

OPEN ACCESS

EDITED BY
Jiyong Woo,
Kyungpook National University,
Republic of Korea

REVIEWED BY
Shuangming Yang,
Tianjin University, China
Wooseok Choi,
IBM, Switzerland

*CORRESPONDENCE
Mostem Heidarpur
✉ heidar@uwindsor.ca

RECEIVED 04 November 2023
ACCEPTED 19 December 2023
PUBLISHED 28 February 2024

CITATION
Heidarpur M, Ahmadi A and Ahmadi M (2024)
The silence of the neurons: an application to
enhance performance and energy efficiency.
Front. Neurosci. 17:1333238.
doi: 10.3389/fnins.2023.1333238

COPYRIGHT
© 2024 Heidarpur, Ahmadi and Ahmadi. This
is an open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](#). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

The silence of the neurons: an application to enhance performance and energy efficiency

Moslem Heidarpur^{1*}, Arash Ahmadi² and Majid Ahmadi¹

¹Department of Computer and Electrical Engineering, University of Windsor, Windsor, ON, Canada,
²Department of Electronics, Carleton University, Ottawa, ON, Canada

Introduction: Simulation of biological neural networks is a computationally intensive task due to the number of neurons, various communication pathways, and non-linear terms in the differential equations of the neuron.

Method: This study proposes an original modification to optimize performance and power consumption in systems, simulating or implementing spiking neural networks. First, the proposed modified models were simulated for validation. Furthermore, digital hardware was designed, and both the original and proposed models were implemented on a Field-Programmable Gate Array (FPGA).

Results and discussion: Moreover, the impact of the proposed modification on performance metrics was studied. The implementation results confirmed that the proposed models are considerably faster and require less energy to generate a spike compared with unmodified neurons.

KEYWORDS

Izhikevich neuron, biological neuron, CORDIC, digital implementation, neuromorphic, spiking neural network, FPGA

1 Introduction

Inspired by the brain, Spiking Neural Networks (SNNs) represent the third generation of neural networks where neurons communicate through sparse sequences of spikes. In comparison to classic Artificial Neural Networks (ANNs), SNNs have a higher learning and processing speed by considering the timing of events as another variable (Pfeiffer and Pfeil, 2018). Furthermore, SNNs are orders of magnitude more energy-efficient due to a low spiking rate and activity. Such networks are also useful for studying biological neural networks and brain diseases (Pastur-Romay et al., 2016). New learning algorithms for spiking neural network are also evolving (Yang and Chen, 2023a,b; Yang et al., 2023b). Yang et al. (2024) proposes a neuromorphic architecture designed for online learning through dendrites to improve the efficiency of spike-driven learning and enhance the performance of processing spatiotemporal patterns. A novel fault-tolerant address event representation approach is proposed the study by Yang et al. (2023a) for the spike information routing to improve the efficiency and reliability of smart traffic navigation.

Several computational models have been proposed to mimic biological neurons (Hodgkin and Huxley, 1990; Gerstner and Kistler, 2002; Izhikevich, 2003; Brette, 2005). In these models, there is often a trade-off between being concise and computationally efficient or being biologically plausible and complex.

Simulation of SNNs is computationally complex due to the number of neurons, various communication pathways, and non-linear terms in the differential equations describing

these models. As implementation platforms, researchers have used PCs (NeMo, 2024; The Brian Simulator, 2024), supercomputers (EPFL, 2024), analog (Schemmel et al., 2010; Covi et al., 2015), digital (Merolla et al., 2011; Akopyan et al., 2015; Barchi et al., 2019), mixed analog/digital VLSI neuromorphic hardware (Benjamin et al., 2014; Wang et al., 2014; Neckar et al., 2019), and FPGAs (Neil and Liu, 2014; Liu et al., 2018; Heidarpur et al., 2019), to simulate and realize SNNs. Each platform comes with its own set of strengths and weaknesses. Nevertheless, in all aforementioned platforms, enhancing speed and minimizing energy consumption are very important.

An important contributing factor to the computational complexity of neuron models is calculating non-linear terms in their differential equations. In view of this, researchers have proposed a variety of approaches to speed up systems simulating or implementing spiking neurons where, in general, accuracy is exchanged for performance. Such methods include Piece-Wise Linear (PWL) approximation (Yamashita and Torikai, 2014; Heidarpur et al., 2017), Coordinate Rotation Digital Computer (CORDIC) (Heidarpour et al., 2016; Elnabawy et al., 2018), an asynchronous cellular automaton used in Matsubara and Torikai (2013), a nonlinear function evaluation technique (Jokar et al., 2019), bit-serial reduced-range multipliers (Karim et al., 2017; Kueh and Kazmierski, 2017), and a novel rotate-and-fire neuron (Hishiki and Torikai, 2011), among others.

