Neurocomputing 165 (2015) 468–476

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Digital multiplierless implementation of the biological FitzHugh–Nagumo model

M. Nouri [a], Gh.R. Karimi [a,*], A. Ahmadi [a], D. Abbott [b]

[a] Electrical Engineering Department, Faculty of Engineering, Razi University, Tagh-E-Bostan, Kermanshah 67149, Iran
[b] School of Electrical & Electronic Engineering, The University of Adelaide, SA 5005, Australia

ARTICLE INFO

Article history:
Received 16 July 2014
Received in revised form
24 February 2015
Accepted 10 March 2015
Communicated by R.W. Newcomb
Available online 20 April 2015

Keywords:
Spiking neural network
Piecewise linear model
FitzHugh–Nagumo (FHN)
Model
Field programmable gate array (FPGA)

ABSTRACT

High accuracy implementation of biological neural networks (NN) is a task with high computational overheads, especially in the case of large scale realizations of neuromorphic algorithms. This paper presents a set of piecewise linear FitzHugh Nagumo (FHN) models, which can reproduce different behaviors, similar to the biological neuron. This paper presents a set of equations as a model to describe the mechanisms of a single neuron, which are implementable on digital platforms. Simulation results show that the model can reproduce different behaviors of the neuron. The proposed models are investigated, in terms of digital implementation feasibility and computational overhead, targeting low cost hardware realization. Hardware synthesis and physical implementations on FPGA show that the proposed models can produce a range of neuron behaviors with higher performance and lower implementation costs compared to the original model.

1. Introduction

During the past few decades, neuroscientists have been looking for new horizons to clarify how neural networks in the brain process information. Thus researchers in this field have investigated a variety of applications. To understand how the brain processes information, simulation and implementation of brain like networks are considered essential [1]. Motivated by biological discoveries, pulse coupled neural networks with spike timing are employed as a necessary component in biological information processing systems, such as the brain [1]. Many different models have been given for spiking neural networks, representing their dynamical behavior. These models are based on the biochemical behavior of the neuron structures and are usually modelled by a set of differential equations. Although detailed neuron models result in imitating most experimental measurements to a high degree of accuracy; due to their complexity, a majority are difficult to implement in large scale artificial spiking neural networks.

Many kinds of simplified models are presented for studies in the field of neural information coding, memory, and network dynamics. These models must reproduce the biological behavior of different types of neurons with higher performance and significantly lower hardware overhead compared to the original model. In general, there is a trade off between model accuracy and its computational complexity. For example, when it is required to understand how neuronal behavior depends on measurable physiological parameters, such as the maximal conductance, steady state activation/inactivation functions and time constants, the Hodgkin Huxley type [2] models are technically more suitable, but are computationally intensive and thus prohibit stimulations of large neuron populations. Since the influential work of Hodgkin and Huxley [3], there has been a continued interest in the dynamical systems perspective of a neuron. The Hodgkin Huxley, Hindmarsh Rose [4], and FitzHugh Nagumo (FHN) [5,6] models are the most successful dynamical models in computational neuroscience for capturing neural spiking behaviors. An intricate explanation of these and several other models can be found in [7]. The Hodgkin Huxley model, with four differential equations, is capable of creating all of the neural spiking behaviors but is highly nonlinear. Although the FHN model involves two differential equations, it is not more complex than the Hodgkin Huxley and Hindmarsh Rose models, does not use a reset or add noise [3]. In this paper, we present greatly simplified models that can exhibit most of the neural spiking behavior reproduced by well known dynamical systems models. Note that we use the term *complexity* to refer to the presence of redundancies in the model in addition to its capability of capturing neural spiking behaviors, and difficulty in parameter estimation. This neuron model has been commonly accepted as an accurate and computationally tractable model, while producing a wide range of cortical pulse coding behaviors. Implementation of this model, targeting different platforms, has been the subject of studies in terms of efficiency and large scale simulations based on optimal transfer capability of the

* Corresponding author.
E-mail address: ghkarimi@razi.ac.ir (Gh.R. Karimi).

http://dx.doi.org/10.1016/j.neucom.2015.03.084
0925-2312/

spike signals provided by addressing event representation [8,9]. There are three major approaches for addressing this challenge:

1. Analog implementations are considered to become a strong choice for direct implementation of neuro inspired systems [7,10,11]. In this approach, electronic components and circuits are employed to mimic neuronal dynamics. Due to its high performance and well developed technology, an analog VLSI implementation enables prototyping of neural algorithms to test theories of neural computation, structure, network architecture, learning and plasticity and also simulation of biologically inspired systems in a real time operation. Although these analog solutions are fast and efficient, requiring a long development time [1,12].
2. Special purpose hardware has been presented to implement neurobiological functions using software based systems for large scale simulations such as Blue Brain [13], Neurogrid [14], and SpiNNaker [15]. Even though these systems are flexible and biologically realistic having high performance, the presented hardware approaches suffer from limited programming capability and high cost. Unfortunately the cost and development time make these approaches impractical for public access, especially for large scale simulations of neuromorphic algorithms.
3. While analog neural chips inherently have limited programming capability, recently, reconfigurable digital platforms have been used to realize spiking neurons. This approach uses digital computation to emulate individual neural behaviors in parallel and distributed network architecture to implement a system level dynamic. Although digital computation consumes more silicon area and power per function in comparison with the analog counterpart, its development time is considerably lower and is not susceptible to power supply, thermal noise, or device mismatch. In addition, high precision digital computation makes it possible to implement networks with high dynamic range, greater stability, reliability, and repeatability.

Also simulation and implementation of a single neuron as well as large scale systems, in order to understand the brain behavior, have been the subject of numerous studies in recent years [14,16,17]. Therefore, efficient implementations of spiking neural network (SNN) systems are essential in this field of research. With this purpose in mind, this paper suggests a neuronal implementation based on the FitzHugh Nagumo (FHN) model, proposing a set of multiplierless models called MDL1 and MDL2. These models are computationally low cost compared with the original model and can be implemented on digital platforms such as field programmable gate arrays (FPGAs). To represent the neuron mechanisms, the FitzHugh Nagumo neuron model is considered. Simulation and implementation results demonstrate that the proposed models produce the same dynamics and can trace the original model in different states. The paper is organized as follows: background and mechanisms of the proposed neuron have been described in Section 2. Section 3 presents the modified neuron models, dynamics and phase plane trajectories in detail. Design and hardware implementation are discussed in Section 4. Section 5 presents implementation results. The paper concludes in Section 6.

