

## Multiple perspectives on the hardware implementations of biological neuron models and programmable design aspects

Nimet KORKMAZ\*, İsmail ÖZTÜRK, Recai KILIÇ

Department of Electrical & Electronic Engineering, Faculty of Engineering, Erciyes University, Kayseri, Turkey

Received: 02.09.2013

Accepted/Published Online: 30.06.2014

Final Version: 23.03.2016

**Abstract:** A few individual design examples of programmable device-based biological neuron model implementations are available in the literature, but there is no comprehensive study that examines analog and digital programmable design examples of frequently studied biological neuron model implementations. The aim of this paper is to present a comparative study about programmable and reconfigurable implementations of the FitzHugh–Nagumo, Izhikevich, and Hindmarsh–Rose neuron models. Since field programmable analog array (FPAA) and field programmable gate array (FPGA) devices offer several advantages such as flexible design possibilities, reduction of the complexity of design, real-time modification, and software control for programmable and reconfigurable implementations of neuron models and neural structures, they are preferred in these implementations as analog and digital programmable devices. Experimental results agree with the numerical simulations and verify the design effectiveness of FPAA- and FPGA-based implementations.

**Key words:** Neuron models, field programmable analog array, field programmable gate array, circuit implementation

### 1. Introduction

Several biological neuron models have been reported in the literature for the purpose of examining parameter changes, which affect the behaviors, physiological structures, and information transfer mechanisms of the neurons, or for studying the interactions between neurons outside the living body. The first neuron model was the classical neuron model proposed by McCulloch and Pitts in 1943 [1]. Several assumptions, such as the “all or nothing” principle, were made in this model to facilitate easy calculation, and these assumptions constitute a basis for the other models. In 1952, the most successful and widely used neuron model, the Hodgkin–Huxley (HH) model, was presented in the literature [2]. The ionic mechanism and the electrical current on the membrane surface were discussed in this comprehensive model. After that, the FitzHugh–Nagumo (FHN) neuron model, which is a simplified version of the HH neuron model, was proposed [3]. In 1972, Nagumo and Sato [4] defined a neuron model, and the weakly coupled Wilson–Cowan neuron model was defined in the same year [5]. The Morris–Lecar (ML) neuron model is a conductance-based neuron model and it was proposed in 1981 [6]. In order to describe oscillations in barnacle giant muscle fiber, the ML neuron model has biological significance. The Hindmarsh–Rose (HR) neuron model was proposed in 1984 [7]; it has both several neuronal behaviors and an accurate output frequency–input current relationship. In addition to having a simple mathematical description, the Izhikevich neuron model includes very rich neuronal dynamics, as does the HH neuron model,

\*Correspondence: [nimetdahasert@erciyes.edu.tr](mailto:nimetdahasert@erciyes.edu.tr)

and it was proposed in 2003. Besides these models [1–8], there are integrate-and-fire type models [9]; they only describe the electrical activities of neurons and do not deal with the electrophysiology of them. The activities of biological neuron models are tested by utilizing simulation tools or experimental setups. While software examinations of biological neuron models simulate the neuronal features, hardware implementations emulate these features and the real-time signals can be used in studies such as real-world robotic applications and central pattern generators [10–12]. For these reasons, although the required number of components depends on the complexity of the biological neuron model, it is preferred to implement biological neuron models by using the minimum number of components and allowing a high precision parameter setup in a most accurate way. To implement the biological neuron models, there are design techniques such as individual application-specific integrated circuit (ASIC) implementations and discrete circuit applications in the literature [13–24]. Very-large-scale integration (VLSI)-based realizations of neuron models present a compact structure, and a simple network consisting of a few neurons can be constructed in a very small area on the chip. However, they do not offer reconfigurability features. To obtain rapid prototyping in practical applications, discrete circuit-based implementations of neurons are highly efficient, but as the order of nonlinearity increases, it is necessary to adjust the parameters of the neuron model more sensitively. As the neuron number in the network increases, the circuit to be realized also gets more complicated and the necessity for a new prototype occurs for each model. Thus, discrete circuit-based implementations become inadequate to implement a neural network structure that consists of several neurons. For these reasons, field programmable analog array (FPAA) and field programmable gate array (FPGA) devices provide more practical usage and rapid prototyping features than ASIC and discrete implementations for examining neural interactions such as the adjustment of weights and coupling of neurons. Their reconfigurability features allow us to realize different neuron models without the need for any hardware changes. Because of these useful features of FPAA and FPGA, they can be used effectively to realize neuron models. In this study, first some overviews are presented about neuron models and then the FHN, Izhikevich, and HR neuron models are implemented with both FPAA and FPGA devices to evaluate their neuron model realization performances in terms of design flexibility, power consumption, area consumption, dynamic range, and frequency range. Finally, their programmable design aspects are compared in the framework of alternative hardware techniques.

In this context, the frequently studied biological neuron models in the literature are overviewed in Section 2. In Section 3, the analog and digital programmable implementation details of the FHN, Izhikevich, and HR neuron models are given. Discussions and conclusions about the implementation methods of neuron models are presented in Sections 4 and 5.

## 2. An overview of the biological neuron models

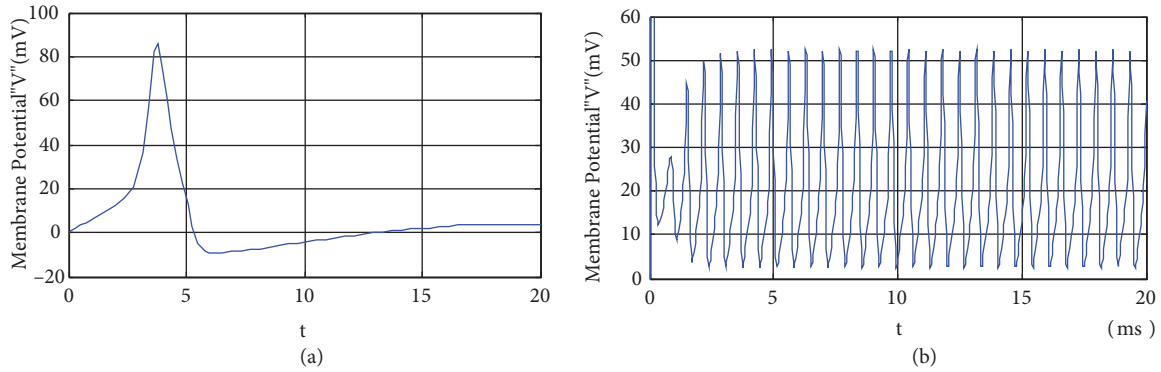
A real biological neuron exhibits a wide variety of dynamical behaviors, such as quiescent, spike, burst, and chaotic behaviors. These rhythmic behaviors are called neuron oscillation and they can be defined by mathematical equations similar to oscillators. A biological neuron model is a structure that defines neuron behaviors with mathematical equations. In this section, well-known biological neuron models in the literature are tested via a numerical software tool for the purpose of checking the accuracy of the experimental results by comparing the numerical results.

The HH neuron model is a detailed model that gives accurate results about the electrical characteristics of a neuron. In particular, it explains the production and propagation of action potential by using the ionic concentration changes and the descriptions of ionic channel structures [2]. The HH neuron model is defined

by Eq. (1) in Table 1. In Eq. (1),  $E_K$ ,  $E_{Na}$ , and  $E_L$  are equilibrium potentials that depend on the ionic species.  $g_K$ ,  $g_{Na}$ , and  $g_L$  (mS/cm<sup>2</sup>) are the ion conductances and  $V$ ,  $C$ , and  $I$  are the membrane potential, membrane capacitance, and membrane input current, respectively. The numerical simulation results of this model are given in Figure 1. Figure 1a represents the action potential of the neuron. Here, the resting potential of the neuron is about 5 mV. The figure shows the depolarization, repolarization, and hyperpolarization states at ranges of 0–3 ms, 6–16 ms, and 6–16 ms, respectively. An increment in the frequency of spikes with the increment of membrane input current is observed, as shown in Figure 1b.

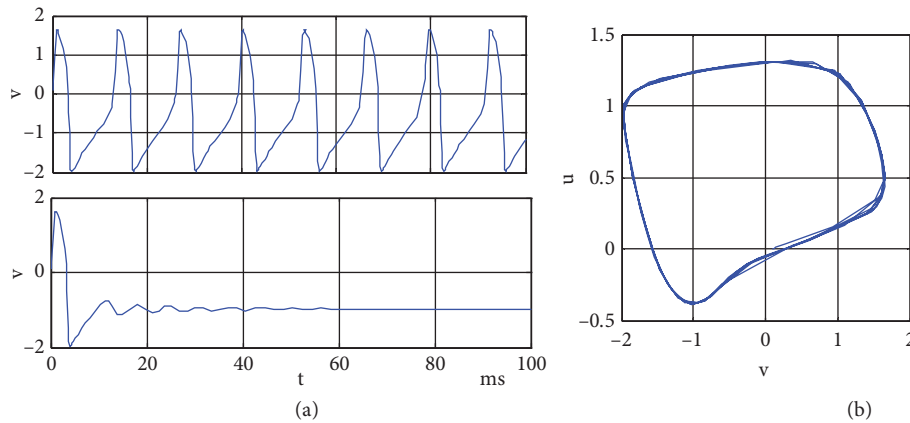
**Table 1.** A description table of well-known biological neuron models in the literature.

