

Digital Multiplierless Realization of Two Coupled Biological Morris-Lecar Neuron Model

Mohsen Hayati, Moslem Nouri, Saeed Haghir, and Derek Abbott, *Fellow, IEEE*

Abstract—Modeling and implementation of biological neural networks are significant objectives of the neuromorphic research field. In this field, neuronal synchronization plays a significant role in the processing of biological information. This paper presents a set of piecewise linear (MLPWL1) and multiplierless piecewise linear (MLPWL2) neuron models, which mimic behaviors of different types of neurons, similar to the biological behavior of conductance-based neurons. Both simulations and a low-cost digital implementation are carried out to compare the proposed models to a single ML neuron and two coupled ML neurons, demonstrating the required range of dynamics with a more efficient implementation. Hardware implementations on a field-programmable gate array (FPGA) show that the modified models mimic the biological behavior of different types of neurons with higher performance and significantly lower implementation costs compared to the previous realizations of the ML model. The mean normalized root mean square errors (NRMSEs) of the MLPWL1 and MLPWL2 models are 3.70% and 4.89%, respectively, as compared to the original ML model.

Index Terms—Field-programmable gate array (FPGA), Morris-Lecar (ML) neuron model, spiking neural networks (SNN).

I. INTRODUCTION

IN RECENT decades, neuroscientists have been searching pathways to elucidate neural networks and activity in the brain. In elucidating how the brain works, neuroscientists typically propose specific models that can explain their theoretical and experimental observations [1], [2]. In order to explain the central neural system, one can consider a system that consists of primary basic units, i.e., neurons. Therefore, understanding of single neuron behavior as a primary building block, plays a critical role in this approach [3].

Spiking Neural Network (SNN) paradigms are significant for neuromorphic engineers and their research efforts in developing artificial neural networks have increased, recently [3], [4], [5]–[22].

A number of mathematical equations model behaviors underlying spiking neural networks [2], [13]. In the general case, the

equations of single neuron can produce systems with dynamical behaviors [6], [8] and can be represented by a set of models. Moreover, these models are described by nonlinear ordinary differential equations and hardware implementations of them have been studied intensively [3], [17], [23]–[27]. On the other hand, to provide devices that more accurately mimic biological systems, realization and implementation of neural networks are significant areas of interest [28].

In recent years, different types of neuron models have been presented. In these models, two main mechanisms are significant:

- Conductance-based models with biological precision.
- Spiking-based models, which describe temporal behavior of cortical spike trains [2], [13].

In these cases, when it is required to understand how neuronal behavior depends on measurable physiological parameters, the Hodgkin-Huxley type [5] models are more suitable and can explain the physiological mechanisms of neuronal behaviors [13]. However, for realization of cortical spike trains or spike-timing behaviors of neurons, spike-based models are appropriate. On the other hand, spike-based models such as the Integrate and Fire (IF), the Leaky-Integrate and Fire (LIF), and Izhikevich models, cannot describe the biological behaviors of neurons in the central nervous system (CNS). Indeed, conductance-based models, such as the Hodgkin-Huxley (HH) model, are computationally high cost for large scale simulation and have a prohibitive bottleneck when implemented [10]. There is a trade-off between model accuracy and its complexity. Thus we need to choose a simpler model to adopt hardware realization that displays the required biological behaviors.

Among the biological models that are suitable for these goals, the Morris-Lecar (ML) model is a prime example. The Morris-Lecar model [34] is a simple biophysical model and a prototype for a wide variety of neurons. It is a conductance-based model, introduced to explain the dynamics of the barnacle muscle fiber that describes the neuronal firing in a manner closely related to the biology. The ML model is described by two coupled first order differential equations. The first, models the evolution of the membrane potential and the second, models the activation of potassium current.

The implementation of neural models on different platforms has been studied [2], [13], [14]. Electronic components, circuits, and VLSI have been utilized to mimic neuronal dynamics. Moreover, an analog VLSI implementation can be used for the realization of neural models. Although these analog implementations are fast and efficient, they are inflexible and require a long development time [2], [13], [28].

While digitally implemented neurobiological networks consume more silicon area and power in comparison to analog implementations, they have shorter development times and

Manuscript received December 04, 2014; revised February 04, 2015; accepted April 04, 2015. Date of publication June 17, 2015; date of current version June 24, 2015. This paper was recommended by Associate Editor T. S. Gotarredona.

M. Hayati is with the Department of Electrical Engineering, Kermanshah branch, Islamic Azad University, Kermanshah, and Faculty of Engineering, Razi University, Tagh-E-Bostan, Kermanshah 67149, Iran (e-mail: mohsen_hayati@yahoo.com).

M. Nouri, and S. Haghir are with the Department of Electrical Engineering, Faculty of Engineering, Razi University, Tagh-E-Bostan, Kermanshah 67149, Iran (e-mail: mo.nouri@pgs.razi.ac.ir; moslemnouri70@gmail.com).

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

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2015.2423794

are more flexible. Moreover, a high precision digital implementation makes it possible to implement networks with high dynamic range and stability. Recently, reconfigurable digital platforms have been used to realize nervous systems and can provide compact and flexible solutions for biologically realistic neural systems [9]–[16]. In recent studies, the digital implementation of various neuron models, such as Izhikevich [17] and adaptive exponential [3] neuron models, have been considered. Computational cost, speed, and biological validity are the critical challenges of the digital implementation. Hence, a neuron model either is only sufficient universal to produce rich firing patterns exhibited by biological neurons or it is fast, simple, and effective computational cost. Note that the main limitation of the previous studies, to implement large scale networks on FPGA, is the number of available fast multipliers on a chip, i.e., [29], in which only 32 neurons are implemented because there are only 32 multipliers in the utilized FPGA boards. Moreover, a multiplier is a high cost building block in terms of area, delay, and power consumption. In [30] and [31], neuronal models are implemented, without any modifications to the original basic model. In order to realize conductance-based models, the original biophysical Morris-Lecar neuron model is implemented [32], [33]. Since the original Morris-Lecar neuron model has a high cost compared with Izhikevich and AdEx spike-based models, optimization of the model for the realization of a large number of neurons is significant. Therefore, a high-speed multiplierless design conductance-based model, such as the Morris-Lecar model, is significant. Aimed toward low hardware overhead, high-speed design, and a highly efficient realization of a two coupled ML model as a main block for large-scale considerations, this study presents a significantly simplified implementation of the Morris-Lecar conductance based neuron model. This paper, based on the ML model, proposes a set of piecewise linear (PWL) approximations. Compared with the original ML model, the modified models have reduced computational expense and can be implemented on low-cost and widely available hardware platforms such as field-programmable gate arrays (FPGAs). Results show that the modified models can mimic the biological ML model and represent the required range of dynamic neuron behaviors.

The rest of the paper is organized as follows. The next section presents a brief background of the original ML model, while in Section III, the dynamical analysis of the ML model is explained. In Section IV the proposed models and error analysis are investigated. Dynamical behavior of a system of two coupled neurons are described in Section V. Design and hardware implementation are discussed in Section VI. Section VII presents implementation results. Finally, Section VIII concludes the paper.

II. MORRIS-LECAR (ML) MODEL

The Morris-Lecar (ML) model is one of the simplest models for the production of action potentials and neural activity [34]. The model has three ionic currents: a potassium current, a membrane leakage current, and a calcium current. This model exhibits many important features of neuronal activity. The ML model produces action potentials, also a threshold for firing, and the model demonstrates sustained oscillations at a high level of applied current.