2. Background

A simplified version of the Hodgkin Huxley model is the FHN model [5,6]. This model consists of several parameters, two equations and one condition, without the auxiliary reset equations, described given by

$$\begin{cases} \dot{v} = \dfrac{v^3}{3} \quad w+I \\ \dot{w} = \dfrac{1}{T}(a \quad bw+v) \end{cases} \tag{1}$$

where v represents the membrane potential of the neuron and w is the activation of K^+ ionic currents and inactivation of Na^+ ionic currents represents a membrane recovery variable. Here, a and c are scaling parameters, I is a constant stimulus current (input excitation current), τ is a constant parameter and all the variables are dimensionless. The spiking threshold, b, varies from 0 to 1, and to obtain electrical spiking, usually $b < 0.5$ is chosen [16]. Since hardware implementation requires lower area and power per function, power and cost are very important factors, so using multiplier is a costly approach for hardware implementation. For reducing multipliers, the proposed modified equation is suggested in the next section.

3. The proposed models

3.1. Mathematical approximation

1. *MDL1 and MDL2 models*: To improve the computational efficiency of the model, the proposed models can be considered for the neuron model. Dynamics of the proposed models can be given by the following equations defined as: MDL1:

$$\begin{cases} \dot{v} = v(\sqrt{3} \quad |v|) \quad w+I \\ \dot{w} = \dfrac{1}{T}(a \quad bw+v). \end{cases} \tag{2}$$

MDL2:

$$\begin{cases} \dot{v} = 0.3\sinh(v)(\sqrt{3} \quad |v|) \quad w+I \\ \dot{w} = \dfrac{1}{T}(a \quad bw+v). \end{cases} \tag{3}$$

Fig. 1 presents matching accuracy of proposed models. It is noted that, in MDL2 model, we can use hyperbolic calculation unit (HCU) for producing $2\sinh(v)$ function similar to the method presented in [18], according to Fig. 2. Multiply operations between term $2\sinh(v)$ in other sections of equation is shift and add operations, because $\sinh(v) = (2^v \quad 2^{(-v)})/2$ and can be implemented according to [18] strategy.
2. *System level implementation*: To investigate the functional behavior of MDL1, MDL2 and original models, they are modelled using VHDL AMS. To compare these models, we applied input signals with the same intensity and frequency to all of them. These comparisons are illustrated in Fig. 3.

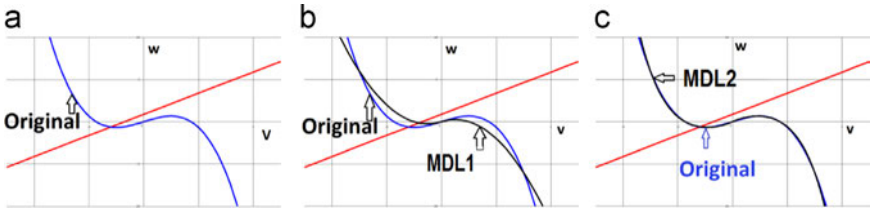

Fig. 1. Equilibrium v–w locus of the original and proposed models with these conditions a 0.7 mV, b 0.8 and τ 13 Ω. (a) Original nullclines. (b) MDL1 model corresponding to v equation. (c) MDL2 model corresponding to v equation. This figure shows how the variations of the input signal, i, changes the state of the nullclines and finally the trajectories of these equations.

3.2. Dynamics and phase plane trajectories

To present a more rigorous justification for the proposed models, the nullclines provide an insight into the similarity in dynamics in comparison with the original model.

1. *Dynamical system*: dynamical systems theory attempts describe systems that are constantly changing [20]. A dyna mical system is a set of variables where a deterministic rule describes the evolution of the state variables with time.
   The evolution rule of dynamical systems is given implicitly by a relation that gives the future state of the system [20]. Mathematical models describing the biological neuron models can be given as (time continuous differential equation)

$$\begin{cases} \dot{v} = F(v,w)+c \\ \dot{w} = G(v,w)+d. \end{cases} \tag{4}$$

   For original and proposed models, we have

$$\text{original}: \begin{cases} \dot{v} = v \quad \frac{v^3}{3} \quad w+I \\ \dot{w} = \frac{1}{T}(a \quad bw+v) \end{cases} \tag{5}$$

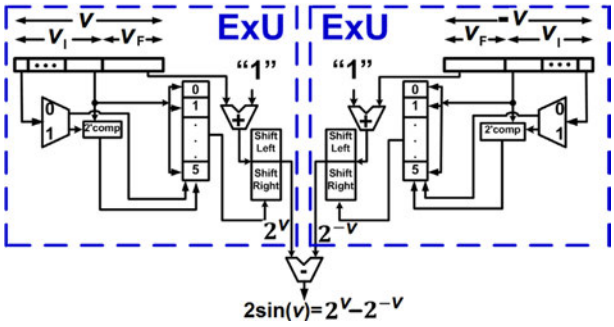

Fig. 2. Hyperbolic calculation unit (HCU). Here, v and $-v$ are the inputs of the Exponential block (ExU) that are composed of two parts for each of them [18]. The sign bits have determinant roles for the computation.

$$\text{MDL1}: \begin{cases} \dot{v} = v(\sqrt{3} \quad |v|) \quad w+I \\ \dot{w} = \frac{1}{T}(a \quad bw+v) \end{cases} \tag{6}$$

$$\text{MDL2}: \begin{cases} \dot{v} = 0.3\sinh(v)(\sqrt{3} \quad |v|) \quad w+I \\ \dot{w} = \frac{1}{T}(a \quad bw+v). \end{cases} \tag{7}$$

2. *Phase portrait*: A phase portrait is a geometric representation of the trajectories of a dynamical system in the phase plane. Each set of initial conditions is represented by a different curve, or point [20].
3. *Equilibrium points*: Mathematically, the point $\hat{v} \in \mathrm{R}^n$ is an equilibrium point for the following differential equation:

$$\begin{cases} \frac{dv}{dt} = F(v,w)+c \\ \frac{dw}{dt} = G(v,w)+d \end{cases} \tag{8}$$

   if $F(t,\hat{v}) = G(t,\hat{v}) = 0$ for any time (t) [20].
4. To find the type of Equilibrium points in both linear and non linear systems, we need to find Jacobian matrix [20]. Assum ing a 2 D dynamical system,

$$\text{original}: \begin{cases} \dot{v} = v \quad \frac{v^3}{3} \quad w+I \\ \dot{w} = \frac{1}{T}(a \quad bw+v) \end{cases} \tag{9}$$

$$J(v,w) = J(v_{\mathrm{eq}}, w_{\mathrm{eq}}) = \begin{bmatrix} \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial w} \\ \frac{\partial \dot{w}}{\partial v} & \frac{\partial \dot{w}}{\partial w} \end{bmatrix} \tag{10}$$

$$J(v,w) = J(v_{\mathrm{eq}}, w_{\mathrm{eq}}) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{11}$$

where A, B, C, D are the coefficients of the Jacobian matrix. So for original and proposed models:

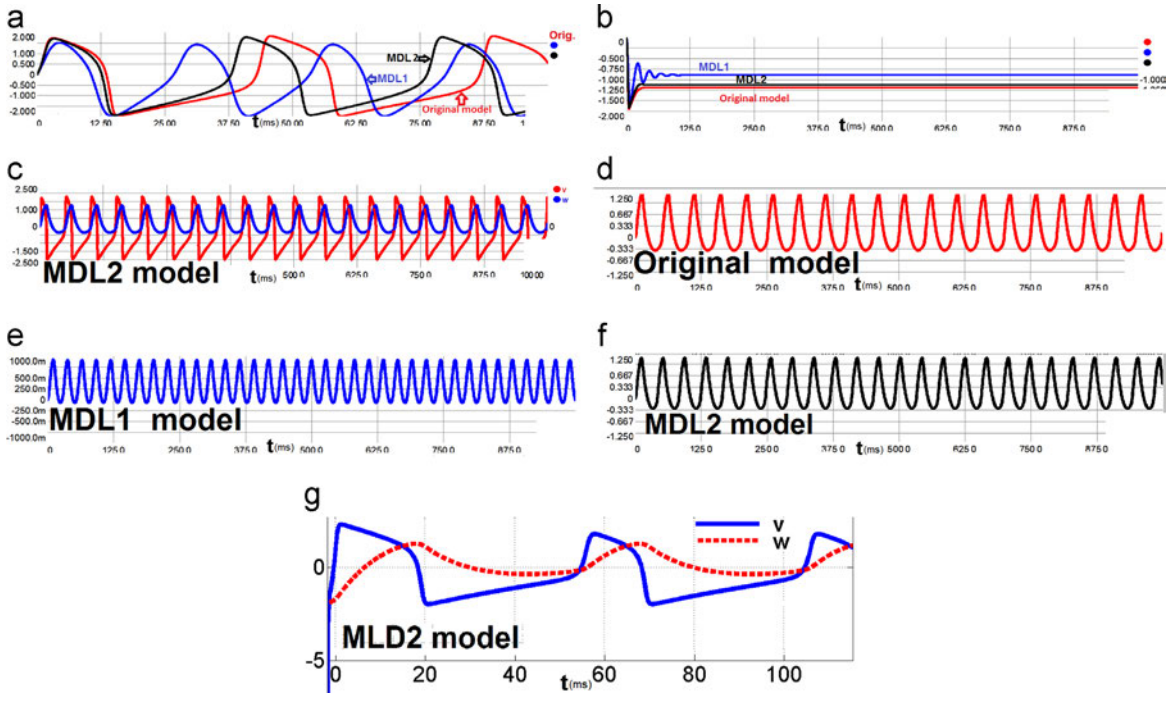

Fig. 3. Output of the original and proposed models. (a) Membrane potential (I 0.4 mA). (b) Membrane potential (I 0 mA). (c) Current variable and membrane potential of MDL2 model. (d) Membrane potential of the MDL1 model. (e) MDL2 model when ($I > 0$). (g) (c) Current variable and membrane potential of MDL2 model (I 0.33 mA).

$$J(v_{\text{original}}, w_{\text{original}}) = J(v_{\text{eq}}, w_{\text{eq}}) = \begin{bmatrix} A = 1 & v^2 & B = & 1 \\ C = \frac{1}{\tau} & & D = & \frac{b}{\tau} \end{bmatrix} \quad (12)$$

$$J(v_{\text{MDL1}}, w_{\text{MDL1}}) = J(v_{\text{eq}}, w_{\text{eq}}) = \begin{bmatrix} A = \sqrt{3} + |v|(v & 1) & B = & 1 \\ C = \frac{1}{\tau} & & D = & \frac{b}{\tau} \end{bmatrix} \quad (13)$$

$$J(v_{\text{MDL1}}, w_{\text{MDL1}}) = J(v_{\text{eq}}, w_{\text{eq}}) \begin{bmatrix} A & 0.3\cosh(v)(\sqrt{3} - |v|) + 0.3(|v|)\sinh(v) & B & -1 \\ C & \frac{1}{\tau} & D & \frac{-b}{\tau} \end{bmatrix}. \quad (14)$$

5. *Eigenvalues*: Eigenvalues can be obtained using the following equations:

$$\begin{cases} T = A + B \\ Z = AD \quad BC \\ P(\lambda) = \lambda^2 \quad T\lambda + Z \end{cases} \quad (15)$$

where A, B, C and D are the coefficients of the Jacobian matrix. By solving these equations, we find two real eigenvalues equal to (λ_1, λ_2). So for the original and proposed models, we have

$$\text{original}: \begin{cases} T = & v^2 \\ Z = \frac{b(v^2 \quad 1)}{\tau} + \frac{1}{\tau} \\ P(\lambda) = \lambda^2 + v^2\lambda + \frac{b(v^2 \quad 1)}{\tau} + \frac{1}{\tau} \end{cases} \quad (16)$$

$$\text{MDL1}: \begin{cases} T = \sqrt{3} + |v|(v \quad 1) \quad 1 \\ Z = \frac{b(\sqrt{3} + |v|(v \quad 1))}{\tau} + \frac{1}{\tau} \\ P(\lambda) = \lambda^2 \quad (\sqrt{3} + |v|(v \quad 1) \quad 1)\lambda + \frac{b\sqrt{3} + |v|(v \quad 1)}{\tau} + \frac{1}{\tau} \end{cases} \quad (17)$$

$$\text{MDL2}: \begin{cases} T = 0.3\cosh(v)(\sqrt{3} \quad |v|) + 0.3(|v|)\sinh(v) \quad 1 \\ Z = \frac{0.3b(\cosh(v)(\sqrt{3} \quad |v|) + 0.3(|v|)\sinh(v))}{\tau} + \frac{1}{\tau} \\ P(\lambda) = \lambda^2 \quad (0.3\cosh(v)(\sqrt{3} \quad |v|) + 0.3(|v|)\sinh(v) \quad 1)\lambda + \frac{0.3b(\cosh(v)(\sqrt{3} \quad |v|) + 0.3(|v|)\sinh(v))}{\tau} + \frac{1}{\tau}. \end{cases} \quad (18)$$

Therefore for 2 D dynamical system; we have λ_1, λ_2 and, depend ing on these points, the type of equilibrium points will be determined [19].