Power dissipation and density are another important concern and one of the major challenges that need to be resolved for the massive large-scale implementation of neuromorphic systems. Total power dissipation is the sum of two components: static and dynamic power dissipation. Dynamic power is associated with activity and switching events in the core or I/O of the device (Rabaey et al., 2003). Power optimization could be at the circuit, logic, architectural, or system levels (Devadas and Malik, 1995). Various methods, mostly at the circuit and logic levels, are suggested to reduce power consumption of spiking neural networks (Lee et al., 2004; Indiveri et al., 2006; Tao and Rusu, 2015; Kohno and Aihara, 2016). Analysis of static power, which is independent of circuit activity and primarily from transistor leakage, is out of the scope of this study.

For computer simulation and digital hardware implementation, the differential equations describing spiking neurons are discretized and numerically solved by evaluating them at every time step, even when a neuron is silent. In this study, through the observation of both the input current and the derivative of the membrane potential, we discovered that the amount of computation could be reduced when a neuron is silent or spikes at a slow rate. Based on this insight, we proposed a modification to avoid computing certain terms of the differential equations in the spiking neuron model. During a fast spiking state, the neuron can switch back to the full calculation of Ordinary Differential Equations (ODEs). The proposed technique has the potential to save energy and time, making it particularly valuable when implementing large networks. In the case of software implementation, the proposed improvement over spiking neuron can reduce the total number of times that ODEs are evaluated and therefore reduce the total number of computation required for simulating neuron. In this study, software simulations were performed to measure

the improved efficiency resulted from applying proposed method. Additionally, in the case of hardware implementation, the proposed method reduces the total number of computation required and also saves the energy consumption, which is the most useful if devices are operated on battery. The main focus of the paper remains the advantages of proposed method for hardware implementation.

As a case study, the Izhikevich neuron model (Izhikevich, 2003) was modified using the proposed technique and further simulated for validation. Additionally, a network consisting of both the original and modified models was developed, trained, and implemented on hardware to study the effects of the proposed modification on both accuracy and performance.

The rest of the study is organized as follows: Section 2 reviews the Izhikevich neuron and presents the proposed method to save computations in neuronal differential equations. Section 3 investigates the validity of the proposed models through error analysis and studies their impacts on computer simulation performance. Section 4 discusses the FPGA implementation procedure and how the proposed modification optimizes energy consumption and speed of hardware. Finally, Section 5 concludes the study.