The simplest form of the ML model is a 2-D dynamical model. The state is determined by two variables, namely, the

membrane potential by activation of Ca^{2+} channels and the slow activation of K^+ channels. Mathematically, the model consists of two differential equations

$$\begin{cases} C_M \frac{dV}{dt} = I_{\text{stimulus}} - g_L(V - V_L) - g_{\text{Ca}} m_{\infty}(V)(V - V_{\text{Ca}}) \\ - g_{\text{K}} n(V - V_{\text{K}}) \\ \frac{dn}{dt} = \lambda(V)(n_{\infty}(V) - n) \end{cases} \quad (1)$$

where

$$\begin{cases} m_{\infty}(V) = 0.5 \left\{ 1 + \tanh \left(\frac{V - V_1}{V_2} \right) \right\} \\ n_{\infty}(V) = 0.5 \left\{ 1 + \tanh \left(\frac{V - V_3}{V_4} \right) \right\} \\ \lambda(V) = \lambda_{\text{max}} \cosh \left(\frac{V - V_3}{2V_4} \right). \end{cases} \quad (2)$$

Here, V determines the membrane potential and n is the fraction of open K^+ channels. Other parameters describing the ML model are given as follows:

- C_M : Membrane capacitance ($\mu\text{F}/\text{cm}^2$);
- I_{stimulus} : Applied current ($\mu\text{A}/\text{cm}^2$);
- $g_L, g_{\text{Ca}}, g_{\text{K}}$: Maximum or instantaneous conductance for leakage current, Ca^{2+} , and K^+ pathways, respectively (mS/cm^2);
- $V_L, V_{\text{Ca}}, V_{\text{K}}$: Equilibrium potentials corresponding to leakage current, Ca^{2+} and K^+ conductances, respectively (mV);
- $m_{\infty}(V), n_{\infty}(V)$: Fraction of open Ca^{2+} and K^+ channels of steady state;
- $\lambda(V)$: Rate constant for opening of K^+ channels (s^{-1});
- λ_{max} : Maximum rate constant for opening K^+ channels (s^{-1});
- V_1 : Membrane potential at $m_{\infty} = 0.5$ mV;
- V_2 : Reciprocal of slope of voltage dependence of m_{∞} (mV);
- V_3 : Membrane potential at $n_{\infty} = 0.5$ mV;
- V_4 : Reciprocal of slope of voltage dependence of n_{∞} (mV).

By scaling these parameters, the model can reproduce different types of spiking patterns such as: tonic, bursting and excitable.

III. DYNAMICS OF ML MODEL

To explain the transition from resting state to spiking state (bifurcation), the interactions of the two nullclines play an important role [6], [8], [35]. It will be convenient to write (1), (2) as

$$\begin{cases} \frac{dV}{dt} = f(V, n) \\ \frac{dn}{dt} = g(V, n). \end{cases} \quad (3)$$

The nullclines of the ML model are given as

$$\begin{cases} \frac{dV}{dt} = 0 \Rightarrow f(V, n) = 0 \\ \frac{dn}{dt} = 0 \Rightarrow g(V, n) = 0 \end{cases} \quad (4)$$

$$\begin{cases} n = \frac{I_{\text{stimulus}} - g_L(V - V_L) - g_{\text{Ca}} m_{\infty}(V)(V - V_{\text{Ca}})}{g_{\text{K}}(V - V_{\text{K}})} \\ n = 0.5 \left\{ 1 + \tanh \left(\frac{V - V_3}{V_4} \right) \right\}. \end{cases} \quad (5)$$

Fixed points are where the two nullclines intersect. The nullclines divide the phase plane into separate regions; in each of these regions, the vector field points in the direction of one of the four quadrants:

- i) $f > 0, g > 0$; ii) $f > 0, g < 0$; iii) $f < 0, g > 0$; iv) $f < 0, g < 0$.

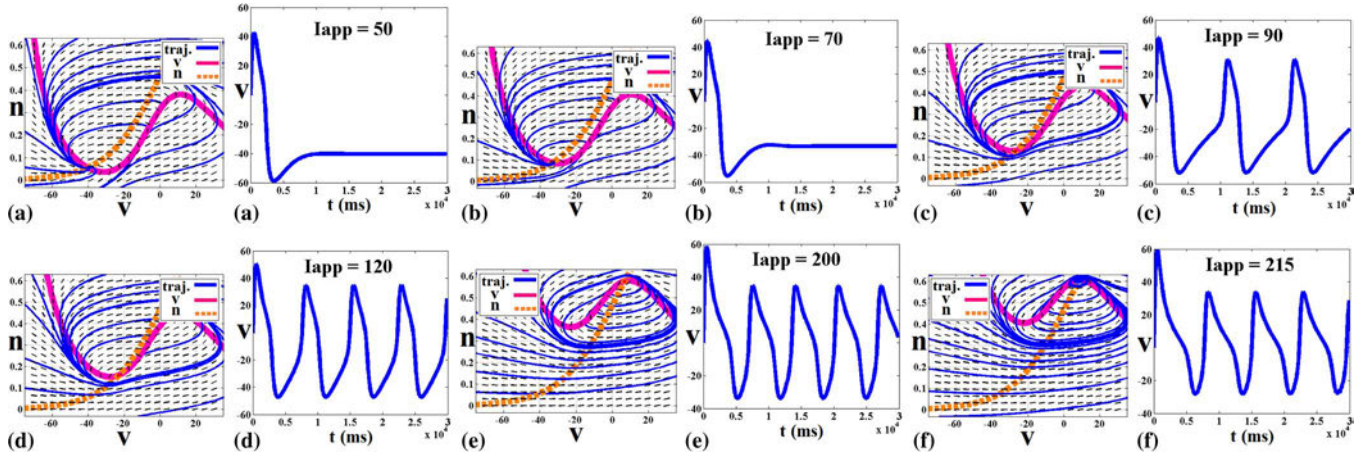


Fig. 1. The Hopf bifurcation for the Morris-Lecar model. (a)–(c) Stable region by increasing the stimulus current. (d)–(e) Unstable region in the Hopf bifurcation. (f) Stable region in the Hopf bifurcation.

TABLE I
PARAMETER VALUES FOR THE HOPF AND SADDLE-NODE BIFURCATIONS.
ABBREVIATIONS: PARAMETER (PA.), HOPF BIFURCATION (HB),
SADDLE-NODE BIFURCATION (SNB)

Pa.	HB	SNB	Unit	Pa.	HB	SNB	Unit
g_{Ca}	4.4	4	mS/cm ²	V_L	-60	-60	mV
λ_{max}	0.04	0.07	s ⁻¹	g_K	8	8	mS/cm ²
V_3	2	12	mV	g_L	2	2	mS/cm ²
V_4	30	17.4	mV	V_1	-1.20	-1.20	mV
V_{Ca}	120	120	mV	V_2	18	18	mV
V_K	-84	-84	mV	C_M	20	20	μF/cm ²

For a bifurcation analysis of equilibrium points, the Jacobean matrix and eigenvalues are required [6], [8], [36] and the Jacobean matrix can be obtained as

$$J(V, n) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} A &= \frac{\partial f(V, n)}{\partial V}, B = \frac{\partial f(V, n)}{\partial n}, C = \frac{\partial g(V, n)}{\partial V}, \\ D &= \frac{\partial g(V, n)}{\partial n} \quad (7) \\ \begin{cases} A = \frac{1}{C_M} [-g_L - g_K n - g_{Ca} m_\infty(V) \\ + \frac{g_{Ca}(V - V_{Ca})}{2V_2} \left[\frac{1}{\cosh\left(\frac{V - V_1}{V_2}\right)} \right]^2] \\ B = \frac{-g_K}{C_M} (V - V_K) \\ C = \left(\frac{\lambda_{max}}{2V_4} \right) \sinh\left(\frac{V - V_3}{2V_4}\right) (V) (n_\infty(V) - n) \\ + \lambda(V) \left(\frac{1}{2V_4} \right) \left[\frac{V}{\cosh\left(\frac{V - V_3}{V_4}\right)} \right]^2 \\ D = -\lambda(V). \end{cases} \quad (8) \end{aligned}$$

According to $J(V, n)$, the stability of the fixed point is determined. The fixed points are stable if $A + D < 0$ and they are unstable if $A + D > 0$. On the other hand, the fixed point is stable if both of the eigenvalues of this matrix have a negative real part and is unstable if at least one of the eigenvalues has a positive real part.