This analysis shows that the behavior of the proposed models (especially MDL2), without input, is quite similar to the original model as shown in Fig. 4 (1 (a) 1 (c)). In addition, it can be seen that the origin is a globally stable equilibrium point of the total system. Thus, for all values of I, the proposed models are globally stable. We have the same type of equilibrium points. So, similar to the original model, there are stable states in the proposed models that are shown in Fig. 4 (2 (a) 5 (c)). Thus it is verified that the proposed models can replace the original model.

4. Design and hardware implementation

The proposed system architecture for MDL2 model is presented in Fig. 5, which includes system block diagram. This architecture is described in more detail in the following subsections.

4.1. Equations discretization

Each design is included of two blocks to calculate v and w in models as indicated in their corresponding equations. In the first step for implementation, it is essential to discretize equations. In this research, we have utilized the Euler method as

$$\text{original}: \begin{cases} v[n+1] = v[n] + dt(v[n] \quad (v[n] \times v[n] \times v[n])/3 \quad w[n] + I[n]) \\ w[n+1] = w[n] + dt((v[n] + a \quad bw[n])/\tau). \end{cases} \quad (19)$$

This follows similarly for MDL1 equations which are given by

$$\text{MDL1}: \begin{cases} v[n+1] = v[n] + dt(v[n](\sqrt{3} \quad v[n]) \quad w[n] + I[n]) \\ w[n+1] = w[n] + dt((v[n] + a \quad bw[n])/\tau). \end{cases} \quad (20)$$

The MDL2 are given by

$$\text{MDL2}: \begin{cases} v[n+1] = v[n] + dt(0.3\sinh(v)(\sqrt{3} \quad v[n]) \quad w[n] + I[n]) \\ w[n+1] = w[n] + dt((v[n] + a \quad bw[n])/\tau) \end{cases} \quad (21)$$

where dt represents time step in the Euler method. Table 1 verifies these discrete equations and the number of stages in pipeline implementation.

4.2. Bit width determination and the proposed architecture

To determine the bit width of the variables and parameters, two basic points must be taken into account. The first is the minimum/maximum bounds of the values at each point of the hardware structure and the second is the minimum/maximum required logic shifts either to right or left during calculations. In this structure, v has almost a range from 2 to 2 for tonic neurons and these values need 3 bits. The other variables and parameters need fewer than v for their value ranges but to avoid any overflow and also increasing accuracy of the calculations, a bit width of 20 has been considered where 10 bits for the integer part and 10 bits are for the fractional part.

4.3. N unit

As shown in Fig. 6, this unit is a digital multiplierless imple mentation of the neuron model as described in Eq. (21). This unit includes v_{pipeline}, w_{pipeline}, v_{buffer} and w_{buffer}. v_{pipeline}, w_{pipeline} are v and w in Eq. (22) respectively. These are implemented in a pipeline structures with v_{stage} and w_{stage} stages, where v_{buffer} and w_{buffer} are the buffer registers of the v and w values and $v_{\text{buffer}-\text{size}}$ and $w_{\text{buffer}-\text{size}}$ are the size of v and w states. With every rising edge of the clock, these buffers are shifted and get 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 these arith metic modules are in pipeline state, which means we need a register after arithmetic state in order to save data. Accordingly,

