. 3. The variables,  $V$  and  $R$ , are performed in pipeline structures with  $V_s$  and  $R_s$  stages, where  $V_{\text{buffer}}$  and  $R_{\text{buffer}}$  are the buffer registers for the  $V$  and  $R$  values and  $V_{\text{buffer-size}}$ , and  $R_{\text{buffer-size}}$  are the sizes of these connected buffers, respectively. The hardware comprises simple LUT modules, Sub or Add, and logic shift operations

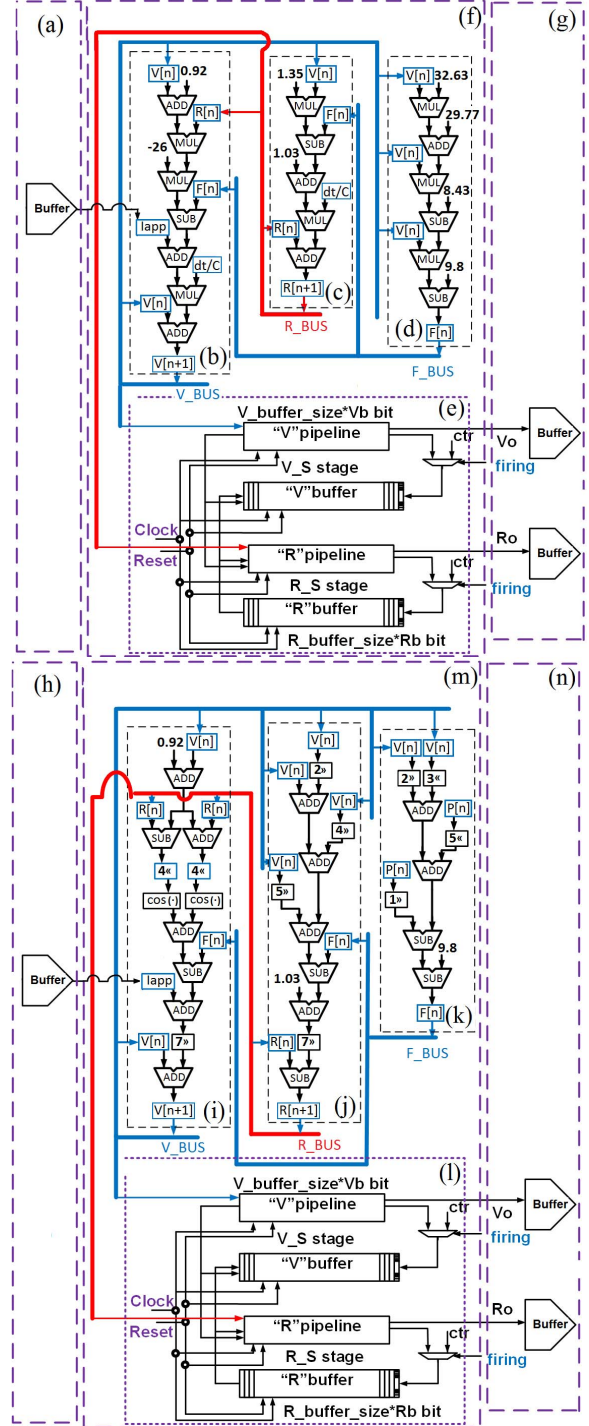


Fig. 3. The pipelined digital neuromorphic circuits to implement the NW and PW models. (a) The input buffer unit of the NW model. (b)  $V$  pipeline in NW model. (c)  $R$  pipeline in NW model. (d)  $F$  pipeline in NW model. (e) Arithmetic pipelines. (f) The NW neuron model. (g) The output buffer unit of the NW model. (h) The input buffer unit of the PW model. (i)  $V$  pipeline in PW model. (j)  $R$  pipeline in PW model. (k)  $F$  pipeline in PW model. (l) Arithmetic pipelines. (m) The PW neuron model. (n) The output buffer unit of the PW model. Abbreviations:  $V_{\text{Data\_Bus}}$  ( $V_{\text{BUS}}$ ),  $R_{\text{Data\_Bus}}$  ( $R_{\text{BUS}}$ ),  $F_{\text{Data\_Bus}}$  ( $F_{\text{BUS}}$ ), Adder (ADD), Subtractor (SUB), Multiplier (MUL), Cosine ( $\cos(\cdot)$ ), “ $7\gg$ ” indicates that the input receives seven shifts to the right and “ $3\ll$ ” means three shifts to the left.

that replace high-cost multipliers. Hence, the hardware computational overhead is significantly reduced. As the first step towards digital implementation, it is necessary to discretize



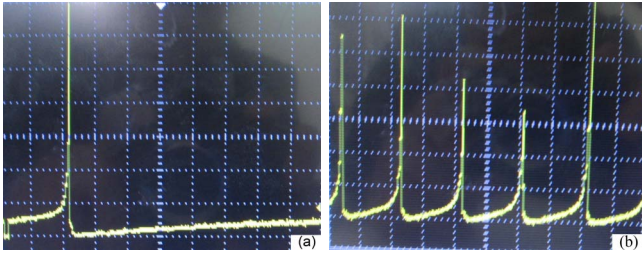


Fig. 4. Output of the PW model implemented on XILINX Virtex-II Pro XC2VP30. (a) Spike Latency. (b) Tonic spiking. The horizontal axis denotes time (time scale = 2.5 ms), and voltage is displayed on the vertical axis (voltage scale = 10 mV).