Bifurcation theory is elucidated in terms of how solutions change, as parameters in a model are varied. Using bifurcation theory, we can classify the types of transitions that take place as we change parameters. In particular, we can predict for which

TABLE II
EQUILIBRIUM POINTS FOR THE HOPF BIFURCATION DIAGRAM

Stimulus current	Fixed point type	Fixed point value
50 μA	Spiral Sink	(-40.30, 0.06)
70 μA	Spiral Sink	(-33.32, 0.09)
90 μA	Spiral Sink	(-26.60, 0.13)
120 μA	Spiral Source	(-23.10, 0.15)
200 μA	Spiral Source	(6.65, 0.57)
215 μA	Spiral Sink	(8.50, 0.60)

value of $I_{stimulus}$ the fixed point loses its stability and oscillations emerge. There are several different types of bifurcations. The most important types of bifurcations can be realized by the ML model.

A. The Hopf Bifurcation

The Hopf bifurcation is the mechanism through which one can go from a stable fixed point to an oscillation [2]. In this type of bifurcation, the stable fixed point first becomes unstable before merging with the other fixed points. In this case, we choose the parameters as in Table I. For the Hopf regime, the bifurcation diagram can be described by variation of $I_{stimulus}$, in the ML equations. In this state, by increasing the input stimulus, there are two fixed points for this current that a Hopf bifurcation occurs at $I_{stimulus} = I_1, I_{stimulus} = I_2$. It is expected that a fixed point is stable if all of the eigenvalues have a negative real part and it is unstable if at least one of the eigenvalues has a positive real part.

Fig. 1 demonstrates that Hopf bifurcation can appear by varying the stimulus current, $I_{stimulus}$, where the fixed point is stable for $I_{stimulus} < I_1 = 90 \mu A$ or $I_{stimulus} > I_2 = 210 \mu A$. Also, it is unstable for other regions. Therefore, a Hopf bifurcation occurs at I_1 and I_2 . Also, in Table II we can see that for $I_{stimulus} < I_1, I_{stimulus} > I_2$, the fixed points are stable and for $I_1 < I_{stimulus} < I_2$, there are unstable fixed points.

B. Saddle-Node Bifurcation

Another mechanism is called a saddle-node on a limit cycle (SNLC). It is also called a saddle-node on an invariant circle (SNIC) [2]. As depicted in Fig. 2, one of the fixed points is always a saddle (unstable) fixed point and when the saddle and node points are come together, finally disappear. On the other hand, when these points merge, they are called a saddle-node

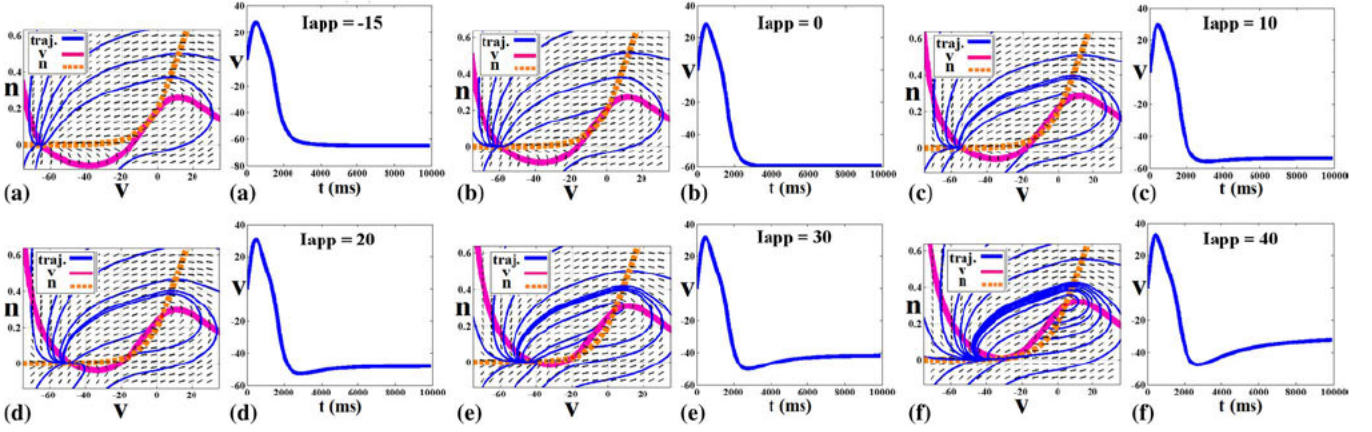


Fig. 2. Saddle-node bifurcation for the Morris-Lecar model. (a)–(f). By increasing I_{stimulus} , saddle-node equilibrium points vanish.

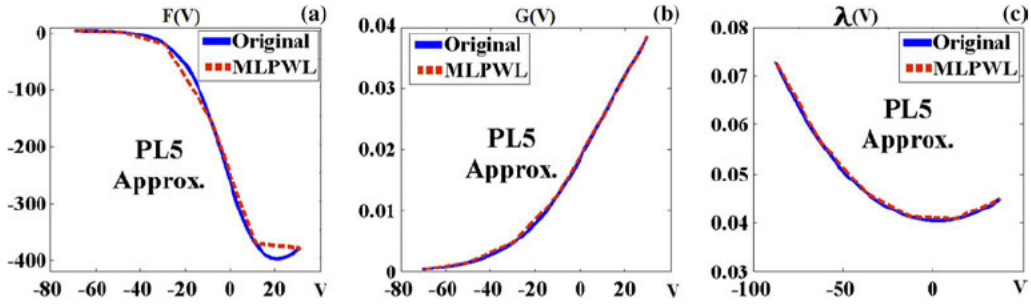


Fig. 3. Piecewise linear (PWL) models. (a) PWL5 model, F function (blue solid line), PWL5 approximation (red dotted lines). (b) PWL4 model, G function (blue solid line), PWL4 approximation (red dotted lines). (c) PWL5 model, $\lambda(V)$ function (blue solid line), PWL5 approximation (red dotted lines).

TABLE III
EQUILIBRIUM POINTS FOR THE SADDLE-NODE BIFURCATION DIAGRAM

Stimulus current	Nodal Sink	Saddle Point	Nodal Source	Spiral Source
-10 μA	(-64.70, 0.00)	—	—	—
0 μA	(-59.47, 0.00)	(-9.48, 0.08)	(0.16, 0.20)	—
10 μA	(-54.08, 0.00)	(-12.56, 0.06)	—	(1.74, 0.23)
20 μA	(-48.36, 0.00)	(-15.70, 0.04)	—	(2.90, 0.26)
30 μA	(-41.84, 0.00)	(-19.56, 0.03)	—	(3.87, 0.28)
40 μA	—	—	—	(4.70, 0.30)

point. Also, in Table III, we can see that by increasing stimulus current, saddle-node equilibrium points vanish.

IV. PROPOSED MODELS AND ERROR ANALYSIS

A. Proposed Models

To improve the computational efficiency of the model and reduce the implementation costs, the original ML model has been modified. The hyperbolic functions in the ML equations as a significant target for modification, are a challenge in a low-cost digital implementation.

1) *MLPWL1*: In this section, proposed modifications to the original model are presented. The main motivation of these modifications is the implementation cost for the modified design.

The membrane potential equation of ML model can be rewritten as

$$\begin{cases} C_M \frac{dV}{dt} = I_{\text{stimulus}} - g_L(V - V_L) - F(V) - g_K n(V - V_K) \\ F(V) = g_{Ca} m_{\infty}(V)(V - V_{Ca}) \end{cases} \quad (9)$$

where F is a hyperbolic-based function and as depicted in Fig. 3(a), which can be approximated by five PWL segments (shown

with red dotted lines), representing linear and hyperbolic terms in the equation.

Accordingly, the PWL5 model is given by

$$F(V) = \begin{cases} m_0 V + k_0 & ; V_1 < V < V_2 \\ m_1 V + k_1 & ; V_2 < V < V_3 \\ m_2 V + k_2 & ; V_3 < V < V_4 \\ m_3 V + k_3 & ; V_4 < V < V_5 \\ m_4 V + k_4 & ; \text{else} \end{cases} \quad (10)$$

where $m_i, 0 < i < 4$ are the slopes of lines in the PWL5 approximation of the F function. Consequently, the hyperbolic term is mathematically approximated by a line, which has a significantly lower implementation cost compared with the original term.