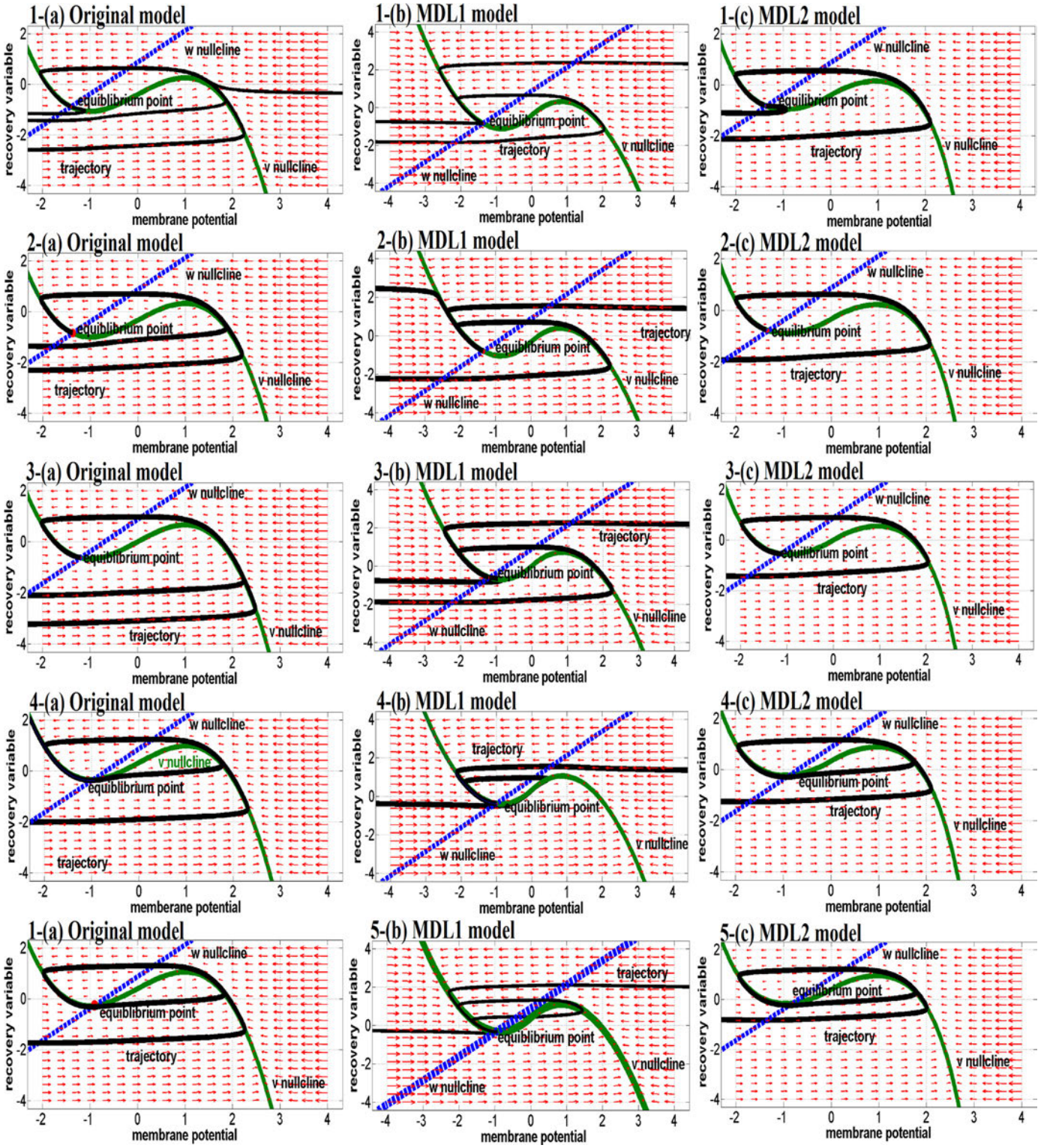

Fig. 4. v–w curve of original model, MDL1 and MDL2 models and nullclines of these systems with increasing the current for seven states of (a)–(g) with the following conditions a 0.7 mV, b 0.8 and τ 13 Ω. This figure shows how the increasing of the input signals (i) changes the state of the nullclines and finally trajectories of these equations.

following conditions must be satisfied:

$$\begin{cases} N = v_{\text{buffer-size}} + v_{\text{stage}} = w_{\text{buffer-size}} + w_{\text{stage}} \\ v_{\text{buffer-size}} = w_{\text{buffer-size}} \\ v_{\text{stage}} = w_{\text{stage}} \end{cases} \tag{22}$$

where N is the number of neurons. We define the stage delay in order to synchronize all the equations, for the case of unequal numbers of stages. In each time sample, average values of v and w are calculated and in the next time sample, they are applied as inputs. This guarantees that the final neuron provides an output with appropriate timing. After finding the average value, the first neuron is upgraded to obtain new value in the next rising edge clock. This condition is provided by applying neuron input at the last pipeline stage. This process omits the average computing delay that is upgraded in the next time sample (for applying to the

neuron). For an improved comparison between original, MDL1 and MDL2 models, the scheduling diagrams (Data Flow Graph with scheduling control steps) based on Eqs. (19) (21) are drawn in Fig. 7. The number of minimum resources in the scheduling of the original and proposed models is presented in Table 2.

4.4. Control unit

The control unit (CU) is responsible for two major tasks: (1) providing suitable signal for the output provider unit in order to select suitable neuron which is selected by K bit input of the user defined number applied to the control unit. (2) Providing suitable signal for average computing unit in order to calculate the average value of neuron's output. The architecture of the control unit is presented in Fig. 8. The CU equipped with a counter that acts as a pointer and fetches the neuron number. For rising edge of the clock pulse, this counter is increased by one and when the counter reaches N this yields the reset time. The comparator checks the value of the counter register as a `pointer register` and when the counter is at N, the time is reset. The reset instruction points to the first neuron and sends this to the average computing unit as an instruction. Another comparator compares the value of `pointer register` with the selective neuron number that is the `selector register` and sends a suitable instruction to the Output Provider Unit. The `selector register` is updated by enabling the `write enable` signal.

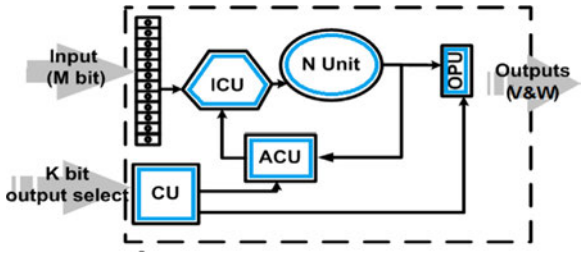

Fig. 5. The proposed system architecture (system block diagram). Abbreviations: neuron unit (N Unit), input computing unit (ICU), output provider unit (OPU), control unit (CU), average computing unit (ACU).

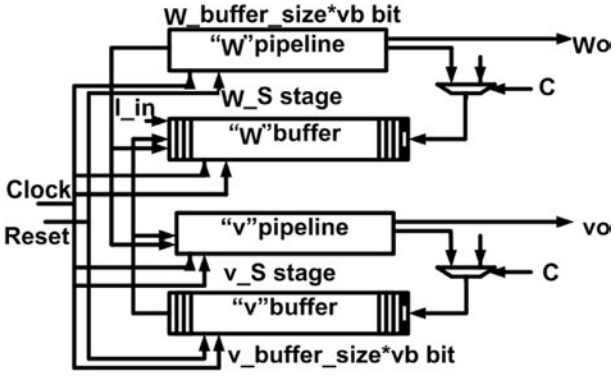

Fig. 6. General overview of N unit. (pip. pipeline, buf. buffer).

4.5. Average computing unit

The average computing unit (ACU) computes v and w state variables. The architecture of the ACU is presented in Fig. 9. The ACU performs the task by reading the `accumulator register` with a serial accumulator repetitively and accumulates it with the v and w variables of the neuron unit and the outcome returns to the register in every clock pulse. The `accumulator register` is storing of N 1 number of v and w input variables at the $(N\ 1)^{th}$ clock pulse after the latest reset. Therefore, the adder's output is the final addition of N variables of v and w at the mentioned clock pulses. This amount must be saved in `auxiliary register` after N clock pulses in order to apply neuron model in the next time sample. The value of the `auxiliary register` must be divided by N average values that are sent to the input computing unit. We have selected N as a power of 2 in order to replace the multiplication operation by a shift operation.

4.6. Input computing unit

The input computing unit (ICU) is designed as the structure shown in Fig. 10(a), caches the input, obtains v and w variable averages from the input port and ACU and then computes the difference. Afterward, it multiplies the result by constant K. The ICU must perform this operation without any delay. After computing a new average that is produced by average computing unit, it should give the updated input to the neuron immediately. Therefore, this unit is designed via a combination of gates and does not include any register. We should notice that the implementation complexity of the K constant multiplier can be substantially decreased by selecting suitable value for K. If a suitable value of K is chosen, one may implement it with pairs of accumulators and shift logic operations. In this paper, we select K in such a way that multiplication can be replaced by simple shift operation.