Neuron model	Equations	Parameters	Eq. no.						
HH neuron model	$C_M \dot{V} = I - \bar{g}_K n^4 (V - E_K) - \bar{g}_{Na} m^3 h (V - E_{Na}) - \bar{g}_L (V - E_L)$ $\dot{n} = \alpha_n (1 - n) - \beta_n n, \quad \alpha_n = \frac{0.01(10 - V)}{e^{(10 - V)/10} - 1}, \quad \beta_n = 0.125e^{(-V/80)}$ $\dot{m} = \alpha_m (1 - m) - \beta_m m, \quad \alpha_m = \frac{0.1(25 - V)}{e^{(25 - V)/10} - 1}, \quad \beta_m = 4e^{(-V/18)}$ $\dot{h} = \alpha_h (1 - h) - \beta_h h, \quad \alpha_h = 0.07e^{(-V/20)}, \quad \beta_h = \frac{1}{e^{(30 - V)/10} + 1}$	$E_K = -12$ mV $E_{Na} = 120$ mV $E_L = 10.6$ mV $g_K = 36$ mS/cm <sup>2</sup> $g_{Na} = 120$ mS/cm <sup>2</sup> $g_L = 0.3$ mS/cm <sup>2</sup> $C = 1$ μF/cm <sup>2</sup>	(1)						
FHN neuron model	$\dot{v} = c(v - u + I - v^3/3)$ $\dot{u} = (v - bu + a)/c$	$a = 0.7$ $b = 0.8$ $c = 3$	(2)						
Izhikevich neuron model	$\dot{v} = 0.04v^2 + 5v + 140 - u + I$ $\dot{u} = a(bv - u)$ $v \geq 30 \text{ mV} \Rightarrow \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$	<table border="1"> <tr> <td>Spike</td> <td><math>a = 0.02, b = 0.2, c = -65, d = 8, I = 14</math></td> </tr> <tr> <td>Burst</td> <td><math>a = 0.02, b = 0.2, c = -50, d = 2, I = 15</math></td> </tr> <tr> <td>Fast spike</td> <td><math>a = 0.1, b = 0.2, c = -65, d = 0, I = 14</math></td> </tr> </table>	Spike	$a = 0.02, b = 0.2, c = -65, d = 8, I = 14$	Burst	$a = 0.02, b = 0.2, c = -50, d = 2, I = 15$	Fast spike	$a = 0.1, b = 0.2, c = -65, d = 0, I = 14$	(3)
Spike	$a = 0.02, b = 0.2, c = -65, d = 8, I = 14$								
Burst	$a = 0.02, b = 0.2, c = -50, d = 2, I = 15$								
Fast spike	$a = 0.1, b = 0.2, c = -65, d = 0, I = 14$								
HR neuron model	$\dot{v} = u - v^3 - bv + I - w$ $\dot{u} = 1 - 5v^2 - u$ $\dot{w} = \mu(s(v - v_{rest}) - w)$	<table border="1"> <tr> <td>Spike</td> <td><math>b = 2.96, I = 5</math></td> </tr> <tr> <td>Burst</td> <td><math>b = 2.6, I = 2.66</math></td> </tr> <tr> <td>Chaotic</td> <td><math>b = 2.96, I = 3</math></td> </tr> </table>	Spike	$b = 2.96, I = 5$	Burst	$b = 2.6, I = 2.66$	Chaotic	$b = 2.96, I = 3$	(4)
Spike	$b = 2.96, I = 5$								
Burst	$b = 2.6, I = 2.66$								
Chaotic	$b = 2.96, I = 3$								
ML neuron model	$C \dot{v} = -\bar{g}_{Ca} m_\infty(V)(V - E_{Ca}) + \bar{g}_K w(V - E_K) + \bar{g}_L (V - E_L) + I$ $\dot{w} = \phi \frac{[w_\infty(V) - w]}{\tau_w(V)}$ $m_\infty(V) = 0.5[1 + \tanh\{(V - V_1)/V_2\}]$ $w_\infty(V) = 0.5[1 + \tanh\{(V - V_3)/V_4\}]$ $\tau_w(V) = 1 / \cosh\{(V - V_3)/(2V_4)\}$	$V_1 = -1.2$ mV $V_2 = 18$ mV $V_3 = 2$ mV $V_4 = 17.4$ mV $g_{Ca} = 4$ mS/cm <sup>2</sup> $g_K = 8$ mS/cm <sup>2</sup> $g_L = 2$ mS/cm <sup>2</sup> $V_K = -84$ mV $V_L = -60$ mV $\phi = 1/15$	(5)						



**Figure 1.** Numerical simulation results of HH neuron model for: a)  $I = 10 \mu\text{A}/\text{cm}^2$ , b)  $I = 100 \mu\text{A}/\text{cm}^2$ .

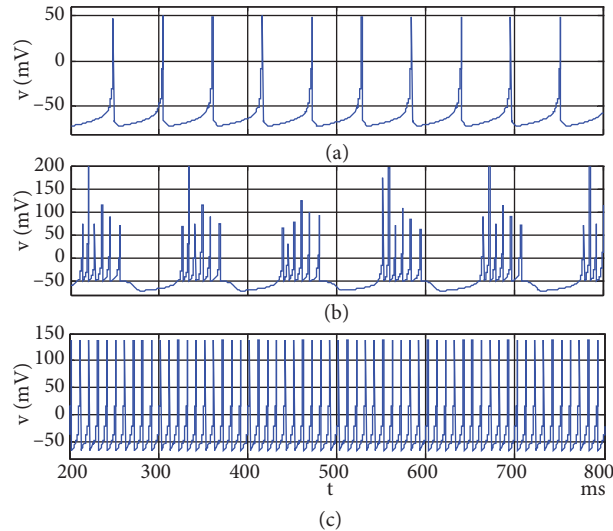
To reduce the variable parameters of the HH neuron model, a number of studies were put forward by FitzHugh [25]. In subsequent works, the FHN neuron model, defined by Eq. (2) in Table 1, was introduced by assimilating to the Van der Pol oscillator [26]. In Eq. (2),  $v$  is the membrane potential,  $u$  is the recovery parameter, and  $I$  represents the membrane input current.  $a$ ,  $b$ , and  $c$  are variable parameters. The simulation results of this system are presented in Figure 2. The numerical simulation results show that the FHN neuron is quiescent up to  $I = 0.32$  and then it exhibits only spiking behavior. Although the frequency of the spike signal depends on membrane input current in the real neurons, it does not depend on membrane input current in this model. The FHN neuron model is not able to exhibit all neuron dynamics, so it is less adequate than the other neuron models.



**Figure 2.** Numerical simulation results of FHN neuron model: a) for  $I = 0.34$  and  $I = 0.32$  membrane potential, b)  $v$ - $u$  phase portrait illustration.

The Izhikevich neuron model was proposed in 2003 [8]. This model has calculation easiness, as in the integrate-and-fire type neuron models, and it includes substantial dynamics, as in the HH neuron model. The Izhikevich neuron model is given in Eq. (3) of Table 1 [8]. In Eq. (3)  $v$  is the membrane potential,  $u$  is the recovery parameter, and  $I$  represents the membrane input current.  $a$  describes the time scale recovery variable  $u$ , and  $b$  is the sensitivity of  $u$ .  $c$  and  $d$  are the after-spike reset parameters. The spiking behavior, bursting behavior, and fast spiking behavior of the Izhikevich neuron model are given in Figures 3a, 3b, and 3c, respectively. In the original study that introduced the Izhikevich neuron model [8], the various neuron behaviors are produced by depending on the changes of four parameters ( $a, b, c, d$ ) in Eq. (3). If these parameters in Eq.

(3) are adjusted to appropriate values similarly, most of the neuron dynamics can be observed via the Izhikevich neuron model.