On the other hand, in the second equation of ML model, there are nonlinear terms and this equation can be rewritten as

$$\begin{cases} \frac{dn}{dt} = G(V) - n\lambda(V) \\ G(V) = \lambda(V)n_{\infty}(V) \end{cases} \quad (11)$$

where G is a hyperbolic-based function and as depicted in Fig. 3(b), can be approximated by four lines (shown with red dotted lines), representing linear and hyperbolic terms in the equation. Accordingly, the PWL4 model is given by

$$G(V) = \begin{cases} m_0 V + k_0 & ; V_1 < V < V_2 \\ m_1 V + k_1 & ; V_2 < V < V_3 \\ m_2 V + k_2 & ; V_3 < V < V_4 \\ m_3 V + k_3 & ; \text{else} \end{cases} \quad (12)$$

where $m_i, 0 < i < 3$ are the slopes of lines in the PWL4 approximation of the G function. Also, the $\lambda(V)$ function can be approximated by five linear segments. As depicted in Fig.

3(c), linear and hyperbolic terms in the equation are presented. Therefore, the PWL5 model for $\lambda(V)$ is given by

$$\lambda(V) = \begin{cases} m_0V + k_0 & ; V_1 < V < V_2 \\ m_1V + k_1 & ; V_2 < V < V_3 \\ m_2V + k_2 & ; V_3 < V < V_4 \\ m_3V + k_3 & ; V_4 < V < V_5 \\ m_4V + k_4 & ; \text{else.} \end{cases} \quad (13)$$

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.

Using piecewise linear approximations converts the ML equations to

$$\begin{cases} C_M \frac{dV}{dt} = I_{\text{stimulus}} - g_L(V - V_L) - F(V) - g_K n(V - V_K) \\ \frac{dn}{dt} = G(V) - n\lambda(V). \end{cases} \quad (14)$$

As can be shown in these approximated equations, there are two multiplication operations. In the digital hardware implementation, multiplier operator has significantly higher implementation cost than the multiplierless design, which is discussed as follows.

2) *MLPWL2*: The key idea of this approximation is to convert the exponential function to powers of 2, which can be simply implemented by a digital logic shift. Therefore, replacing multipliers with logic shift and add operations leads to a considerable hardware reduction. As a consequence, the original ML model is rewritten as

$$\begin{cases} C_M \frac{dV}{dt} = I_{\text{stimulus}} - g_L(V - V_L) - F(V) \\ - g_K \sinh(n)(V - V_K) \\ \frac{dn}{dt} = G(V) - \lambda_{\max} \cosh\left(\frac{V - V_4}{2V_4}\right) n. \end{cases} \quad (15)$$

Therefore, multiplication operations are replaced by shift and add operations as demonstrated at Section VI. The result is a low-cost implementation of the ML model.

B. Definition of Errors

In this case, root mean square error (RMSE) is used as an error between the proposed models and the original ML model in terms of output membrane potentials.

For MLPWL models, it can be defined as

$$\text{RMSE}(V_{\text{MLPWL}}, V_{\text{ML}}) = \sqrt{\frac{\sum_1^n (V_{\text{MLPWL}} - V_{\text{ML}})^2}{n}}. \quad (16)$$

Whereas, the normalized root mean square error (NRMSE) is defined as

$$\text{NRMSE} = \frac{\text{RMSE}}{V_{\max} - V_{\min}} \quad (17)$$

where V_{ML} is the original membrane potential, and V_{MLPWL} is the approximated membrane potential with values given in Table IV.

Accordingly, modified parameters for different states of spiking activity in the proposed MLPWL models were calculated for time step value, 10 ms. Table V indicates that proposed models have an acceptable accuracy. The mean NRMSEs of the MLPWL1 and MLPWL2 models are 3.70% and 4.89%, respectively, as compared to the original ML model.

TABLE IV
MLPWL MODIFIED COEFFICIENTS

coefficient	Approximated Function		
	$F(V)$	$G(V)$	$\lambda(V)$
(m_0, k_0)	$(-0.38, -23.00)$	$(0.00, 0.01)$	$(-0.01, 0.02)$
(m_1, k_1)	$(-2.80, -110.00)$	$(0.00, 0.01)$	$(-0.00, 0.03)$
(m_2, k_2)	$(-11.00, -280.00)$	$(0.00, 0.02)$	$(-0.00, 0.04)$
(m_3, k_3)	$(-2.60, -350.00)$	$(0.00, 0.02)$	$(0.00, 0.04)$
(m_4, k_4)	$(1.80, -440.00)$	—	$(0.00, 0.04)$

TABLE V
RMSE AND NRMSE COMPUTATIONS FOR DIFFERENT STIMULUS CURRENTS WITH TIME Step = 10 ms

Stimulus Current	MLPWL1		MLPWL2	
	RMSE	NRMSE%	RMSE	NRMSE%
50 μA	1.10	5.87	1.24	7.01
70 μA	2.50	4.22	2.61	3.70
115 μA	33.34	0.89	17.42	1.54
120 μA	31.76	4.61	11.80	2.73
200 μA	33.32	0.84	28.80	6.60
212 μA	26.90	5.80	35.20	7.80
Mean error	21.47	3.70	16.17	4.89

TABLE VI
SPECIFIC VALUES OF THE TIME SCALES, THRESHOLD VALUES, AND CONTROL PARAMETERS

$$h_s = -2 \quad d_s = 3 \quad \tau_s = 10 \quad S_s = 1$$

V. THE DYNAMICAL BEHAVIOR OF TWO COUPLED ML NEURONS

A. Synapse Model

Synapses are specialized structures that allow either chemical or electrical signals to pass from the pre-synaptic to the target postsynaptic neuron with an associated synaptic strength or efficacy [4]. This model consists of a synapse [37], [38], which includes a terminal that incorporates a presynaptic neuron and a postsynaptic neuron, and can be specified as follows:

$$\begin{cases} \tau_s \frac{dZ}{dt} = [1 + \tanh(S_s(V_{\text{pre}} - h_s))](1 - Z) - \frac{Z}{d_s} \\ I_{\text{syn}} = k_s(Z - Z_0) \end{cases} \quad (18)$$

where Z is the synaptic activation variable and there are seven parameters describing the synaptic activity as follows:

- τ_s : Time delay (s);
- S_s, d_s : Responsible for activation and relaxation of Z ;
- h_s : Threshold parameter for activation of Z ;
- I_{syn} : Synaptic current;
- k_s : Plays the role of conductance;
- Z_0 : Reference level of Z .

If the presynaptic neuron is in a quiescent state then Z_0 can be obtained as follows:

$$Z_0 = \frac{2d_s}{1 + 2d_s + \exp(2I_{\text{stimulus}})}. \quad (19)$$

In this approach, when $V_{\text{pre}} < h_s$, the synapse is inactive and $Z = 0$. Increasing V_{pre} makes $S_s(V_{\text{pre}} - h_s)$ positive and switches the hyperbolic tangent function to positive values. Once activated, the synaptic terminal provides the postsynaptic current as I_{syn} . The specific values of the time scales, threshold values, and control parameters of the synaptic coupling model have been presented in Table VI.