4.7. Output provider unit

The output provider unit (OPU), shown in Fig. 10(b), saves the v and w variables in a register called the `output register` in N clock pulses. The control unit provides writing activation for this register in order to save valid v and w variables. The OPU also transforms saved digital numbers in the `output register` to an analog signal via a Digital Analog Converter (DAC).

5. Results

Neuron behaviour is described with VHDL code. In these simulations, "v" and "w" are 20 bits wide that are used in fixed point calculations. Neuron parameters are chosen as $a=0.7$, $b=0.8$, $\tau=13$. Also initial values are $v_0=\frac{1}{1024}$ and $w_0=\frac{1}{1024}$, so that 10 bits are for the integer part and 10 bits for the fractional part. To

Table 1
Discrete equations of v and w for original and modified models, and number of stages in pipeline implementation.

Models	Output variable	Discrete equation	Pipeline stage
Original model	v	$v[n+1]\quad v[n]+dt(v[n]-(v[n]\times v[n]\times v[n])/3-w[n]+I[n])$	6
Original model	w	$w[n+1]\quad w[n]+dt((v[n]+a-bw[n])/\tau)$	4
MDL1 model	v	$v[n+1]\quad v[n]+dt(v[n](\sqrt{3}-v[n])-w[n]+I[n])$	6
MDL1 model	w	$w[n+1]\quad w[n]+dt((v[n]+a-bw[n])/\tau)$	4
MDL2 model	v	$v[n+1]\quad v[n]+dt(0.3\sinh(v)(\sqrt{3}-v[n])-w[n]+I[n])$	6
MDL2 model	w	$w[n+1]\quad w[n]+dt((v[n]+a-bw[n])/\tau)$	4

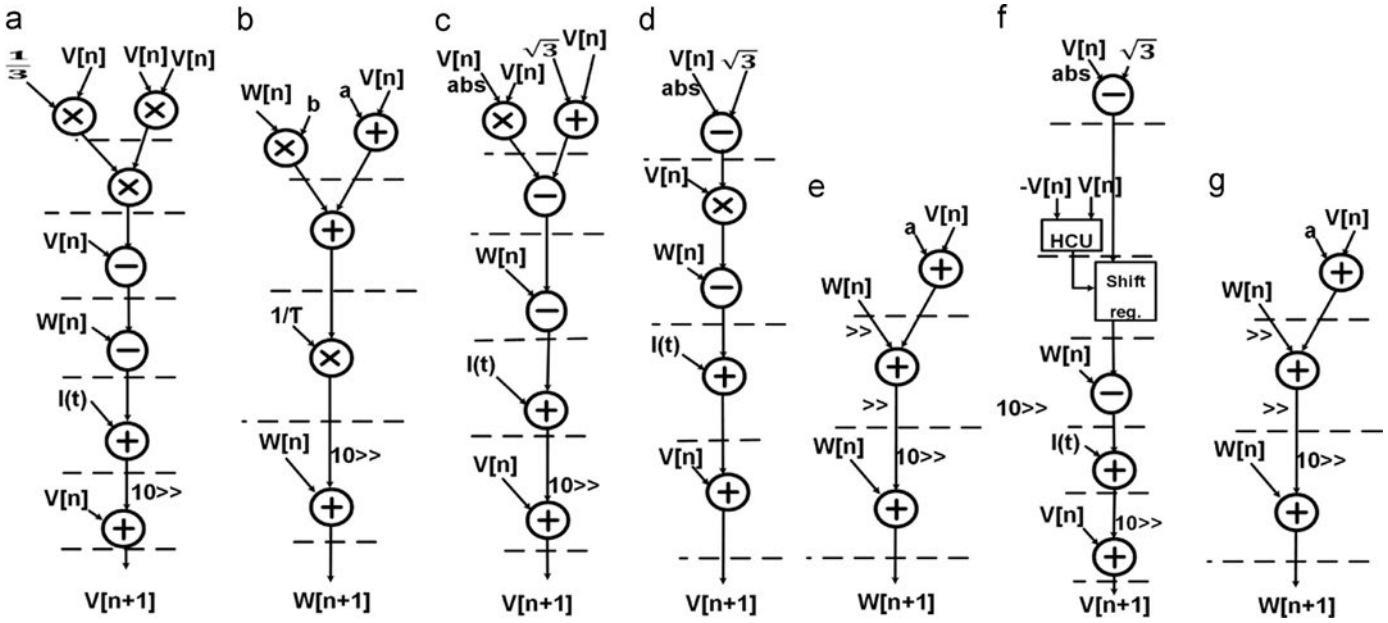

Fig. 7. Arithmetic pipelines. (a) v Pipeline in original model. (b) w Pipeline in original model. (c) v Pipeline in MDL1 model. (d) v Pipeline in MDL1 model. (e) w Pipeline in MDL1 model. (f) v Pipeline in MDL2 model. (g) w Pipeline in MDL2 model.

Table 2
The number of minimum resources in the scheduling of v and w.

Resources	Output	Original model	MDL1 model	MDL2 model
Adder	v	4	4	7
Multiplier	v	3	1	–
Multiplexer	v	–	–	2
Adder	w	3	3	3
Multiplier	w	2	–	–
Multiplexer	w	–	–	–

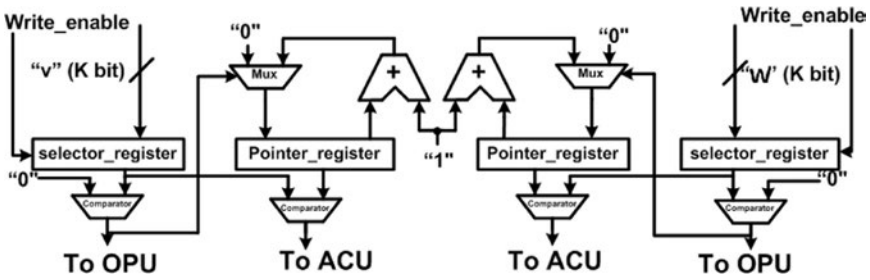

Fig. 8. Control unit (CU) of the proposed model. Abbreviations: comparator (Com.), average computing unit (ACU), output provider unit (OPU), Write_enable (W_en).

complete the comparison between the original model and the proposed models, we applied the same input signals as driving stimulus to all of them. Results are shown in Fig. 3. Responses of the proposed models have acceptable similarity to the original model. For measurement of hardware resources used in this project, we sensitize the design that is mentioned in the last sections, and the necessary resources for a single neuron that is realized in both of MDL1 and MDL2 models in pipeline configura tion are implemented. The proposed circuit is implemented on a XILINX Virtex II Pro development board. Fig. 11 displays oscillo scope photograph of MDL2 model implementation. As observable, MDL2 model has excellent accuracy and low cost implementation. Table 3 shows the synthesis results with the optimization goal.