TABLE II

UTILIZATION OF DEVICES ON THE XILINX VIRTEX-II PRO.  
 ABBREVIATIONS: COMBINATIONAL MULTIPLIER (COM. MUL.), FULL PIPELINED MULTIPLIER (F. PIPE. MUL.), AND MULTIPLIER (MUL.)

Resource	NW model Com. Mul.	NW model F. Pipe. Mul.	NW model Both Mul.	PW model
Slices	288	862	340	241
4LUTs	1178	1011	667	617
Frequency	21 MHz	191.70 MHz	12.45 MHz	191.81 MHz

the equations. We achieve this process of discretization by exploiting the Euler method.

#### IV. IMPLEMENTATION RESULTS

The modified model is implemented on a XILINX XUP VirtexII Pro (XC2VP30) Development System to validate the PW neuron model. Fig. 4 contains oscilloscope photographs that illustrate the dynamical behaviors of our modified model realized on this FPGA platform. The device utilization for realization of a single neuron based on the NW and the PW models are summarized in Table II. Fig. 4 demonstrates the output of the PW model realized on a XILINX Virtex-II Pro XC2VP30. The results show that the implemented neuron model is readily deployable compared with the NW model, due to circumventing the need for computationally intensive multiplier operations. In the PW neuron model, the nonlinear polynomial terms in the NW model have been replaced by the piece-wise linear model that can be implemented in a multiplierless hardware design.

#### V. CONCLUSION

In this brief a set of piece-wise linear approximations nonlinear Wilson (NW) model is presented. The simulation and

implementation results demonstrate that the proposed modification follows the same dynamical behavior in the time and phase domains as the NW model. This model only requires arithmetic add and shift operations, it can be implemented on FPGA without using an embedded multiplier, which can limit the number of neurons on a chip. The presented model is considerably faster and less area consuming than the NW model. The mean normalized root mean square error (NRMSE) and maximum absolute error (MAE) of the PW model are 6.32% and 0.31%, respectively, as compared to the NW model.

#### REFERENCES

- [1] H. Soleimani and E. M. Drakakis, "An efficient and reconfigurable synchronous neuron model," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 1, pp. 91–95, Jan. 2018, doi: [10.1109/TCSII.2017.2697826](https://doi.org/10.1109/TCSII.2017.2697826).
- [2] M. Hayati, M. Nouri, D. Abbott, and S. Haghiri, "Digital multiplierless realization of two-coupled biological Hindmarsh–Rose neuron model," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 5, pp. 463–467, May 2016.
- [3] M. Nouri, M. Jalilian, M. Hayati, and D. Abbott, "A digital neuromorphic realization of pair-based and triplet-based spike-timing-dependent synaptic plasticity," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 6, pp. 804–808, Jun. 2018.
- [4] E. Jokar and H. Soleimani, "Digital multiplierless realization of a calcium-based plasticity model," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 7, pp. 832–836, Jul. 2017.
- [5] T. Matsubara, H. Torikai, and T. Hishiki, "A generalized rotate-and-fire digital spiking neuron model and its on-FPGA learning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 10, pp. 677–681, Oct. 2011.
- [6] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "A digital realization of astrocyte and neural glial interactions," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 2, pp. 518–529, Apr. 2016.
- [7] R. M. Rose and J. L. Hindmarsh, "The assembly of ionic currents in a thalamic neuron. I. The three-dimensional model," *Proc. Roy. Soc. London B Biol. Sci.*, vol. 237, no. 1288, pp. 267–288, Aug. 1989.
- [8] 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, 1952.
- [9] H. R. Wilson, *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience*. Oxford, U.K.: Oxford Univ. Press, 1999.
- [10] A. Cassidy, S. Denham, P. Kanold, and A. Andreou, "FPGA based silicon spiking neural array," in *Proc. Biomed. Circuits Syst. Conf. (BIOCAS)*, Montreal, QC, Canada, Nov. 2007, pp. 75–78.
- [11] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "Digital multiplierless realization of two coupled biological Morris–Lecar neuron model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 7, pp. 1805–1814, Jul. 2015.
- [12] M. Nouri, G. R. Karimi, A. Ahmadi, and D. Abbott, "Digital multiplierless implementation of the biological FitzHugh–Nagumo model," *Neurocomputing*, vol. 165, pp. 468–476, May 2015.
- [13] E. M. Izhikevich, *Dynamical Systems in Neuroscience*, Cambridge, MA, USA: MIT Press, 2007.
- [14] G. Q. Zhong, K. S. Tang, G. Chen, and K. F. Man, "Bifurcation analysis and circuit implementation of a simple chaos generator," *Latin Amer. Appl. Res.*, vol. 31, no. 3, pp. 227–232, 2001.