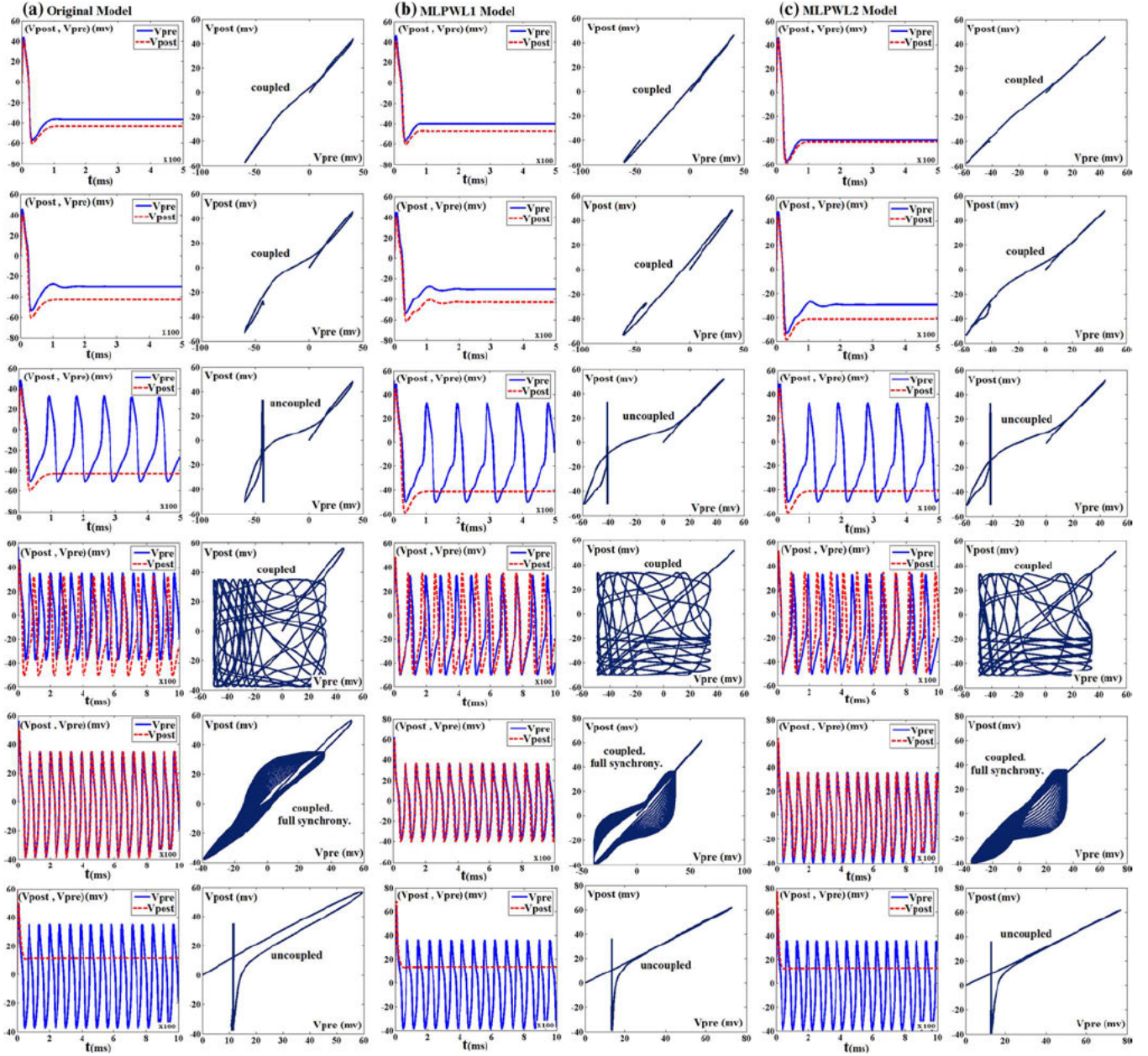


Fig. 4. The dynamic behavior and phase portraits of two coupled Morris-Lecar neurons with the different values of current stimulus and conductance coefficient. For $I_{\text{stimulus}} = 90 \mu\text{A}$ and $k_s = 200$, full synchrony will occur. (a) The synaptic coupling of the original model. (b) The synaptic coupling of the MLPWL1 model. (c) The synaptic coupling of the MLPWL2 model.

B. Synapse Modified Model

The nullclines of the synapse can be written as

$$\frac{dZ}{dt} = 0 \Rightarrow Z = \frac{1 + \tanh(S_s(V_{\text{pre}} - h_s))}{1 + \frac{1}{a_s} + \tanh(S_s(V_{\text{pre}} - h_s))}. \quad (20)$$

Based on the threshold value h_s , Z increases to 1 or decreases to 0 with the rate proportional to $1/\tau_s$ and therefore, the modified model can be written as

$$Z = \frac{1 + \text{sign}(S_s(V_{\text{pre}} - h_s))}{2} = \begin{cases} 1 & ; V_{\text{pre}} > h_s \\ 0 & ; V_{\text{pre}} < h_s \end{cases} \quad (21)$$

The differential equation is mathematically approximated by a signum function, which has a significantly lower implementation cost compared with the original synapse model.

C. Synaptic Coupling Model

In this section, dynamical behaviors of two coupled ML neurons are presented. In this way, we can see the various dynamical behaviors as the current stimulus of the presynaptic neuron and the conductance coefficient of the synaptic terminal are varied. The synaptic terminal acts as an active gate and when the presynaptic voltage level reaches its threshold value, voltage transmission can occur. This state depends on the input stimulus, the coupling of the neurons with the same potentials ($V_{\text{pre}} = V_{\text{post}}$), and when two coupled neurons are synchronized. As mentioned previously, the synchronization effects of coupled neurons are significant for the processing of biological signal and play significant roles in the elucidation of diseases, such as Parkinson's, essential tremor, and epilepsy.

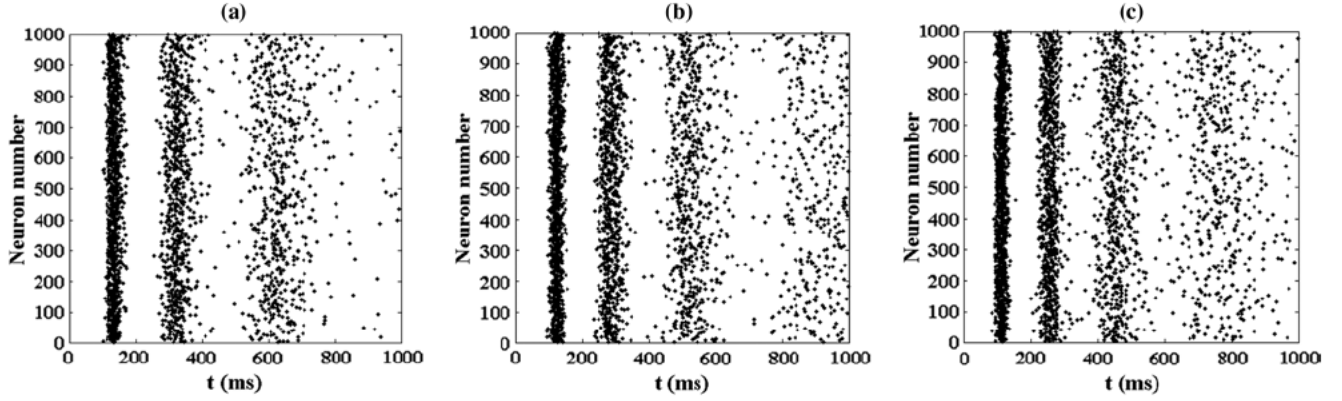


Fig. 5. Raster plot representing the activity of 1000 neurons from a population of tonic neurons for original model (a) and all of proposed model (b)–(c). Each neuron is randomly connected to other neurons. The figure illustrates how patterns of action potential of a population of neurons synchronized. (a) Original model. (b) MLPWL1 model. (c) MLPWL2 model.

Consequently, by appropriate selection of input current stimulus and synaptic conductance coefficient, the synchronization effects can be controlled.