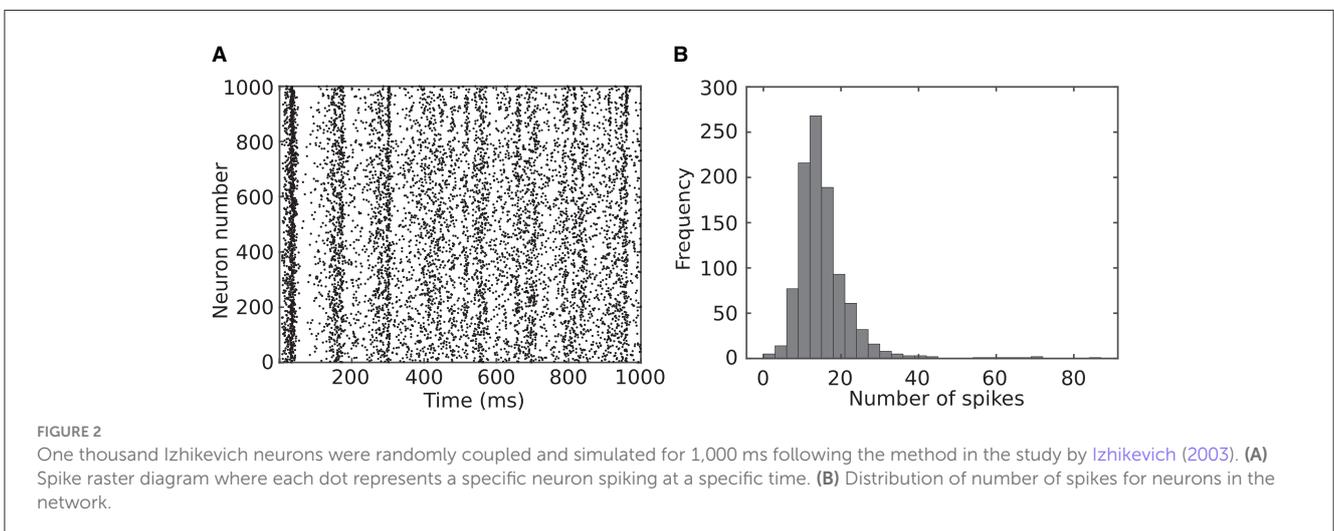
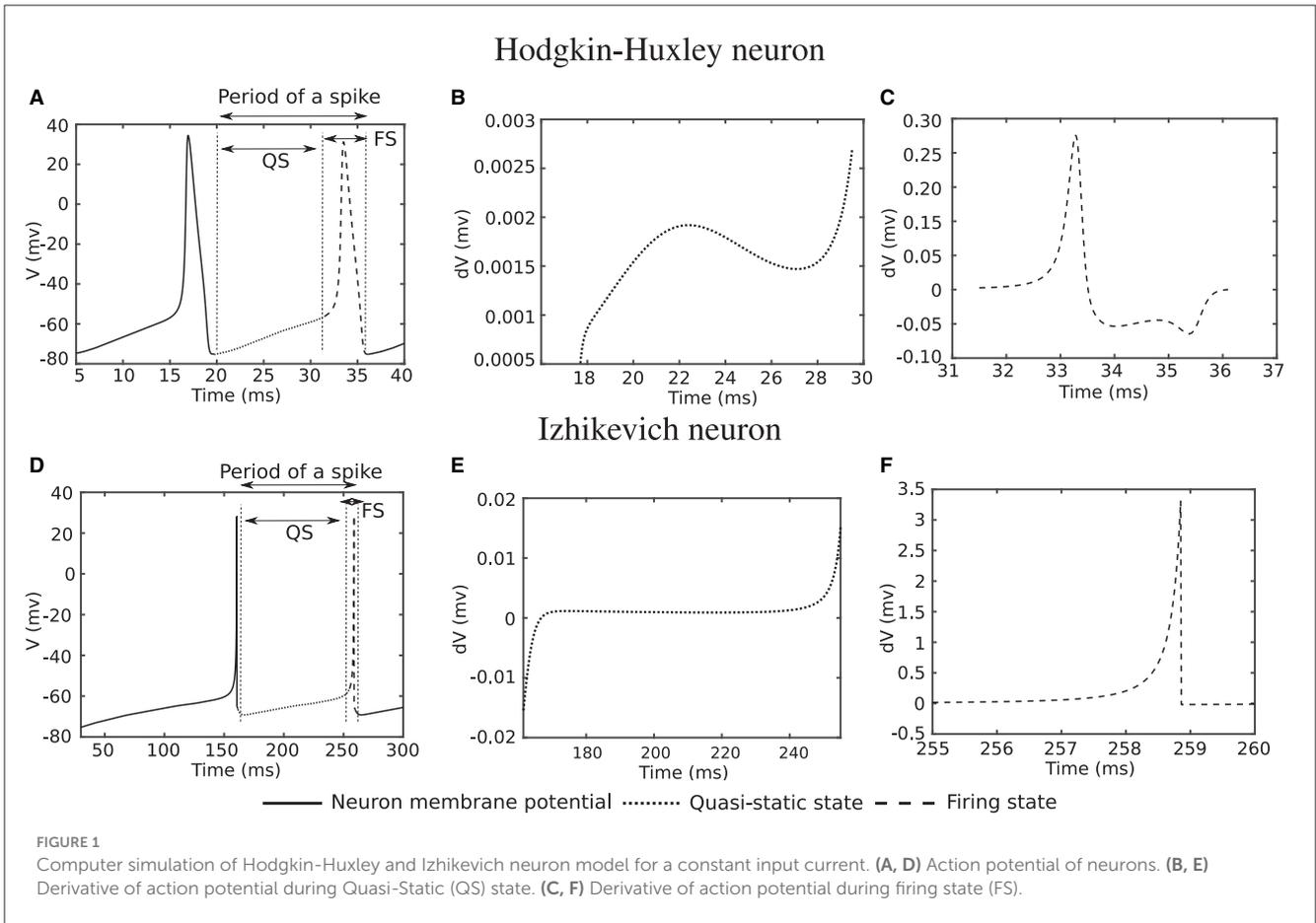
2 Proposed duplex neuron

This section presents a modification to optimize performance and power consumption in systems simulating or implementing spiking neural networks.

2.1 Background

Researchers have presented various models to simulate and study the behavior of biological neurons. These models are formulated as coupled differential equations that need continuous evaluation over time, which typically involves numerical methods since spiking neuron models do not have analytical solutions. Furthermore, bifurcation analysis helps to study qualitative changes in the dynamics of neuron as a function of certain parameters, such as synaptic strengths, time constants, or external inputs (Izhikevich, 2007). One of the bifurcation parameters, determining the states of a neuron, transitioning from silent to firing, and influencing its rate, is the input current. In Figure 1, the simulation of two well-known spiking neuron models, Hodgkin–Huxley (HH) (Hodgkin and Huxley, 1990) and Izhikevich (Izhikevich, 2003), is depicted for a constant input current. The Hodgkin–Huxley (HH) neuron model is a mathematical model that describes the generation and propagation of action potentials, or spikes, in biological neurons, whereas the Izhikevich neuron model is a simplified model that aims to capture essential features of neuronal dynamics while minimizing computational complexity