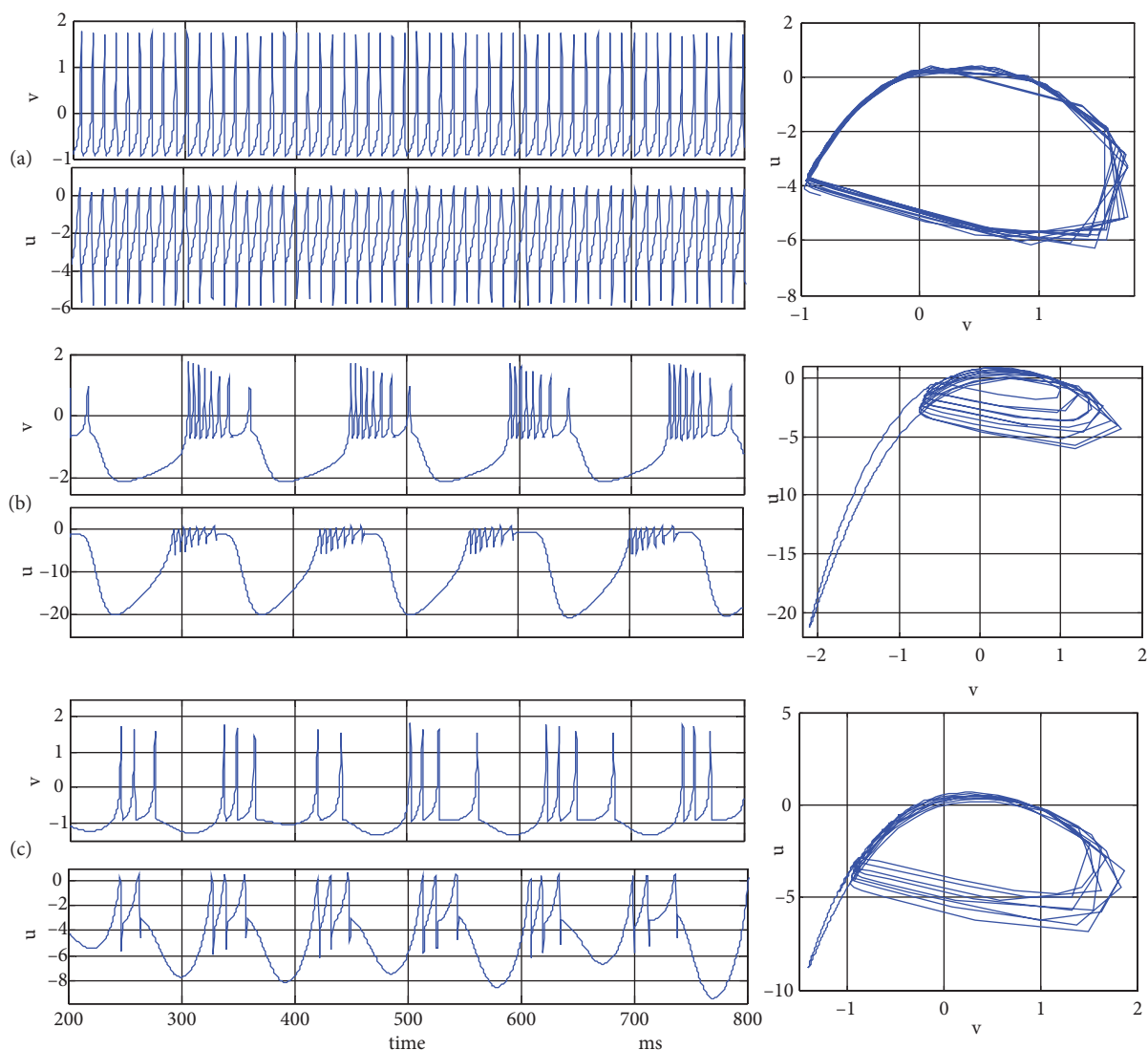


**Figure 3.** Numerical simulation results of Izhikevich neuron model: a) spiking behavior, b) bursting behavior, c) fast spiking behavior.

The FHN neuron model [3] is the simplified version of the HH neuron model. This model is defined by two differential equations, but it does not have the capability of exhibiting certain behaviors of the neurons. The HR neuron model was derived from the FHN neuron model by adding one differential expression to its equations [7]. Despite its simplicity, the HR neuron model is able to exhibit several of the dynamic behaviors of a real neuron and it is defined by Eq. (4) in Table 1. In this equation, the  $I$  parameter represents the membrane input current,  $b$  controls the transition between bursting and spiking,  $\mu$  controls the spiking frequency and the number of spikes per burst in the case of spiking and bursting respectively,  $s$  adjusts adaptation, and  $v_{rest}$  is the resting potential. By fixing the parameters  $\mu = 0.01$ ,  $s = 4$ , and  $v_{rest} = -1.6$ , the HR neuron model dynamics are investigated with a numerical simulation tool, depending on parameters  $I$  and  $b$  as in [27]. Numerical simulation results for spike, burst, and chaotic behaviors of this model are shown in Figures 4a, 4b, and 4c, respectively.

The ML neuron model is expressed as a mathematical description of the giant barnacle muscle fiber [6]. It is known as a conductance-based model and uses calcium, potassium, and ohmic leak current. The ML neuron model is defined by Eq. (5) in Table 1. In Eq. (5),  $v$  is the membrane potential;  $w$  represents the slow activation of potassium channels;  $I$  is the membrane input current;  $g_{Ca}$ ,  $g_K$ , and  $g_L$  are conductances of leak current,  $E_K$ ,  $E_{Ca}$ , and  $E_L$  are equilibrium potentials; and  $V_1, V_2, V_3$ , and  $V_4$  represent variable parameters. The numerical analysis results of this model are given in Figure 5. For  $I = 0$ , the ML neuron is in the quiescent state. As the value of the membrane input current increases, the ML neuron exhibits spiking behaviors, but the frequency of the spike signal does not change in the numerical simulation and the ML neuron passes again to the quiescent state in the later values of  $I$ . The ML neuron model does not exhibit other neuron dynamics such as in the FHN neuron model.

As seen from the numerical analyses, the simplest neuron model derived from the HH neuron model is the FHN neuron model. Additionally, the Izhikevich neuron model has both calculation easiness and the ability to exhibit many different neuron behaviors [8]. The HR neuron model is used in synchronization application studies

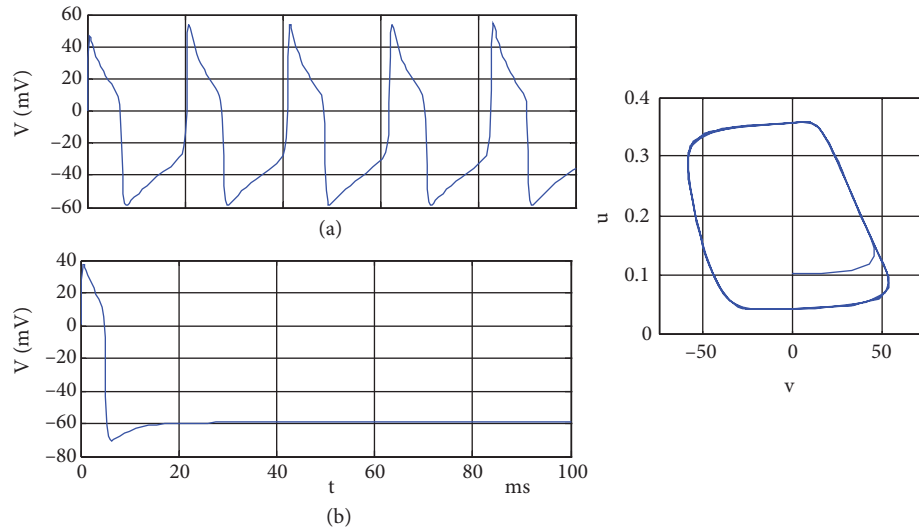


**Figure 4.** HR neuron dynamics and x-y plane demonstration for: a) spiking mode, b) bursting mode, and c) chaotic mode.

[27–30]. The FHN, Izhikevich, and HR neuron models are often utilized in hardware implementation studies of biological neuron models. Thus, these neuron models are preferred and implemented using both analog and digital programmable devices in the following section to evaluate their neuron model realization performances in terms of the design flexibility, power consumption, area consumption, dynamic range, and frequency range.

### 3. Programmable analog and digital implementations of neuron models

In this section, the FHN, Izhikevich, and HR neuron models are implemented with both FPAA and FPGA in order to give the whole design of frequently studied models for the first time. Here, the design processes of FPAA- and FPGA-based neuron model implementations are presented in detail.



**Figure 5.** Numerical simulation results of ML neuron model for: a)  $I = 60 \mu\text{A}/\text{cm}^2$ , b)  $I = 0 \mu\text{A}/\text{cm}^2$ .