This coupled original model is specified as follows:

$$\begin{cases} C_M \frac{dV_{pre}}{dt} = I_{stimulus} - g_L(V_{pre} - V_L) \\ -g_{Ca}m_{\infty}(V_{pre})(V_{pre} - V_{Ca}) - g_K n_{pre}(V_{pre} - V_K) \\ \frac{dn_{pre}}{dt} = \lambda(V_{pre})(n_{\infty}(V_{pre}) - n_{pre}) \\ \tau_s \frac{dZ}{dt} = [1 + \tanh(S_s(V_{pre} - h_s))] (1 - Z) - \frac{Z}{d_s} \\ C_M \frac{dV_{post}}{dt} = I_{syn} - g_L(V_{post} - V_L) \\ -g_{Ca}m_{\infty}(V_{post})(V_{post} - V_{Ca}) - g_K n_{post}(V_{post} - V_K) \\ \frac{dn_{post}}{dt} = \lambda(V_{post})(n_{\infty}(V_{post}) - n_{post}). \end{cases} \quad (22)$$

Fig. 4 shows the different behaviors of two coupled neurons. In the general case, for specific values of the stimulus current ($I_{stimulus}$) and the conductance coefficient (k_s), full synchrony will be occur. As demonstrated in Fig. 4, the proposed models can mimic the biological ML model with high precision. The raster plots of the simulations are presented in Fig. 5. The network activities of the original model and the proposed models with the approximately same inputs are very similar in structure.

VI. HARDWARE IMPLEMENTATION

This section presents the hardware implementation structure for the MLPWL2 model. In order to obtain an improved comparison between original, MLPWL1 and MLPWL2 models, according to (1), (2), and (9)–(16), the scheduling diagrams (Data Flow Graph with scheduling control steps) of original and proposed models are drawn in Fig. 6.

A. Equations Discretizing

Each design consists of two blocks to calculate (dV/dt) and (dn/dt) for models as shown in their corresponding equations. In this section, discretization of equations, we utilize the Euler method.

B. Bit Width Determination and Optimization Details

In order to determine the bit-width of the variables and parameters, two basic factors must be taken into account. The factors are the minimum/maximum bounds of the parameter values and the spans of the logic shifts. In the ML model, the spans of V is -70 to 30 mV and the minimum bits for implementing the membrane potentials are 8 bits. In this method, the variable

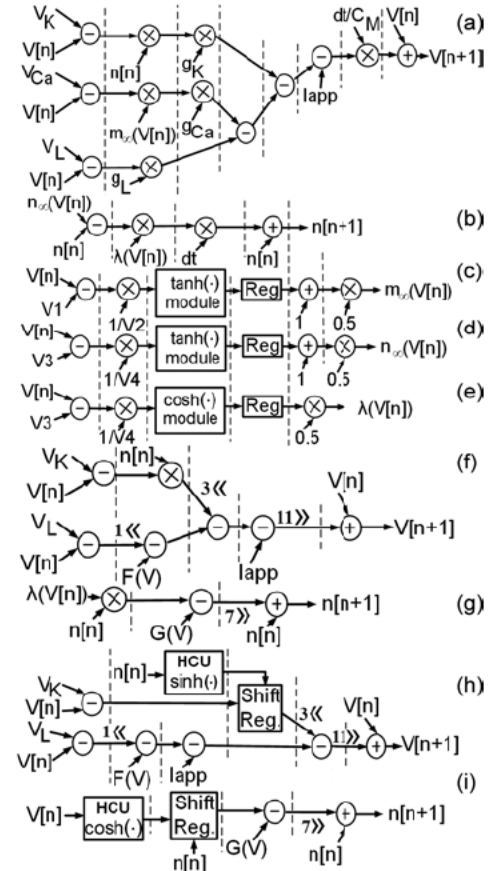


Fig. 6. Comparison between the scheduling diagrams (Data Flow Graph with scheduling control steps) of the original and proposed models. (a) V pipeline in original model. (b) n pipeline in original model. (c) $m_{\infty}(V)$ pipeline in original model. (d) $n_{\infty}(V)$ pipeline in original model. (e) $\lambda(V)$ pipeline in original model. (f) V pipeline in MLPWL1 model. (g) n pipeline in MLPWL1 model. (h) V pipeline in MLPWL2 model. (i) n pipeline in MLPWL2 model.

range is from -2^8 to 2^8 . Thus the variable V can be implemented in this range. The other variables and parameters such as MLPWL line coefficients need 9 bits for the integer part and 14 bits for the fraction. In the bit-width determination, if the maximum logic shifts to right or left are not considered, then overflow can occur. To avoid any overflow and also increasing accuracy of the calculations, a bit width of 30 that consists of 10 bits for integer part and 20 bits for the fraction, is considered.

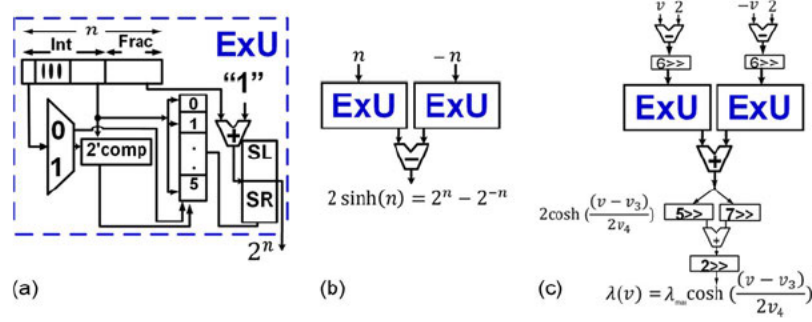


Fig. 7. Hyperbolic Calculation Unit (HCU) using: Exponential Unit (ExU), shift-to-left (SL), shift-to-right (SR), two's complement (2' comp). (a) Exponential Unit (ExU) [2]. (b) $\sinh(\cdot)$ calculation unit. (c) $\cosh(\cdot)$ calculation unit.

C. Hyperbolic Calculation Unit (HCU)

An approach for implementation of an exponential function in multiplierless form is presented in [2]. Generating the exponential functions with powers of 2, is the key idea of this approach, which is realized by a logic shift. Replacing multipliers with logic shift operations leads to a significant low-cost implementation [2]. The MLPWL2 model needs two stages of an exponential units (Exp). In [2] an exponential unit has been designed using powers of 2 that is demonstrated in Fig. 7—where, v is the input, which consists of *Frac* and *Int* parts. Note that *Frac* extracts the fractional part and *Int* gives the integer part. At first, the sign bit (which is used to distinguish positive numbers and negative numbers) is checked. If it is 0 the last six bits of the integer part are given to the search segment for finding true bits, otherwise, if the sign bit is 1, the last six bits of the integer part are applied to a two's complement unit. The output of two's complement unit enables a 2-level shift unit that is then the input of a further shift unit and 2^v is thereby computed. This unit performs the search using a sign bit checker [2]. So v can be negative or positive integer. By using this method, we are able to use this architecture twice in our proposed design to calculate hyperbolic functions—this is called the hyperbolic calculation unit (HCU).

D. Pipelined Design

Fig. 8 shows a digital multiplierless implementation of the neuron model according to (15). This unit includes V_{pipeline} , n_{pipeline} , V_{buffer} , and n_{buffer} . V_{pipeline} , n_{pipeline} that are represented by V and n in (23). These variables are performed in pipeline structures with V_{stage} and n_{stage} stages, where V_{buffer} and n_{buffer} are the buffer registers for the V and n values and $V_{\text{buffer-size}}$ and $n_{\text{buffer-size}}$ are the size of V and n states. With every rising edge of the clock the buffers are shifted to obtain new values. The bit number can be calculated, according to usage and precision. Based on the variable equations, to create repetitive states, buffer outputs are applied to the related arithmetic units. None of the modules are in a pipelined state, implies that registers are needed in order to preserve data. Accordingly, following conditions must be satisfied:

$$\begin{cases} N = V_{\text{buffer-size}} + V_{\text{stage}} = n_{\text{buffer-size}} + n_{\text{stage}} \\ V_{\text{buffer-size}} = n_{\text{buffer-size}} \\ V_{\text{stage}} = n_{\text{stage}} \end{cases} \quad (23)$$

where N is the number of neurons. If the number of stages are not the same, we schedule the delay of each stage in order to allow the synchronization of the equations. This condition

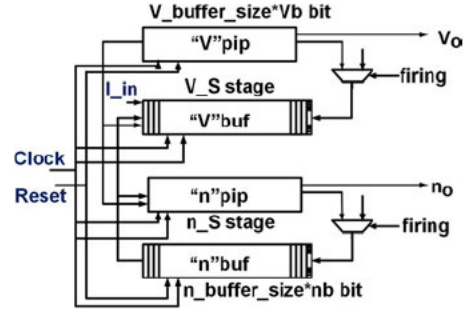


Fig. 8. General overview of the pipeline structure. This unit includes V_{pipeline} , n_{pipeline} , V_{buffer} and n_{buffer} . V_{pipeline} , n_{pipeline} that are represented by V and n in (23). These variables are performed in pipeline structures with V_{stage} and n_{stage} stages, where V_{buffer} and n_{buffer} are the buffer registers for the V and n values and $V_{\text{buffer-size}}$ and $n_{\text{buffer-size}}$ are the size of V and n states. (pip=pipeline, buf=buffer).

is provided by applying neuron input at the latest step of the pipeline stages.