6. Conclusion

This paper demonstrates the MDL1 and MDL2 models, which have the same characteristics as a full FitzHugh Nagumo implemen tation. Our system level simulation results indicate that the MDL1 and MDL2 models have sensitivity to both amplitude and frequency of input signal. In order the achieve a reduction in hardware and computational overhead, a multiplierless hardware structure has been proposed and implemented. These models are applicable for future large scale implementations. These systems can be imple mented on accelerated FPGA hardware. The MDL1 model has lower precision but higher hardware cost. The MDL2 model advantages are excellent accuracy, dynamically tracks the original model and is not

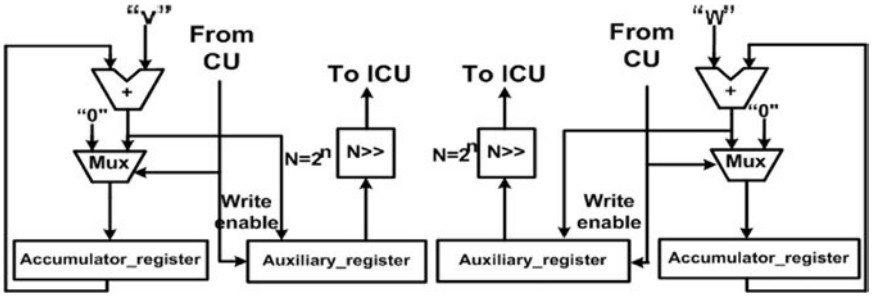

Fig. 9. General structure of the average computation unit (ACU). Abbreviations: Accumulator_register (Ac._reg.), Auxiliary_register (Aux._reg.), Write_enable (W_en).

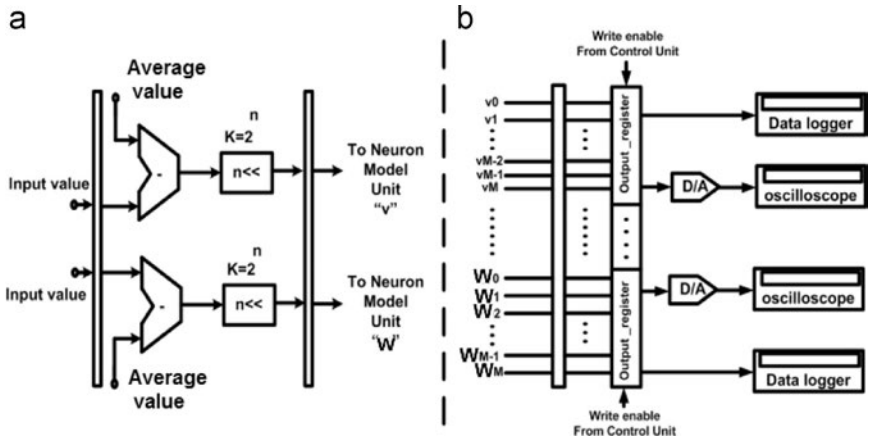

Fig. 10. General overview of input computation unit and output provider unit. (a) Input computation unit. (b) Output provider unit.

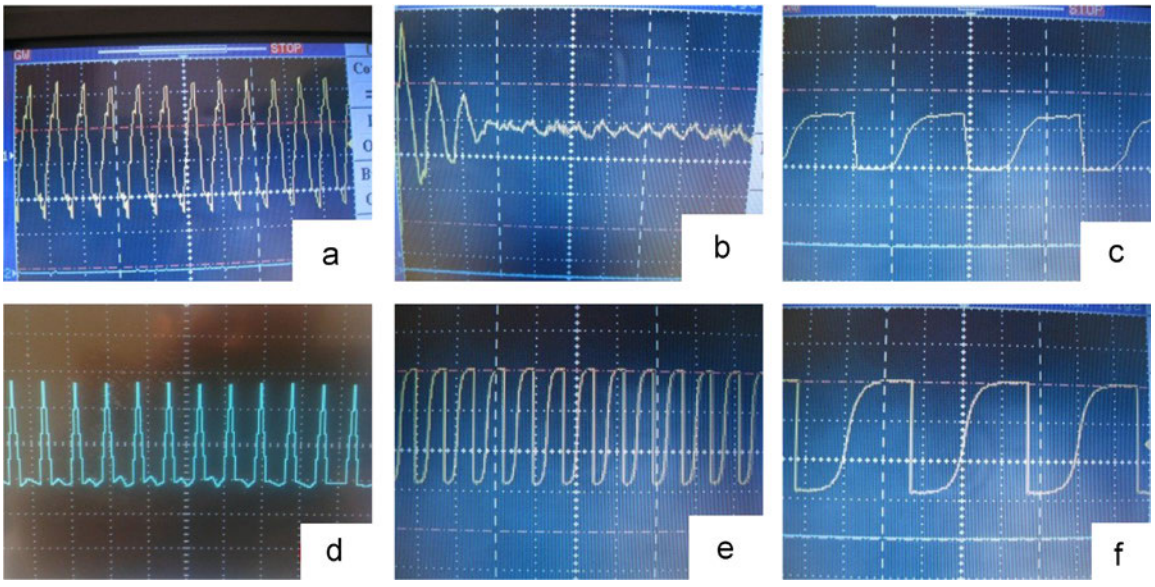

Fig. 11. Output of the MDL2 model implemented on XILINX Virtex-II Pro XC2VP30. (a) Tonic spiking implementation. (b) Adaptation implementation. (c) Tonic spiking implementation (I 27 mA). (d) Tonic spiking implementation (I 33 mA). (e) Delayed regular tonic implementation (I 40 mA). (d) Tonic spiking implementation. The horizontal axis denotes time, and the vertical axis shows voltage (time scale 100 ms and voltage scale 5 mV).

Table 3
Device utilization summary.