### 3.1. FPAA-based implementations

FPAAs provide high stability, accuracy, and rapid prototyping techniques and use configurable analog blocks (CABs), which consist of op-amps, an array of switches, and a capacitor bank. In this study, AN231E04 type FPAA boards were used. Comprehensive descriptions of AN231E04 and configurable analog modules (CAMs) are available from this software tool. FPAA chips have some limitations, such as capacity and a specific saturation level ( $\pm 2$  V). FPAA boards can be combined with each other in order to cope with the capacity problem. To overcome the saturation problem, the system model must be rescaled according to numerical simulation results and saturation levels in the design process. After the rescaling process, the FHN, Izhikevich, and HR neuron models are modified and redefined by Eqs. (6), (7), and (8) in Table 2, respectively. Additionally, the values of the parameters in the models change depending on the rescaling and modification operations and their new values are listed in Table 2.

To implement analog functions such as multiplication, addition, filtering, and rectification, various configurable analog blocks (CAMs) are predefined in the FPAA's software tool with detailed documentation. The CAMs used in the implementations of the FHN, Izhikevich, and HR neuron models are shown in Table 2 and listed as follows: *SUMFILTER*, *TRANSFER FUNCTION*, *MULTIPLIER*, *VOLTAGE*, *SUMDIFF*, and *SUMINV*. The *SUMFILTER* block has a single-pole low-pass filter and three weighted analog inputs, and the state variables  $v$ ,  $u$ , and  $w$  of neuron models are implemented with this block. The *TRANSFER FUNCTION* block generates a user-specified voltage transfer function with 256 quantization steps. The nonlinear functions, which are harder to implement, are embedded on the *TRANSFER FUNCTION* block in the equations. The *MULTIPLIER* block is used for performing the multiplication operation and in this study it is used to obtain quadratic expressions easily. The *VOLTAGE* block is a DC voltage source and its value ranges from  $\pm 2$  V, and the constant value expressions as membrane input current in neuron model equations have been contracted with this block. The remaining addition and subtraction operations have been implemented with the *SUMDIFF* or *SUMINV* blocks. These blocks are similar to the *SUMFILTER* block, but they do not consist of a filter.

After the selection of these CAM blocks and construction of the required connections between these blocks are done by using the FPAA interface tool, the neuron models are transferred to the FPAA board.

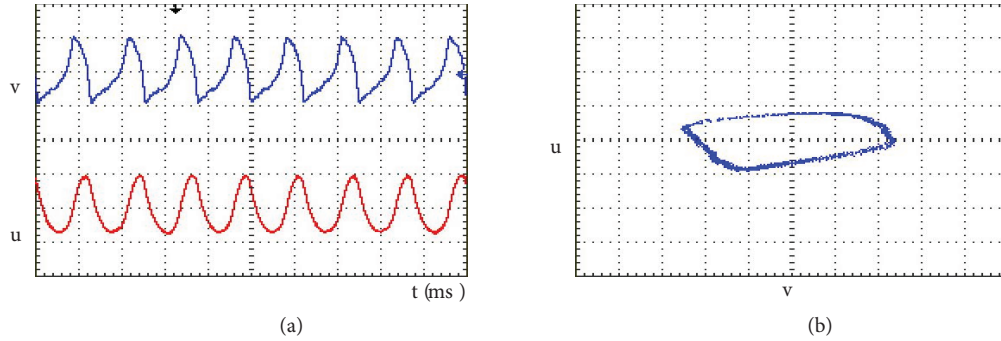
**Table 2.** Mathematical descriptions of biological neuron models after the rescaling operation and the FPAA-based implementation schemes.

Neuron model	Rescaled equations	The FPAA-based implementation scheme of the models
FHN neuron model	$\dot{v}=4v-2u+0.523-4v^3 \quad (6)$ $\dot{u}=0.66v-0.733u+0.233$	
Izhikevich neuron model	$\dot{v}=v+(1.2v^2+5v+4.67+I)-0.13u \quad (7)$ $\dot{u}=u+a(bv-u)$ $v \geq 1mV \Rightarrow \begin{cases} v \leftarrow c \\ u \leftarrow u+d \end{cases}$ <p>where:  <math>a = 0.02, b = 1.5, c = -2.167,</math>  <math>d = 2, I = 0.33.</math></p>	
HR neuron model	$\dot{v}=v-v^3+bv^2+I-aw+cu \quad (8)$ $\dot{u}=e-fv^2$ $\dot{w}=0.9976w+(0.016+0.02v)$ <p>where: <math>a = 2, b = 2.96, c = 4, f = 3.125,</math>  <math>e = 0.125, I = 0.5</math> for spiking behavior and  <math>a = 2, b = 2.6, c = 10, f = 0.055, e = 1, I = 0.8</math>  for bursting behavior.</p>	

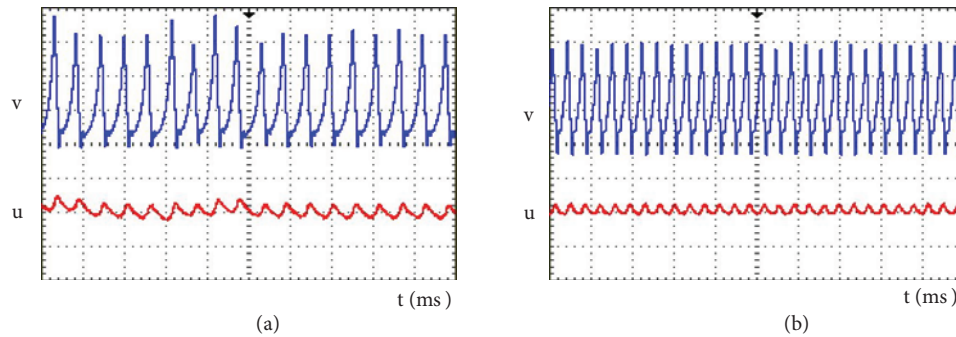
The experimental results are then recorded via an oscilloscope. The time-domain representations of the  $v$  and  $u$  variables for the FHN neuron model are shown in Figure 6a, and the phase portrait illustration of  $v$  and  $u$  is given in Figure 6b. The experimental results of the FPAA-based Izhikevich neuron model are presented in Figure 7. The frequency of the spike signal increases depending on the increment of the membrane input current in the real neurons and the Izhikevich neuron model satisfies this property. The experimental results of FPAA-based implementation are also compatible with this feature. While the membrane input current is 0.33 in Figure 7a, it is increased to 1.17 in Figure 7b. As shown in these figures, the frequency of the spike signal increases depending on this increment. In Figure 8a, the HR neuron exhibits spiking behaviors, and the



$v - u$  phase portrait illustration of the HR neuron model for spiking behavior is shown in Figure 8b. On the other hand, the bursting behavior of the HR neuron model is shown in Figure 8c and the  $v - u$  phase portrait illustration of this behavior is given in Figure 8d.

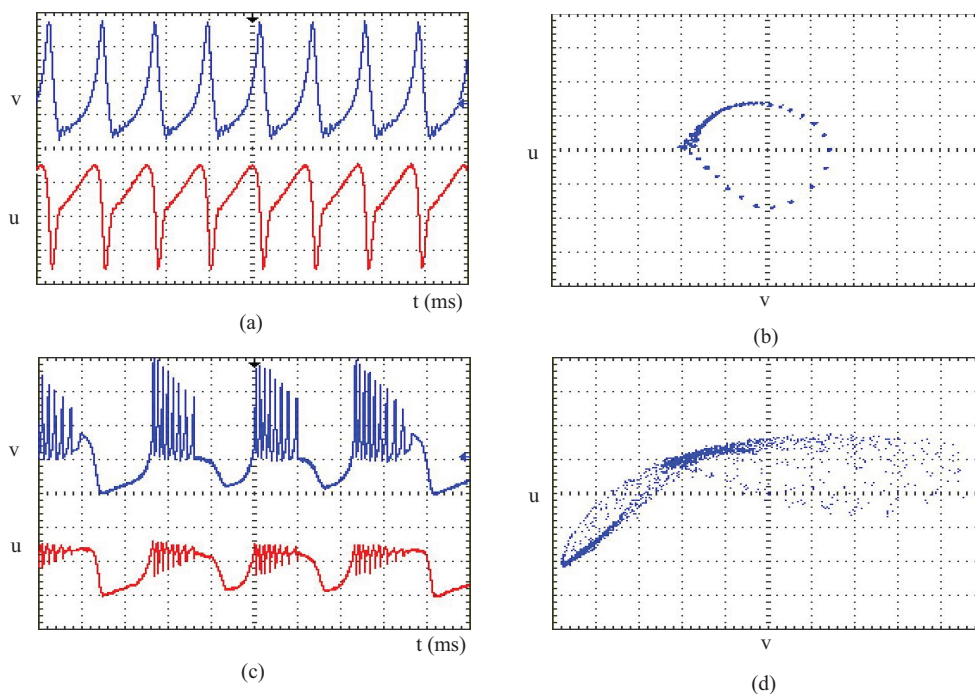


**Figure 6.** The experimental results of FPAAs-based FHN neuron model: a) spiking behavior, x: 0.5 V/div, y: 0.5 V/div, time/div: 2.5 ms; b) v-u phase portrait illustration, x: 0.2 V/div, y: 0.5 V/div.