VII. IMPLEMENTATION RESULTS

Circuits are implemented on a XILINX Virtex-II Pro Development System. Fig. 9, shows oscilloscope photographs of the dynamical behavior of a single neuron implemented on this FPGA platform using ML and the two proposed MLPWL models. The MLPWL2 model has a lower implementation cost. However, the MLPWL1 model has higher accuracy for most cases. The device utilization for implementation of the proposed models is summarized in Table VII.

The results of hardware implementation show that the MLPWL2 model is low-cost compared to the implementation of the original ML neuron model, and this is expected because hyperbolic terms and multiplications require a high area consumption for a circuit implementation. In the MLPWL2 implementation, multipliers are eliminated, but with an increase in the number of slices. However, calculations show there is approximately a 6% overall saving in FPGA area for a given frequency. But there is a trade-off, and for a given area, the new implementation offers about a 9% speed-up.

VIII. CONCLUSION

A set of piecewise linear (MLPWL1) and multiplierless piecewise linear (MLPWL2) models based on the Morris-Lecar model targeting low cost digital implementation have been presented. The results demonstrated that the proposed models have desired accuracy and are suitable for digital implementation. The MLPWL1 and MLPWL2 models have lower computational and hardware cost compared with the original Morris-Lecar neuron model and a set of multiplierless hardware

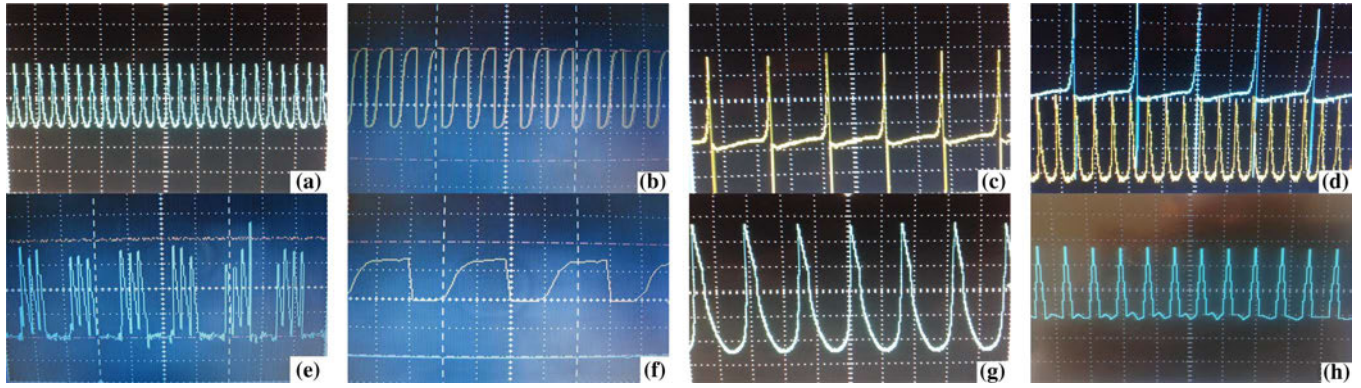


Fig. 9. Output of the MLPWL2 model implemented on XILINX Virtex-II Pro XC2VP30. (a) Membrane potential at current variable ($I = 128 \mu\text{A}$). (b) Membrane potential at $I = 90 \mu\text{A}$ (tonic). (c) Membrane potential at $I = 80 \mu\text{A}$ (tonic). (d) Membrane potential at different stimulus currents. (e) Membrane potential (tonic bursting spiking pattern). (f) Membrane potential at $I = 75 \mu\text{A}$ (tonic). (g) Membrane potential at current variable ($I = 100 \mu\text{A}$). (h) Membrane potential at current variable ($I = 110 \mu\text{A}$). The horizontal axis denotes time (time scale = 10 ms), and the vertical axis shows voltage (voltage scale = 25 mV).

TABLE VII

DEVICE UTILIZATION OF THE XILINX VIRTEx-II PRO. ABBREVIATIONS: RESOURCE (RES.), UTILIZATION (UTIL.), AVAILABLE (AV.), FF' SLICE (FF' S.), 4 INPUT LUTS (LUTS), BONDED IOBS (IOBS), MULT18 \times 18 S (MU.), AND FREQUENCY (FREQ.)

Res.	Original		MLPWL1		MLPWL2		
	Used	Util.	Used	Util.	Used	Util.	Av.
Slices	528	3%	476	3%	1937	14%	13696
FF' S.	605	2%	514	1%	618	2%	27392
LUTs	679	2%	856	3%	3616	13%	27392
IOBs	63	11%	63	11%	63	11%	556
MU.	10	7%	8	5%	0	0%	136
GCLKs	1	6%	1	6%	1	6%	16
Freq.	96.7 MHz		113 MHz		135 MHz		400 MHz

structures have been proposed and implemented based on an effective reduction of hardware and computation. These models are conveniently implemented on FPGA. This hardware is used to demonstrate different dynamics of the ML neuron model depending on the current stimulus, and producing different patterns of spiking activity with minimal computational error.

REFERENCES

- [1] M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design implementation, application, challenges," *Proc. IEEE*, vol. 102, no. 5, pp. 717–737, May 2014.
- [2] S. Gomar and A. Ahmadi, "Digital multiplierless implementation of biological adaptive-exponential neuron model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1206–1219, Apr. 2013.
- [3] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.
- [4] M. Rahimi Azghadi, S. F. Al-Sarawi, D. Abbott, and N. Iannella, "A neuromorphic VLSI design for spike timing and rate based synaptic plasticity," *Neural Netw.*, vol. 45, pp. 70–82, 2013.
- [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] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [7] E. M. Izhikevich, "Resonate-and-fire neurons," *Neural Netw.*, vol. 14, no. 6, pp. 883–894, 2001.
- [8] E. M. Izhikevich, *Dynamic Systems in Neuroscience*. Cambridge, MA, USA: MIT Press, 2007.
- [9] M. J. Pearson, A. G. Pipe, B. Mitchinson, K. Gurney, C. Melhuish, I. Gilhespy, and M. Nibouche, "Implementing spiking neural networks for real-time signal-processing and control applications: A model-validated FPGA approach," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1472–1487, 2007.
- [10] M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, and D. Abbott, "Tunable low energy, compact and high performance neuromorphic circuit for spike-based synaptic plasticity," *PLOS ONE*, vol. 9, no. 2, p. e88326, 2014.
- [11] S. Hashimoto and H. Torikai, "A novel hybrid spiking neuron: Bifurcations, responses, on-chip learning," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 2168–2181, 2010.
- [12] T. Hishiki and H. Torikai, "A novel rotate-and-fire digital spiking neuron and its neuron-like bifurcations and responses," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 752–767, 2011.
- [13] H. Soleimani, A. Ahmadi, and M. Bavandpour, "Biologically inspired spiking neurons: Piecewise linear models and digital implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 12, pp. 2991–3004, 2012.
- [14] F. Grassia, T. Lévi, S. Saïghi, and T. Kohno, "Bifurcation analysis in a silicon neuron," *Artif. Life Robot.*, vol. 17, no. 1, pp. 53–58, 2012.
- [15] L. Geretti and A. Abramo, "The correspondence between deterministic and stochastic digital neurons: Analysis and methodology," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1739–1752, 2008.
- [16] J. Zhou, W. Yu, X. Li, M. Small, and J. A. Lu, "The correspondence between deterministic and stochastic digital neurons: Analysis and methodology," *IEEE Trans. Neural Netw.*, vol. 20, no. 10, pp. 1679–1684, 2009.
- [17] G. Indiveri, E. Chicca, and R. Douglas, "VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 211–221, 2006.
- [18] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Baranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Gamunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S. C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. C. Ballcells, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, and B. Linares-Barranco, "CAVIAR: A 45 k Neuron, 5 M synapse, 12 G connects/s AER hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, 2009.
- [19] R. J. Vogelstein, U. Mallik, J. T. Vogelstein, and G. Cauwenberghs, "Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 253–265, 2007.
- [20] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "A neuromorphic cortical-layer microchip for spike-based event processing vision systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 12, pp. 2548–2566, 2006.
- [21] Blue Brain Project [Online]. Available: <http://bluebrain.epfl.ch>, Dec. 2013.
- [22] S. B. Furber, S. Temple, and A. D. Brown, "High-performance computing for systems of spiking neurons," in *Proc. AISB06 Workshop GC5 Archit. Brain Mind*, 2006, vol. 2, pp. 29–36.
- [23] T. Yu and G. Cauwenberghs, "Analog VLSI biophysical neurons and synapses with programmable membrane channel kinetics," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 3, pp. 139–148, 2010.
- [24] T. Asai, Y. Kanazawa, and Y. Amemiya, "A subthreshold MOS neuron circuit based on the Volterra system," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1308–1312, 2003.