Logic utilization	Used	Available	Utilization (%)
Number of slices	648	1408	46
Number of slices flip flops	526	2816	18
Number of 4 input LUTs	1085	2816	38
Number of bonded IOBs	86	140	61
Number of MUL18 × 18 s	0	0	0
Number of GCLKs	1	16	6

as costly as MDL1 model. Also the dynamic behavior of the neuron has been examined for all of models and our results demonstrated the desired performance for the models, especially the MDL2 model. Its biggest advantage of the MDL2 model is that its behavior is close to a biological neuron. These results give rise to the possibility of employing MDL1 and MDL2 neurons for performing a wide range of signal processing applications.

References

[1] H. Soleimani, A. Ahmadi, M. Bavandpour, Biologically inspired spiking neurons: piecewise linear models and digital implementation, IEEE Trans. Circuits Syst. I, Regul. Pap. 59 (12) (2012) 2991–3004.

[2] A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 117 (August (4)) (1952) 500–544.

[3] R.T. Faghih, K. Savla, M.A. Dahleh, E.N. Brown, Broad range of neural dynamics from a time-varying FitzHugh–Nagumo model and its spiking threshold estimation, IEEE Trans. Biomed. Eng. 59 (March (3)) (2012) 816–823.

[4] R.M. Rose, J.L. Hindmarsh, The assembly of ionic currents in a thalamic neuron I. The three-dimensional model, Proc. R. Soc. B: Biol. Sci. 237 (1288) (1989) 267–288.

[5] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophys. J. 1 (6) (1961) 445–466.

[6] J. Nagumo, S. Arimoto, S. Yoshizawa, An active pulse transmission line simulating nerve axon, Proc. Inst. Radio Eng. 50 (10) (1962) 2061–2070.

[7] E.M. Izhikevich, Which model to use for cortical spiking neurons, IEEE Trans. Neural. Netw. (Spec. Issue Temporal Cod.) 15 (September (5)) (2004) 1063–1070.

[8] C. Morris, H. Lecar, Voltage oscillations in the barnacle giant muscle fiber, Biophys. J. 35 (1981) 193–213.

[9] R. Jolivet, V. Rauch, H.R. Lüscher, W. Gerstner, Integrate-and-Fire models with adaptation are good enough: predicting spike times under random current injection, in: Y. Weiss, B. Schölkopf, J. Platt (Eds.), Advances in Neural Information Processing Systems, vol. 18, MIT Press, Cambridge, MA, 2006, pp. 595–602.

[10] M.G.F. Fuortes, F. Mantegazzini, Interpretation of the repetitive firing of nerve cells, J. Gen. Physiol. 45 (July (6)) (1962) 1163–1179.

[11] M.R. Azghadi, N. Iannella, S.F. Al-Sarawi, G. Indiveri, D. Abbott, Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges, Proc. IEEE 102 (5) (2014) 717–737.

[12] W. Gerstner, R. Brette, Adaptive exponential integrate-and-fire model, Scholarpedia 4 (6) (2009) 8427.

[13] L. Badel, S. Lefort, R. Brette, C. Petersen, W. Gerstner, M.J. Richardson, Dynamic $I-V$ curves are reliable predictors of naturalistic pyramidal-neuron voltage traces, J. Neurophysiol. 99 (December (2)) (2007) 656–666.

[14] Blue Brain Project, Available Online at: ⟨http://bluebrain.epfl.ch⟩, February 2013.

[15] Neurogrid Project, Available Online at: ⟨http://neurogrid.net⟩, February 2013.

[16] S.B. Furber, S. Temple, A.D. Brown, High- performance computing for systems of spiking neurons, in AISB'06 Workshop on GC5: Architecture of Brain and Mind, vol. 2, April 2006, pp. 29–36.

[17] H.C. Tuckwell, R. Rodriguez, F.Y.M. Wan, Determination of firing times for the stochastic FitzHugh–Nagumo neuronal model, Neural Comput. 15 (1) (2003) 143–159.

[18] S. Gomar, A. Ahmadi, Digital multiplierless implementation of biological adaptive-exponential neuron model, IEEE Trans. Circuits Syst. I 61 (April (4)) (2013) 1206–1219.

[19] E.M. Izhikevich, Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting (Computational Neuroscience), MIT Press, Cambridge, MA, 2006.

[20] W. Gerstner, W.M. Kistler, Spiking Neuron Models Single Neurons, Populations, Plasticity, Cambridge University Press, UK, August 2002.

M. Nouri received the B.Sc. degree in electrical engineering from Azad University, Kermanshah, Iran, in 2012, and the M.Sc. degree in electronics engineering from the Department of Electrical Engineering, Razi University, Kermanshah, Iran, in 2014 (with Honors), where, he is currently working toward the Ph.D.degree in electronics engineering. 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, artificial cochlear and integrated circuit design.

Gholamreza Karimi was born in Kermanshah, Iran in 1977. He received the B.S. and M.S. and Ph.D. degrees in electrical engineering from Iran University of Science and Technology (IUST) in 1999, 2001 and 2006 respectively. He is currently an Associate Professor in Electrical Department at Razi University, Kermanshah, since 2007. His research interests include low power analog and digital IC design, neuromorphic VLSI, RF IC design, modeling and simulation of RF mixed signal IC. He is also interested in microwave devices and artificial intelligence systems.

A. Ahmadi received the B.Sc. and M.Sc. degrees in electronics engineering from Sharif University of Technology and Tarbiat Modares University, Tehran, Iran, in 1993 and 1997, respectively, and the Ph.D. degree in electronics from the University of Southampton, U.K., in 2008. From 2008 to 2010, he was a Fellow Researcher with the University of Southampton. He is currently an Associate Professor in the Electrical Engineering Department, Razi University. His current research interests include bio-inspired computing, neuromorphic, hardware implementation of signal processing systems, high-level synthesis and memristors.

D. Abbott 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, NY, USA: 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.

He is a Fellow of the Institute of Physics (IOP) and a Fellow of the IEEE. He has won a number of awards including the South Australian Tall Poppy Award for Science (2004), the Premier's SA Great Award in Science and Technology for outstanding contributions to South Australia (2004), and an Australian Research Council (ARC) Future Fellowship (2012). With his colleagues, he won an IEEE Sensors Journal best paper award (2014). 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, PLOSONE, and is currently on the editorial boards of the Proceedings of the IEEE, the IEEE Photonics Journal, and Nature's Scientific Reports.