**Figure 7.** The experimental results of FPAAs-based Izhikevich neuron model for: a)  $I = 0.33$ , b)  $I = 1.17$ , x: 0.5 V/div, y: 0.5 V/div, time/div: 0.5 ms.

The obtained experimental results for these neuron models agree with the numerical simulation results. There is an available study on FPAAs-based implementation of the FHN neuron model in the literature [31]. While two FPAAs boards were used in the previous study, one FPAAs board is used in our study and the experimental results are compatible with each other. In the FPAAs-based Izhikevich neuron model implementation, only the spiking behavior of this model was achieved with FPAAs devices; other behaviors could not be obtained. The most important factor for this situation is thought to be that the detailed mathematical modeling and theoretical analysis studies of the Izhikevich neuron model depend on its discrete-time system description. For example, the analysis of the Izhikevich neuron model in the original study [8] was handled by utilizing a discretization method, which is more convenient for the digital systems. Since the nonlinear function definitions and the parameter adjustments of the HR neuron model are realized via an interface software tool in FPAAs implementations, realizations of the model become easier and more sensitive according to other hardware options. The CAB capacity usages and power consumptions of the FPAAs-based neuron model implementations are presented in Table 3.



**Figure 8.** The experimental results of FPAA-based HR neuron model: a) spiking mode, x: 0.5 mV/div, y: 0.5 mV/div, time/div: 1 ms; b) v-u phase portrait illustration of the spiking mode, x: 0.5 mV/div, y: 0.5 mV/div; c) bursting mode, x: 0.5 mV/div, y: 0.5 mV/div, time/div: 10 ms; d) v-u phase portrait illustration of the bursting mode, x: 0.2 mV/div, y: 0.2 mV/div.

**Table 3.** Capacity usages and power consumptions of FPAA-based neuron model implementations.

		Power consumptions	CAB1	CAB2	CAB3	CAB4
FHN neuron model		$32 \pm 28$ mW	5/8	5/8	6/8	6/8
Izhikevich neuron model		$57 \pm 17$ mW	4/8	5/8	0/8	0/8
HR neuron model	FPAA 1	$128 \pm 38$ mW	8/8	7/8	5/8	5/8
	FPAA 2	$56 \pm 17$ mW	7/8	0/8	0/8	0/8

### 3.2. FPGA-based implementations

FPGAs consist of programmable digital blocks and programmable interconnections between them. Unlike standard digital microprocessors, FPGAs provide massive parallelism in digital implementations. FPGA boards include programmable I/O units, programmable interconnections, PLL and clock channels, memory units and multipliers, and programmable logic blocks. The Cyclone III IC available on the Altera DE0 educational board is used in this study. A Cyclone III IC has 15,408 digital blocks called logic elements and 112 nine-bit embedded multipliers, and an Altera DE0 board has various peripherals like switches and LEDs. To realize the neuron models with FPGA, very high-speed integrated circuit hardware description language (VHDL) is used. The codes of VHDL are created in the software tool by using 32-bit fixed-point numerical notation.

The FPGA is an example of a programmable and reconfigurable digital device. For this reason, continuous systems must transform discrete structures with a discretization method. Neuron models are usually described by continuous ordinary differential equations. To implement them with an FPGA device, they must also be

discretized. After the Euler discretization method is applied to the FHN, Izhikevich, and HR neuron models, the models can be defined by Eq. (9), (10), and (11), respectively, in Table 4. As the value of discretization constant  $\Delta h$  varies, the system dynamics also change. For example, a burst behavior in the original model may turn into a spike behavior in the discretized model under the same parameter values. Therefore, the  $\Delta h$  value is determined depending on the numerical simulations, which represent original behaviors of the biological neuron models [8]. According to the numerical simulation results, the discretization constant  $\Delta h$  is chosen as 0.01 in all equations in this study and the closest results are obtained for this value. On the other hand,  $\Delta h$  is just a numerical value represented with different bit resolutions in FPGA implementations. As long as the same bit resolutions are used, changes in the  $\Delta h$  value do not affect the power consumption and resource allocation.

**Table 4.** Mathematical descriptions of biological neuron models after the discretization operation for the FPGA-based implementations.

Neuron model	Equations	Parameters		Eq. no.
FHN neuron model	$v_{n+1} = v_n + \Delta h [c(v_n - u_n + I - v_n^3 / 3)]$ $u_{n+1} = u_n + \Delta h [(v_n - bu_n + a) / c]$	$a = 0.7$ $b = 0.8$ $c = 3$		(9)
Izhikevich neuron model	$v_{n+1} = v_n + \Delta h [0.04v_n^2 + 5v_n + 140 - u_n + I]$ $u_{n+1} = u_n + \Delta h [a(bv_n - u_n)]$ $v_{n+1} \geq 1mV \Rightarrow \begin{matrix} v_{n+1} \leftarrow c \\ u_{n+1} \leftarrow u_n + d \end{matrix}$	Spike	$a = 0.02, b = 0.2, c = -65, d = 8, I = 14$	(10)
		Burst	$a = 0.02, b = 0.2, c = -50, d = 2, I = 15$	
		Fast spike	$a = 0.1, b = 0.2, c = -65, d = 0, I = 14$	
HR neuron model	$v_{n+1} = v_n + \Delta h (u_n - v_n^3 + bv_n^2 + I - w_n)$ $u_{n+1} = u_n + \Delta h (1 - 5v_n^2 - u_n)$ $z_{n+1} = z_n + \Delta h (\mu (s(v_n - v_{rest}) - w_n))$	Spike	$b = 2.96$ $I = 5$	(11)
		Burst	$b = 2.6$ $I = 2.66$	
		Chaotic	$b = 2.96$ $I = 3$	

To obtain the  $v^3$  function of the FHN neuron model in Eq. (9), multiplier blocks are used as in Figure 9. The rest of the equations can be constructed with programmable logic blocks. The signal, which is taken from pins on a DE0 board in a parallel way, is applied to a DAC0808 digital/analog convertor. The experimental results are measured from the output of the DAC and they are shown in Figure 10. The obtained experimental results agree with the numerical simulation results and FPAA-based implementation results.

To implement the FPGA-based Izhikevich neuron model, in addition to the advantage of the programmability feature of the FPGA device, a switching mechanism is offered for easy observation of results. Before proceeding to this switching mechanism, the continuous time equations of the model are transformed to discrete-time equations, as shown in Eq. (10). To observe the various dynamics of the Izhikevich neuron model without making any parameter adjustments or reprogramming the device, a switching mechanism is suggested. In this mechanism, a general circuit structure is constructed; the signals are assigned corresponding to the parameters  $a, b, c, d$ , and  $I$ ; and these signals are applied to the general circuit. The values of the variable parameters are the same as those in the numerical simulations. In this way, the FPGA device is programmed once and the limited space available on it is saved. The representation of this switching mechanism is shown in Figure 9.