In Figure 1A, the membrane potential (V) of a Hodgkin–Huxley neuron is demonstrated when the input current (or its integral over time) is sufficiently high in transition of the neuron from silence to firing. The derivative of the membrane potential over the dotted line in Figure 1A is shown in Figure 1B. In this state, which we further refer to as the quasi-static (QS) state,



the derivative tends to remain almost constant with very small changes. The derivative of the membrane potential over the dashed line in Figure 1A is depicted in Figure 1C. The value of the derivative for this state [hereafter referred to as the Firing State (FS)] is considerably larger compared with that over the dotted line. The same scenario also applies to the Izhikevich neuron, with the corresponding simulations shown in Figures 1D–F. What is interesting is that for the Hodgkin–Huxley neuron, the quasi-static

state makes up ~75% of the spiking period, which is a significant portion. This percentage for the Izhikevich neuron is even higher, roughly ~95%. The Izhikevich neuron is in the firing state for only ~5% of a spike period.

The previous paragraph studied neuron behaviors while spiking consistently. To further investigate the behaviors of spiking neuron models, 1,000 Izhikevich neurons were randomly coupled and simulated according to the method utilized in reference

(Izhikevich, 2003). Figure 2A shows the spike raster diagram for this network. Raster diagram provides a visual representation of the timing of action potentials or spikes across multiple neurons over time. Each vertical line in this diagram represents the occurrence of a spike, and the horizontal axis corresponds to time. Raster diagrams provide an intuitive and informative way to visualize the precise timing of spikes from individual neurons and can be used to study the patterns of spike synchrony or specific temporal relationships. Coherent activity could be observed as vertical columns of dots, which is similar to the temporal synchrony (alpha and gamma band rhythms) of neurons observed in biology (Vaidya and Johnston, 2013).

Distribution of number of the spikes for the neurons in this network is presented in Figure 2B. As this figure indicates, most of the neurons (almost 90 %) spike between 6 and 24 times during 1 s of simulation and are silent rest of the time. This indicates the potential of the proposed modification to save computation in a spiking neural network.

In the following sections, Izhikevich neuron was considered as case study, where the proposed method was applied to a single neuron and also an spiking neural network to evaluate performance.

2.2 The duplex neuron

Izhikevich neuron is a two-dimensional model consists of two coupled ODEs as follows:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \quad (1)$$

$$\frac{du}{dt} = a(bv - u) \quad (2)$$

along with an after-spike reset equation as follows:

$$\text{if } v > 30 \text{ mv then } \begin{cases} v \rightarrow v_r \\ u \rightarrow w_r = u + d. \end{cases} \quad (3)$$

where v represents membrane potential, u is recovery variable, and I stands for injected current to the neuron. Other dimensionless parameters are as follows:

- a : Time scale of the recovery variable
- b : Sensitivity of the u to v
- c : After-spike reset value of v
- d : After-spike reset value of u

By regulating these parameters, the Izhikevich model can mimic different neuronal behaviors observed in biological neurons. To simulate this model, Equations 1, 2 were discretized as follows:

$$\begin{aligned} v[n+1] &= (0.04 v[n]^2 + 5v[n] + 140 - u[n] + I[n])dt + v[n] \\ u[n+1] &= a(bv[n] - u[n])dt + u[n] \end{aligned} \quad (4)$$

and numerically solved using the Euler method.

Figure 3A shows simulation of a tonic spiking Izhikevich neuron stimulated with a constant input current. The neuron parameters for this simulation are: $a = 0.02$, $b = 0.2$, $I = 4$ mA, and $dt = 1/32$ ms. Figure 3B plots the derivative of the membrane potential ($v[n+1] - v[n]$). As shown in this figure, the derivative is approximately constant and close to zero (quasi-static state), except when the neuron membrane goes up for a spike (firing state). The small value of the derivative for the quasi-static state implies a small change in action potential when the neuron is in the quasi-static state or when it is silent. To benefit from this property, Equation 4 was modified as follows:

```

1: if |v[n+1] - v[n]| > δ then
2:   α ← 0.04 v[n]^2 + 140 - u[n]
3:   β ← a(bv[n] - u[n])
4: else
5:   do nothing
6: end if
7: v[n+1] = (α + 5v[n] + I[n]) dt + v[n]
8: u[n+1] = (β) dt + u[n]

```

Algorithm 1. Proposed duplex neuron.

When the neuron is silent or in the quasi-static state, α and β do not change significantly, and the last values of these parameters can be used to evaluate neuron ODEs instead of calculating them every time. The modified model is hereafter referred to as the Duplex (DX) neuron. The term $5v[n]$ was excluded from α since the coefficient is relatively larger than other parameters, considering that it constant affects neuron behaviors. The parameter δ is the threshold value that determines whether it is necessary to update α and β or if the last updated value is still valid. A larger value of delta results in greater computational savings but induces higher error and vice versa. Therefore, determining the proper δ is a compromise between accuracy and performance. In the following, we investigate this trade-off.