- [25] E. Farquhar and P. Hasler, "A bio-physically inspired silicon neuron," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 477–488, 2005.
- [26] M. Simoni and S. DeWeelth, "Adaptation in a VLSI model of a neuron," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 7, pp. 967–970, 2002.
- [27] T. Kohno and K. Aihara, "A MOSFET-based model of a class 2 nerve membrane," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 754–773, 2005.
- [28] O. Sharifipour and A. Ahmadi, "An analog implementation of biologically plausible neurons using CCII building blocks," *Neural Netw.*, vol. 36, no. 1, pp. 129–135, 2012.
- [29] A. Cassidy and A. G. Andreou, "Dynamical digital silicon neurons," *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, pp. 289–292, Nov. 2008.
- [30] M. Ambroise, T. Levi, S. Joucla, B. Yvert, and S. Saïghi, "Real-time biomimetic central pattern generators in an FPGA for hybrid experiments," *Frontiers Neurosci.*, vol. 7, no. 6, p. 215, Nov. 2011, hal-00956624.
- [31] F. Grassia, T. Lévi, T. Kohno, and S. Saïghi, "Silicon neuron: Digital hardware implementation of the quartic model," *Proc. Int. Symp. Artif. Life Robot.*, Jan. 2014, DOI 10.1007/s10015-014-0160-2.
- [32] Y. Kang-Le, W. Jiang, W. Xi-Le, and D. Bin, "Morris-Lecar model neurons and neural networks: FPGA implementation and analysis," in *Proc. IEEE 2011 30th Chinese Control Conf. (CCC)*, 2011, pp. 639–642.
- [33] Z. Ronghua, W. Jiang, L. Shuangshuang, and C. Yanqiu, "High-speed simulation for neuron system base on FPGA," in *Proc. IEEE 2011 30th Chinese Control Conf. (CCC)*, 2010, pp. 5500–5504.
- [34] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle fiber," *Biophys. J.*, vol. 35, no. 1, pp. 193–213, 1981.
- [35] J. Touboul and R. Brette, "Dynamics and bifurcations of the adaptive exponential integrate-and-fire model," *Biol. Cybern.*, vol. 99, no. 45, pp. 319–334, Jul. 2008.
- [36] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [37] D. E. Postnov, L. S. Ryazanov, and O. V. Sosnovtsev, "Functional modeling of neural-glia interaction," *BioSystems*, vol. 89, no. 1, pp. 84–91, 2007.
- [38] 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," *J. Biol. Phys.*, vol. 35, no. 4, pp. 425–445, 2009.



Mohsen Hayati received his B.E. degree in electronics and communication engineering from Nagarjuna University, Andhra Pradesh, India, in 1985, and his M.E. and Ph.D. degrees in electronics engineering from Delhi University, Delhi, India, in 1987 and 1992, respectively. He joined the Electrical Engineering Department, Razi University, Kermanshah, Iran, as an Assistant Professor in 1993. Currently, he is a Professor with the Electrical Engineering Department, Kermanshah branch, Islamic Azad University and Razi University. He

has published more than 155 papers in international, domestic journals, and conferences. His current research interests include microwave and millimeter wave devices and circuits, application of computational intelligence, artificial neural networks, fuzzy systems, neuro-fuzzy systems, electronic circuit synthesis, and modeling and simulations.



Moslem Nouri received the B.Sc. degree in electrical engineering from Azad University, Kermanshah, Iran, in 2012, and the M.Sc. degree in electronic engineering from the Department of Electrical Engineering, Razi University, Kermanshah, Iran, in 2014 (with Honors), where, he is currently a Ph.D. student of electronic engineering, Razi University, Kermanshah, Iran.

His research interests include high-frequency high-efficiency power amplifiers and oscillators, resonant dc/dc power converters, numerical simulation of switching circuits, analog and digital electronic circuit design and optimization, bio-inspired computing, neuromorphic, memristor, artificial cochlear and integrated circuit design. (E-mail: moslemnouri70@gmail.com; mo.nouri@pgs.razi.ac.ir)



Saeed Haghiri received the B.Sc. and the M.Sc. degrees in electronic engineering from the Department of Electrical Engineering, Razi University, Kermanshah, Iran, in 2012 and 2014, respectively.

His research interests include digital electronic circuit design and optimization, bio-inspired computing, high performance computing, neuromorphic and integrated circuit design.



Derek Abbott (M'85–SM'99–F'05) was born in South Kensington, London, U.K., in 1960. He received the B.Sc. (honors) degree in physics from Loughborough University, Leicestershire, U.K., in 1982 and the Ph.D. degree in electrical and electronic engineering from The University of Adelaide, Adelaide, S.A. Australia, in 1995, under K. Eshraghian and B. R. Davis.

From 1978 to 1986, he was a Research Engineer at the GEC Hirst Research Centre, London, U.K. From 1986 to 1987, he was a VLSI Design Engineer at Austek Microsystems, Australia. Since 1987, he has been with The University of Adelaide, where he is presently a full Professor with the School of Electrical and Electronic Engineering. He coedited *Quantum Aspects of Life* (London, U.K.: Imperial College Press, 2008), coauthored *Stochastic Resonance* (Cambridge, U.K.: Cambridge University Press, 2012), and coauthored *Terahertz Imaging for Biomedical Applications* (New York: Springer-Verlag, 2012). He holds over 800 publications/patents and has been an invited speaker at over 100 institutions. His interests are in the area of multidisciplinary physics and electronic engineering applied to complex systems. His research programs span a number of areas of stochastics, game theory, photonics, biomedical engineering, and computational neuroscience.

Prof. Abbott is a Fellow of the Institute of Physics (IOP). He has won a number of awards including the South Australian Tall Poppy Award for Science (2004), the Premiers SA Great Award in Science and Technology for outstanding contributions to South Australia (2004), and an Australian Research Council (ARC) Future Fellowship (2012). He has served as an Editor and/or Guest Editor for a number of journals including the *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, *Journal of Optics B*, the *Microelectronics Journal*, *Chaos*, *Smart Structures and Materials*, *Fluctuation and Noise Letters*, *PLOS ONE*, and is currently on the editorial boards of the *PROCEEDINGS OF THE IEEE*, the *IEEE PHOTONICS JOURNAL*, and *Nature's Scientific Reports*.