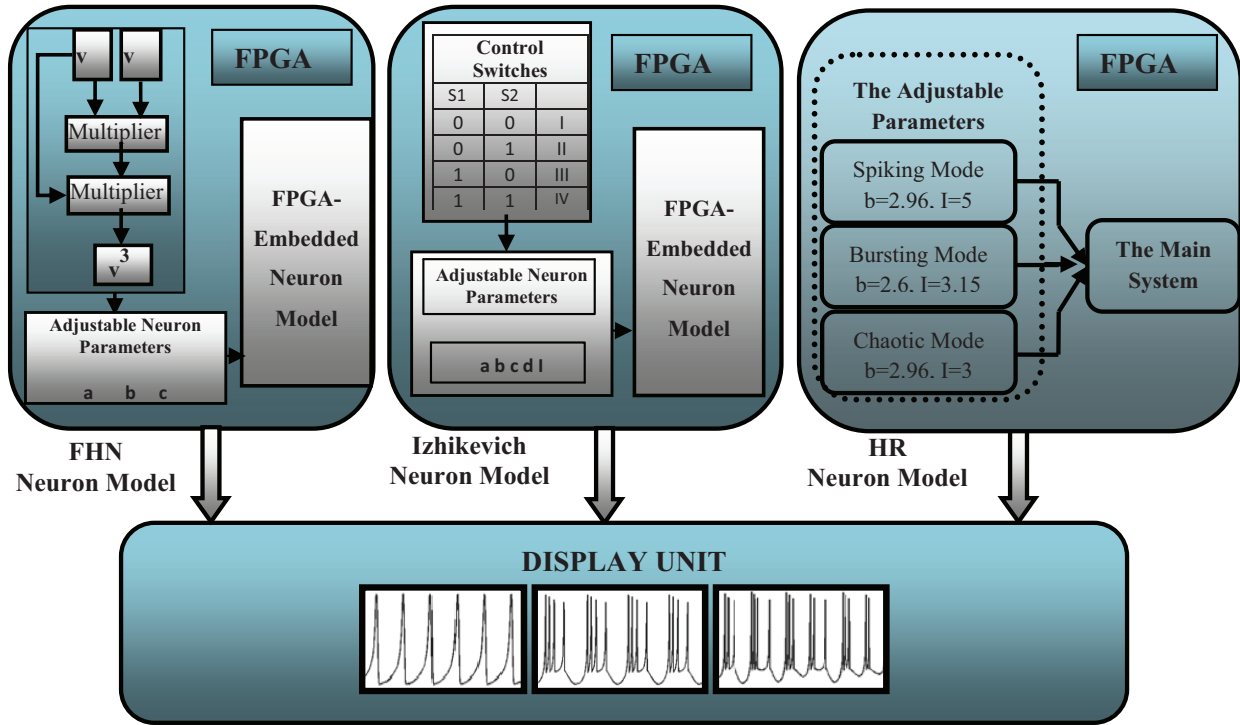


Figure 9. A block diagram of the implementation of the FPGA-based neuron models.

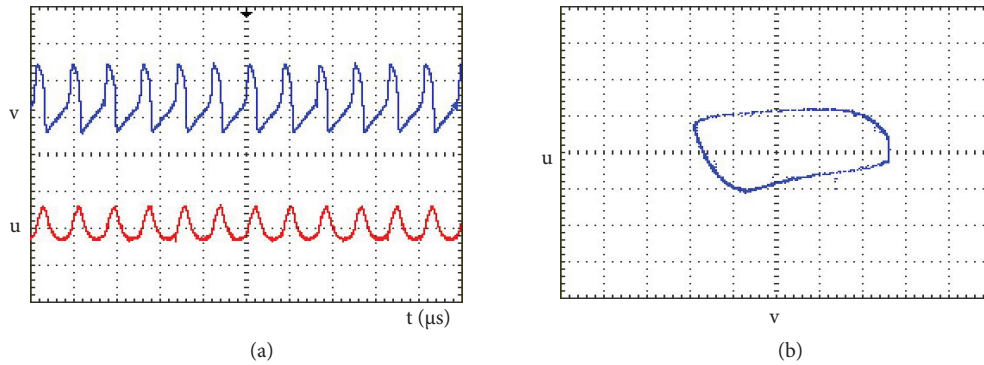
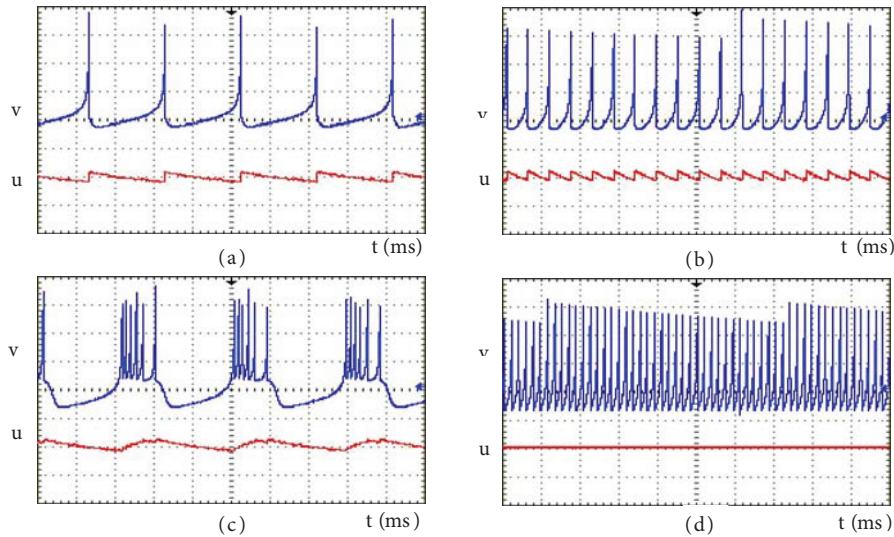


Figure 10. The experimental results of FPGA-based FHN neuron model: a) spiking behavior, x: 5 V/div, y: 5 V/div, time/div: 100  $\mu$ s; b) v-u phase portrait illustration, x: 2 V/div, y: 5 V/div.

When the control switches are in the  $[S1, S2] = [0, 0]$  position, regular spike behavior is observed in the display unit, as shown in Figure 11a. If the membrane input current  $I$  is increased, the frequency of the spike signal increases in real biological neurons. In the  $[S1, S2] = [0, 1]$  position, only the  $I$  parameter is changed and frequency changing can be seen in Figure 11b. While the bursting behavior in Figure 11c of the Izhikevich neuron model is obtained for the  $[S1, S2] = [1, 0]$  position, the fast spiking behavior in Figure 11d is recorded when the control switches are in the  $[S1, S2] = [1, 1]$  position. Thanks to this property, different neuron dynamics were observed without making any parameter adjustments or reprogramming the device.

The difference equations of the HR neuron model are given in Eq. (11) of Table 4 [32]. In the design process, the main system in Eq. (11) is formed using VHDL in software using 32-bit fixed-point numerical notation. Here, the control parameters are  $b$  and  $I$ , because only they affect the fire patterns. These two



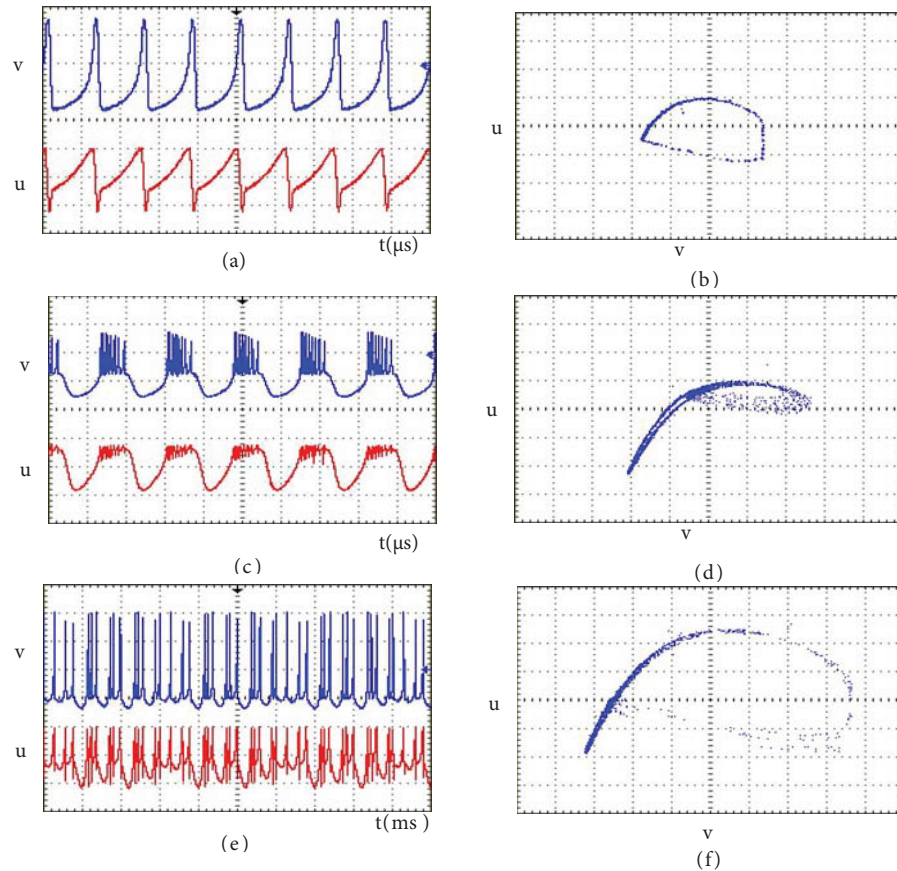
**Figure 11.** The experimental results of FPGA-based Izhikevich neuron model: a) regular spiking behavior ( $I = 14$ ), b) regular spiking behavior ( $I = 50$ ), c) bursting behavior, d) fast spiking behavior [x: 1 V/div, y: 1 V/div, time/div: 1 ms].

parameters are adjusted via the software tool to obtain the different patterns of the neuron with the FPGA and the built system is downloaded to the FPGA device. Experimental measurements are monitored through a DAC block. The representation of the experimental setup of the FPGA-based HR neuron is shown in Figure 9. In Figure 12, three different behaviors of the HR neuron model on the FPGA are given. While regular spiking behavior of the HR neuron model is recorded in the time domain in Figure 12a, the  $v - u$  phase portrait illustration for spiking behavior is presented in Figure 12b. Similar to these illustrations, Figures 12c and 12d show bursting behavior and its phase portrait. Figure 12e represents chaotic behavior of the HR neuron model and Figure 12f the phase portrait of the chaotic dynamic. The obtained experimental results for the FPAAs and FPGA-based HR neuron models agree with the numerical simulation results. The synthesis results and estimated power consumption values of the FPGA-based neuron model implementations are given in Table 5.