3 Computer simulation

This section investigates the impacts of proposed modification on error and performance of a single and a network of Izhikevich neurons.

3.1 Error analysis

The objective of this section is to investigate whether the neurons with the proposed modification still have a valid behavior, which is similar to unmodified neuron.

3.1.1 Qualitative comparison

First, for qualitative comparison, proposed duplex Izhikevich neurons were simulated for different values of the δ , as shown in Figure 4. As shown in this figure, even for greater values of delta such as $\delta = 0.2$ mv, besides small oscillations, the membrane

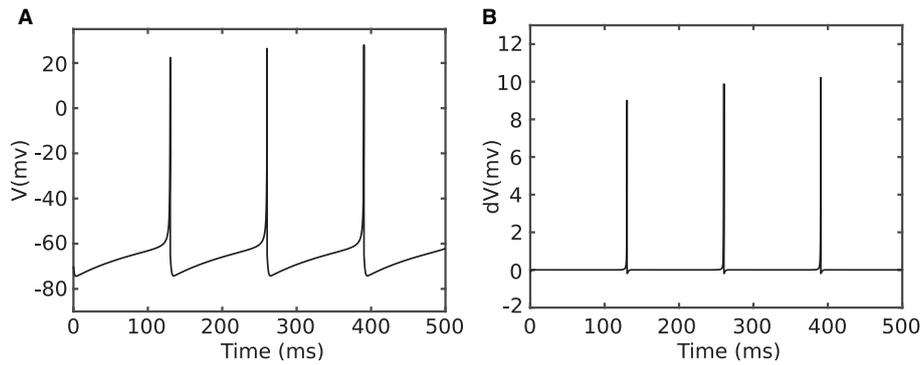


FIGURE 3 Simulation of a tonic spiking Izhikevich neuron which is stimulated with a constant current (Equation 4). (A) Membrane potential $v[n]$ and (B) membrane potential derivative $v[n + 1] - v[n]$.

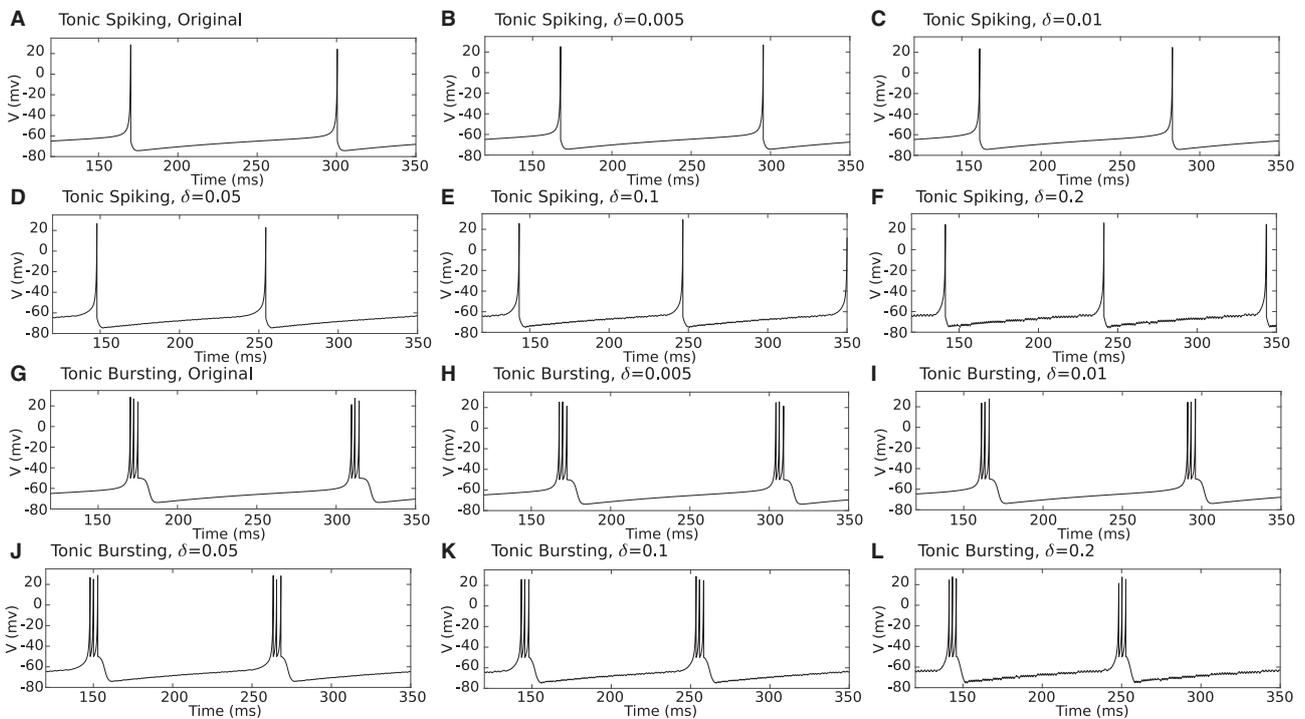


FIGURE 4 (A–L) Computer simulation of a tonic spiking/bursting Izhikevich and proposed duplex Izhikevich neuron for different values of δ . As it is evident in this figure, even for large values of δ such as 0.2 mV, besides small oscillations, waveform of the membrane potentials (v) are close to those of the unmodified model.

potential waveform of the proposed duplex neuron is close to that of the model which constantly evaluates the entire differential equation. Nevertheless, proposed modification evidently results in some discrepancies in timing of spikes.