#### 4. Discussion

After presenting an overview about neuron models and their programmable device-based implementations, the question of which design procedure offers powerful usage to implement the neuron models should be answered. Thus, a qualitative comparison for the implementation methods of biological neuron models is given within several categories, such as achieving the dynamic behaviors of the neurons with these devices, design flexibility, power consumption, area consumption, dynamic range, frequency range, and suitability for network implementation. These mentioned categories are summarized in Table 6. Certain numerical values for some criteria in Table 6, such as power and area consumptions, can be obtained from related references, numerical analyses, and results in Tables 3 and 5. However, some aspects like the suitability for network implementation or dynamic range do not depend on the numerical results and they are a general evaluation about performances of the implementation methods. Thus, in Table 6, performance comparison of devices is summarized instead of giving exact numerical results. According to Table 6, while (+++) shows high performance and (++) represents good performance, (+) means average acceptability.

The first hardware implementation of a neuron model was implemented by using transistor-based circuitry



**Figure 12.** The experimental results of FPGA-based HR neuron model: a) spiking mode, x: 2 V/div, y: 2 V/div, time/div: 50  $\mu$ s; b) x-y phase portrait illustration of the spiking mode, x: 2 V/div, y: 2 V/div, time/div: 50  $\mu$ s; c) bursting mode, x: 2 V/div, y: 2 V/div, time/div: 500  $\mu$ s; d) x-y phase portrait illustration of the bursting mode, x: 1 V/div, y: 1 V/div; e) chaotic mode, x: 2 V/div, y: 2 V/div, time/div: 1 ms; f) x-y phase portrait illustration of the chaotic mode, x: 1 V/div, y: 1 V/div.

**Table 5.** The synthesis results and estimated power consumption values of FPGA-based neuron model implementations.

	Area usage (logic elements)	Embedded multiplier usage	Maximum frequency	Power consumption
FHN neuron model	1042 (7%)	64 (57%)	18.11 MHz	83.02 mW
Izhikevich neuron model	1325 (8.6%)	20 (18%)	14.19 MHz	83.29 mW
HR neuron model	1156 (8%)	64 (57%)	23.1 MHz	83.02 W

and it depended on the HH neuron model [22] and examples performed by discrete devices for the other neuron models in the literature [22,23]. Additionally, several ASIC implementations for almost all neuron models are available in the literature [13–24]. For example, the HH neuron model was realized by using 2- $\mu$ m BICMOS technology in [13]. CMOS-based implementations of the FHN neuron model focused on that these types of implementations present a compact structure [15,16]. There are also some examples of VLSI-based implementations of FHN, Izhikevich, and HR neuron models [17–20]. On the other hand, although there are few individual design examples of programmable device-based biological neuron model implementations in the literature [31,33–35], both analog and digital programmable implementation details of the frequently studied

**Table 6.** A qualitative comparison table for the implementation methods of biological neuron models.

Biological neuron models	FPAA			FPGA			ASIC			Discrete implementation		
	FHN	IZ	HR	FHN	IZ	HR	FHN	IZ	HR	FHN	IZ	HR
To achieve the dynamic behaviors of the neurons	+++	+	+++	+++	+++	+++	+++	+++	+++	+++	+	+++
Design flexibility	+++			+++			+			+		
Power consumptions	++	++	+	++	++	++	+++	+++	+++	++	++	++
Area consumptions	++	+++	+	+++	++	++	+++	+++	+++	(1)		
Frequency range	++			+++			+++			++		
Dynamic range	++			+++			++			+++		
Suitability for network implementation	++			+++			+++			+		

(1): This parameter has been only compared for FPAA, FPGA, and ASIC implementations.

biological neuron models have been presented respectively in this study for the first time. The results obtained in these studies showed that several different ASIC implementations, discrete circuit methods, and programmable devices can be used to achieve the dynamics of neuron behaviors successfully.

As seen from the neuron models' descriptions in Table 1, the models consist of many parameters and nonlinear expressions. A high-precision parameter setup must be used to realize these adjustable functions. This setup process is realized by using software control of FPAA and FPGA devices, so they offer very flexible design procedures yielding reduction in the complexity of the design in contrast to the discrete devices or ASIC applications.

Power consumption is accepted as another comparison parameter for the implementations of neuron models. While FPAA power consumption ranges between 32 and 166 mW, FPGA power consumption is calculated as approximately 83 mW, as presented in Tables 4 and 5. In the discrete circuit realizations, power consumption is strictly related to the used components and the complexity of the model. For example, the power consumption of the FHN neuron model, which is one of the simplest models, was approximately calculated as 6 mW via PSPICE simulation for discrete circuit-based implementation in [23]. The power consumption also increases as the used number of components increases for more complicated models. On the other hand, ASIC implementations have very good results for power consumption. In the VLSI implementations of the FHN and HR neuron models [17,20], power consumptions are calculated as 23  $\mu$ W and 163.4  $\mu$ W, respectively.

According to Tables 4 and 5, the HR neuron model has the highest area usage in FPAA-based implementation, with the FHN neuron model ranking second and the Izhikevich neuron model ranking last. While the area usages of the HR and Izhikevich neuron models are nearly the same, the FHN neuron model can be designed using fewer logic components in FPGA-based implementations. As mentioned previously, ASIC implementations offer compact structure and neuron models, and a simple network consisting of a few neurons can be constructed in a very small area on the chip [13,16]. FPAA and FPGA devices were also used the implementations of neuronal structures in the literature [32,34]. Although the performance of FPAA devices is not as effective as FPGA-based implementations for network structures, the nonlinear functions in models can be realized with FPAA more easily than FPGA. Discrete device-based implementations are harder for realizing the neural structures.

The other classification parameter is the dynamic range. The dynamic range of the FPAA IC is restricted with the rescaling process because of the saturation level of the devices. The dynamic range of the FPGA IC

depends on the bit length of the numerical notation in use instead of supply voltages. When the digital outputs of FPGA are converted to analog, the dynamic range can be adjusted via the amplifier stage of the DAC unit. For these reasons, dynamic ranges of FPGA-based implementations are evaluated as having high acceptability in Table 6. Since discrete circuit implementations have the highest supply voltages, the widest dynamic range belongs to them.

The datasheet of the FPGA device shows that the FPGA has a wide frequency range of 500 MHz. In contrast to this, the default operating frequency in FPAA's is chosen as 2 MHz and can be extended to 8 MHz. Therefore, the frequency range of the FPGA is better than the FPAA's. In ASIC implementations, frequency range varies according to the used methods. For example, a transconductance amplifier with 6 GHz operation frequency was designed in [14] for the RF filter of mobile communication and the biologically plausible neuromorphic hardware and the HH neuron model was implemented with this transconductance amplifier. If current-mode active elements are used in the discrete device-based circuit configuration of neuron models, the dynamic behaviors in MHz can be obtained.

## 5. Conclusions

A comprehensive study about the implementations of biological neuron models was presented in order to give programmable and reconfigurable implementations of the FHN, Izhikevich, and HR neuron models by comparing their design aspects. In order to compare the suitability of design methods for the implementations of biological neuron models, advantages and disadvantages of each implementation method have been discussed in terms of achieving the dynamic behaviors of the neurons with these devices, design flexibility, power consumption, area consumption, dynamic range, frequency range, and suitability for network implementation. Since FPAA and FPGA devices have programmability and reconfigurability features, they have the best performance in terms of design flexibility and suitability for network implementation. For the designs that require low power consumption and high frequency performance, ASIC implementations are the best choice. If the model is not very complex, discrete devices are also a remarkable alternative.

As a result of these comparisons, it can be said that both FPAA and FPGA devices have useful properties and they can be used in neuronal studies at least as effectively as the other design methods. In particular, experimental prototypes of the other incipient neuronal structures are able to be set up, tested, and verified rapidly by taking advantage of the programmability and reconfigurability features of these devices. In real-time applications requiring massive neural hardware, FPAA- and FPGA-based implementations may provide alternative powerful design tools.