3.1.2 Quantitative comparison

In the following, the deviations in the previous section were quantified by calculating the following errors:

Normalized root mean square deviation (NRMSD): This error was calculated to measure the deviation induced by different values of δ to the shape of the membrane potential waveform. Higher

resemblance between the form of the spikes results in a lower value for this error. This error is calculated as follows:

$$NRMSD = \frac{\sqrt{\frac{\sum_{i=1}^n (v_{Un}(n) - v_{DX}(n))^2}{n}}}{v_{Unmax} - v_{Unmin}} \times 100 \quad (5)$$

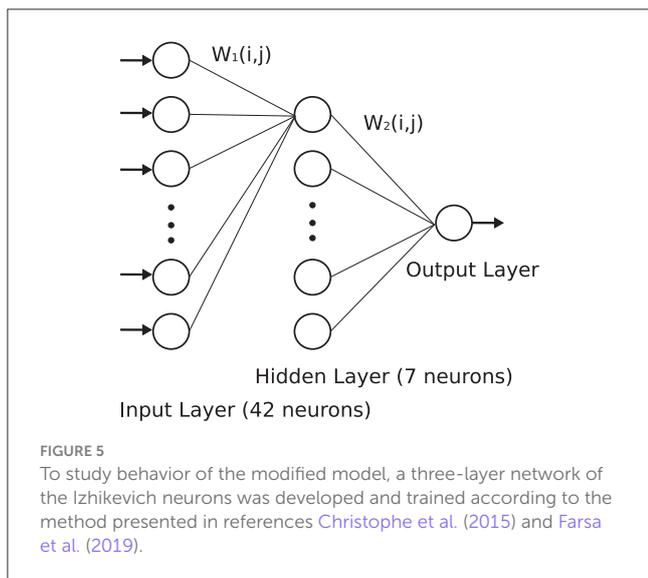
where V_{DX} and V_{Un} are waveforms of the duplex and unmodified Izhikevich neurons. v_{Unmax} and v_{Unmin} are the minimum and maximum of membrane potential (v) between $v[1]$ and $v[n]$, where n is the total number of points that this error is evaluated.

Timing error (TE): This error measures the difference in the time

TABLE 1 Errors, computer saving percentage, and speed up percentage for computer simulation of proposed duplex neuron on the basis of δ (mv).

δ (mv)	TE% (Eq. 6)	NRMSD% (Eq. 5)	CSP% (Eq. 7)	Speed up % (Eq. 8)
0.001	0.04	0.11	00.3	2.52
0.005	1.87	0.12	58.2	10.47
0.010	6.67	0.31	85.3	16.78
0.050	18.25	1.12	91.8	18.19
0.100	20.74	1.51	92.3	18.31
0.200	23.22	1.79	93.7	18.53

The neuron was stimulated with a constant input current of 4 mA.



interval between two consecutive spikes and is calculated as follows:

$$TE = \left| \frac{\Delta ts_{Un} - \Delta ts_{DX}}{\Delta ts_{Un}} \right| \times 100 \tag{6}$$

where Δts_{Un} and Δts_{DX} are spike periods of the unmodified and proposed duplex Izhikevich neuron.

These errors were calculated for a tonic spiking Izhikevich neuron which was simulated with an input current of 4 mA and a time step of 1/32 ms and are presented in Table 1. As data in this table indicate, these errors are higher for larger values of δ . However, the increase rate almost levels out for $\delta > 0.01$ mv.

3.1.3 Network behavior

To study the behavior of the proposed duplex model in an application, a basic three-layer network of Izhikevich neurons was constructed, as shown in Figure 5. The network comprises 42 neurons in the first layer, seven neurons in the second layer, and, finally, one neuron in the output layer. This network was trained using the Spike Time Dependent Plasticity (STDP) rule to recognize two patterns of E and H , following the method used in references Christophe et al. (2015) and Farsa et al. (2019). First, weights were initialized randomly. Furthermore, the network was

stimulated with the pattern E during the training phase. In the next phase, both patterns of E and H were applied randomly to the input of the network to test its validity. For more details, please refer to references Christophe et al. (2015) and Farsa et al. (2019).