## References

- [1] McCulloch WS, Pitts WH. A logical calculus of ideas imminent in nervous activity. *B Math Biol* 1943; 5: 115-133.
- [2] Hodgkin A, Huxley A. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol* 1952; 117: 500-544.
- [3] FitzHugh R. Mathematical models for excitation and propagation in nerve. In: Schawm HP, editor. *Biological Engineering*. New York, NY, USA: McGraw-Hill, 1969. pp. 1-85.
- [4] Nagumo J, Sato S. On a response characteristic of mathematical neuron model. *Kybernetik* 1972; 10: 155-164.
- [5] Wilson HR, Cowan JD. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys J* 1972; 12: 1-24.
- [6] Morris C, Lecar H. Voltage oscillations in the barnacle giant muscle fiber. *Biophys J* 1981; 35: 193-213.



- [7] Hindmarsh JL, Rose RM. A model of neuronal bursting using three couple first order differential equations. *P R Soc Lond B Bio* 1984; 22: 87-102.
- [8] Izhikevich EM. Simple model of spiking neurons. *IEEE T Neural Networ* 2004; 14: 1569-1572.
- [9] Mishra D, Yadav A, Ray S, Kalra PK. Exploring biological neuron models. *Research Magazine of IIT Kanpur* 2006; 7: 13-22.
- [10] Li B, Li Y, Rong X. Gait generation and transitions of quadruped robot based on Wilson-Cowan weakly neural networks. In: *IEEE 2010 International Conference on Robotics and Biomimetics*; 14–18 December 2010; Tianjin, China. New York, NY, USA: IEEE. pp. 19-24.
- [11] Lu J, Kim YB, Ayers J. A low power 65nm CMOS electronic neuron and synapse design for a biomimetic micro-robot. In: *IEEE 2011 International Midwest Symposium on Circuits and Systems*; 7–10 August 2011; Seoul, Korea. New York, NY, USA: IEEE. pp. 1-4.
- [12] Lee YJ, Lee J, Kim YB, Ayers J. Low power CMOS adaptive electronic central pattern generator design. In: *2005 Midwest Symposium on Circuits and Systems*; 7–10 August 2005; Covington, KY, USA. New York, NY, USA: IEEE. pp. 1350-1353.
- [13] Dupeyron D, Le Masson S, Deval Y, Le Masson G, Dom JP. A BiCMOS implementation of the Hodgkin-Huxley formalism. In: *Proceedings of 1996 International Conference on Microelectronics for Neural Networks*; 12–14 February 1996; Lausanne, Switzerland. New York, NY, USA: IEEE. pp. 311-316.
- [14] Han W, Han I. Tunable linear conductance by two MOSFETs and its application to analogue-mixed VLSI for mobile communications and biologically plausible neuromorphic hardware. In: *Proceedings of the 2009 WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communication*; 21–23 February 2009; Cambridge, UK. Sofia, Bulgaria: WSEAS. pp. 86-93.
- [15] Sanchez-Sinencio E, Linares-Barranco B. Circuit implementation of neural FitzHugh-Nagumo equations. In: *Proceedings of the 1989 Midwest Symposium on Circuits and Systems*; 14–16 August 1989; Champaign, IL, USA. New York, NY, USA: IEEE. pp. 244-274.
- [16] Linares-Barranco B, Sanchez-Sinencio E, Rodriguez-Vazquez A, Huertas JL. A CMOS implementation of FitzHugh-Nagumo neuron model. *IEEE J Solid-St Circ* 1991; 26: 956-965.
- [17] Cosp J, Binczak S, Madrenas J, Fernandez D. Implementation of compact VLSI FitzHugh-Nagumo neurons. In: *IEEE 2008 International Symposium on Circuits and Systems*; 18–21 May 2008; Seattle, WA, USA. New York, NY, USA: IEEE. pp. 2370-2373.
- [18] Rangan V, Ghosh A, Aparin V, Cauwenberghs G. A subthreshold a VLSI implementation of the Izhikevich simple neuron model. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*; 31 August–4 September 2010; Buenos Aires, Argentina. New York, NY, USA: IEEE. pp. 4164-4167.
- [19] Demirkol AS, Ozoguz S. A low power VLSI implementation of the Izhikevich neuron model. In: *IEEE 2011 International New Circuits and Systems Conference*; 26–29 June 2011; Bordeaux, France: IEEE. pp. 169-172.
- [20] Lee YJ, Lee J, Kim YB, Ayers J, Volkovskii A, Selverston A, Abarbanel H, Rabinovich M. Low power real time electronic neuron VLSI design using subthreshold technique. In: *Proceedings of the 2004 International Symposium on Circuits and Systems*; 23–26 May 2004; Vancouver, Canada. New York, NY, USA: IEEE. pp. 744-747.
- [21] Yu T, Sejnowski TJ, Cauwenberghs G. Biophysical neural spiking, bursting, and excitability dynamics in reconfigurable analog VLSI. *IEEE T Biomed Circ S* 2011; 5: 420-429.
- [22] Lewis ER. An electronic model of the neuron based on the dynamics of potassium and sodium ion fluxes. In: Reiss RF, editor. *Neural Theory and Modeling*. Stanford, CA, USA: Stanford University Press, 1964. pp. 154-189.
- [23] Binczak S, Jacquir S, Bilbault JM, Kazantsev VB, Nekorkin VI. Experimental study of electrical MFHN neurons. *Neural Networks* 2006; 19: 684-693.
- [24] Gotthans T, Petrzela J, Hrubos Z. Analysis of Hindmarsh-Rose neuron model and novel circuitry realization. In: *Proceedings of the 2011 International Conference Mixed Design of Integrated Circuits and Systems*; 16–18 June 2011; Gliwice, Poland. New York, NY, USA: IEEE. pp. 576-580.

- [25] Keener J, Sneyd J. *Mathematical Physiology*. New York, NY, USA: Springer, 1998.
- [26] FitzHugh R. Impulses and physiological states in theoretical models of nerve membrane. *Biophys J* 1961; 1: 445-466.
- [27] Bizzarri F, Linaro D, Storace M. PWL approximation of the Hindmarsh-Rose neuron model in view of its circuit implementation. In: 2007 European Conference on Circuit Theory and Design; 27–30 August 2007; Seville, Spain. New York, NY, USA: IEEE. pp. 878-881.
- [28] González-Miranda JM. Complex bifurcation structures in the Hindmarsh–Rose neuron model. *Int J Bifurcat Chaos* 2007; 17: 3071-3083.
- [29] Shi Y, Wang J, Deng B, Liu Q. Chaotic synchronization of coupled Hindmarsh-Rose neurons using adaptive control. In: 2009 International Conference on Biomedical Engineering and Informatics; 17–19 October 2009; Tianjin, China. New York, NY, USA: IEEE. pp: 1-5.
- [30] Che Y, Zhang S, Wang J, Cui S, Han C, Deng B, Wei X. Synchronization of inhibitory coupled Hindmarsh-Rose neurons via adaptive sliding mode control. In: 2011 International Conference on Intelligent Control and Information Processing; 25–28 July 2011; Harbin, China. New York, NY, USA: IEEE. pp: 1134-1139.
- [31] Zhao J, Kim YB. Circuit implementation of FitzHugh-Nagumo neuron model using field programmable analog arrays. In: Proceedings of the 2007 Midwest Symposium on Circuits and Systems; 5–8 August 2007; Montreal, Canada. New York, NY, USA: IEEE. pp. 772-775.
- [32] Dahasert N, Öztürk İ, Kılıç R. Experimental realizations of the HR neuron model with programmable hardware and synchronization applications. *Nonlinear Dynam* 2012; 70: 2343-2358.
- [33] La Rosa M, Caruso E, Fortuna L, Frasca M, Occhipinti L, Rivoli F. Neuronal dynamics on FPGA: Izhikevich's model. In: Conference on Bioengineered and Bioinspired Systems II; 29 June 2005; Seville, Spain. Bellingham, WA, USA: SPIE. pp. 87-94.
- [34] Rice KL, Bhuiyan MA, Taha TM, Vutsinas CN, Smith MC. FPGA implementation of Izhikevich spiking neural networks for character recognition. In: International Conference on Reconfigurable Computing and FPGAs; 9–11 December 2009; Quintana Roo, Mexico. New York, NY, USA: IEEE. pp. 451-456.
- [35] Şekerli M, Butera RJ. An implementation of a simple neuron model in field programmable analog arrays. In: 2004 Annual International Conference of the IEEE on Engineering in Medicine and Biology Society; 1–5 September 2004; San Francisco, CA, USA. New York, NY, USA: IEEE. pp. 4564-4567.