To evaluate the behavior of the proposed models, the network was trained with both duplex and original Izhikevich neurons. Figures 6A, C show the membrane potential of the output neuron for the original and the modified model with $\delta = 0.05$ mV. The corresponding raster diagram for the networks during the training and testing phases is shown in Figures 6B, D. In the previous section, the input current of the neuron was presumed to be constant. On the other hand, in a practical application or biology, the input current is not steady and may abruptly change. The results of this test confirm that the behavior of the proposed duplex neuron remains close to the original model even with interrupted input currents.

3.2 Performance analysis

This section investigates the impacts of proposed modification on computer simulation performance.

3.2.1 Single neuron

In the proposed duplex neuron, there is no need to compute α and β terms (Algorithm 1) constantly. The percentage of iterations that these terms are computed in the duplex neuron to the total number of iterations determines the performance improvement. Therefore, Computation Saving Percentage (CSP) was defined and formulated as follows:

$$CSP = \frac{\text{Number of iterations } \alpha \text{ and } \beta \text{ are not computed}}{\text{Total number of iterations}} \times 100 \tag{7}$$

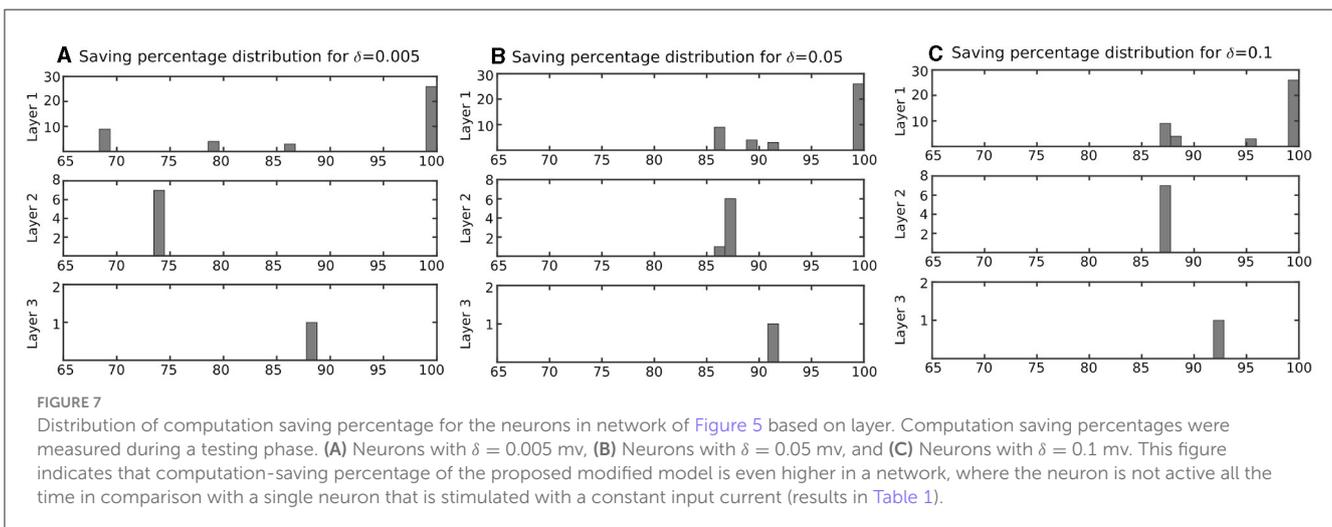
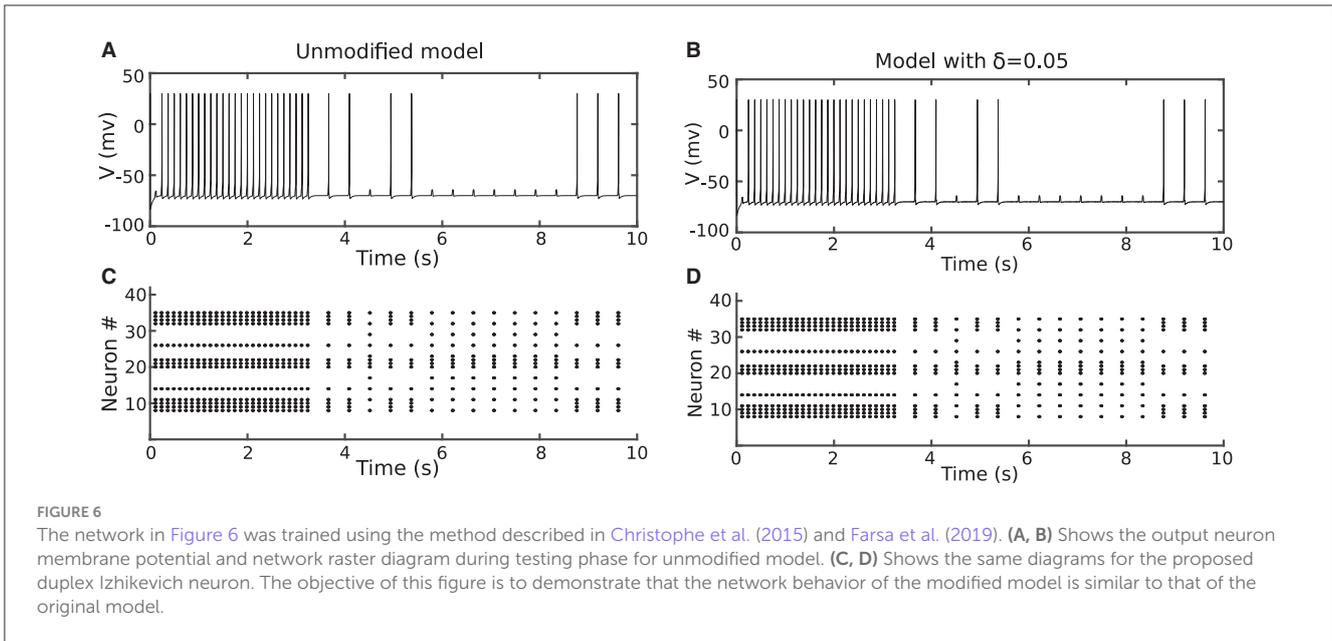
CSP for different values of δ is presented in Table 1. According to the results in this table, for $\delta > 0.01$, more than 90% of the time, there is no need to constantly calculate the square function in Izhikevich neuron ODEs. To study performance improvement, original and duplex neurons with different values of δ were simulated for 20 s in MATLAB software, and total execution times were measured to calculate the Speed Up Percentage (SUP) as follows:

$$\text{Speed up} = \frac{T_{DX} - T_{Un}}{T_{Un}} \times 100 \tag{8}$$

where T_{DX} and T_{Un} are execution time of proposed duplex and unmodified Izhikevich model, respectively. Speed up percentage for various values of δ is presented in Table 1. As data suggest, speed up is higher for larger values of δ . Nevertheless, following the pattern of errors, speed up also levels out after $\delta > 0.01$ mv.

3.2.2 Neurons in a network

The neuron in the previous section was stimulated with a constant current to spike continuously. However, as discussed before, this is not the case in a network of spiking neurons. Therefore, it is expected that the proposed duplex neuron features even better performance improvement in a network. To investigate



this, the computation saving percentage for each neuron in the network, as shown in Figure 5, was calculated during the testing phase. The results for neurons in each layer are presented in Figure 7 on the basis of δ .

Figure 7 column (A) shows the computation saving percentage for duplex neurons with $\delta = 0.005$ mV. Such CSPs are much higher than those calculated for the neurons with a constant input current (Table 1). Indeed, even for the neurons with relatively small δ , which have very low error, the computation saving percentage is considerable in a practical application.

3.2.3 Input current

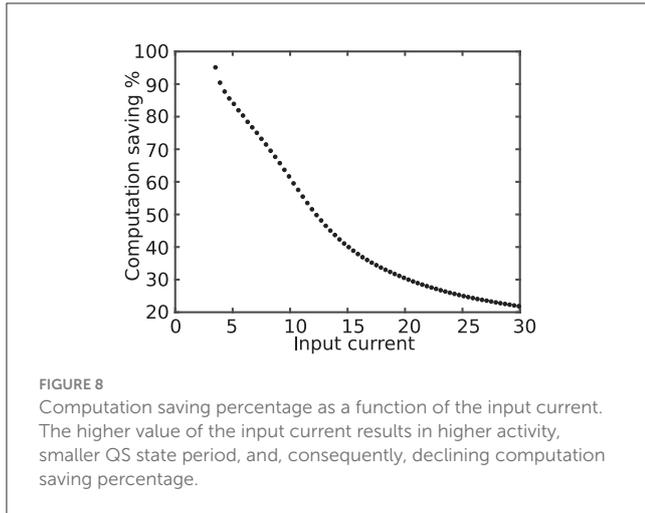
Higher spike rate results in shorter QS period during a spike. This, in turn, leads to lower computation saving percentage and, as a consequence, lower performance improvement.

Figure 8 shows the computation saving percentage as a function of the input current. As it is evident in this figure, computation saving percentage is lower for larger input currents.

In contrary, the quasi-static state makes up the majority of the time for the neurons that rarely fire. Therefore, for silent neurons, the proposed modification saves more computation. In addition to improving execution time, proposed modification also saves energy by decreasing switching activity during simulation.

4 Hardware implementation

This section discusses hardware design and implementation of the proposed duplex neuron models on FPGA and interpretation of results.



4.1 Hardware design

The Data Flow Graph (DFG) for digital implementation of the discretized Izhikevich equations (Equation 4) and reset equation (Equation 3) is shown in Figure 9.

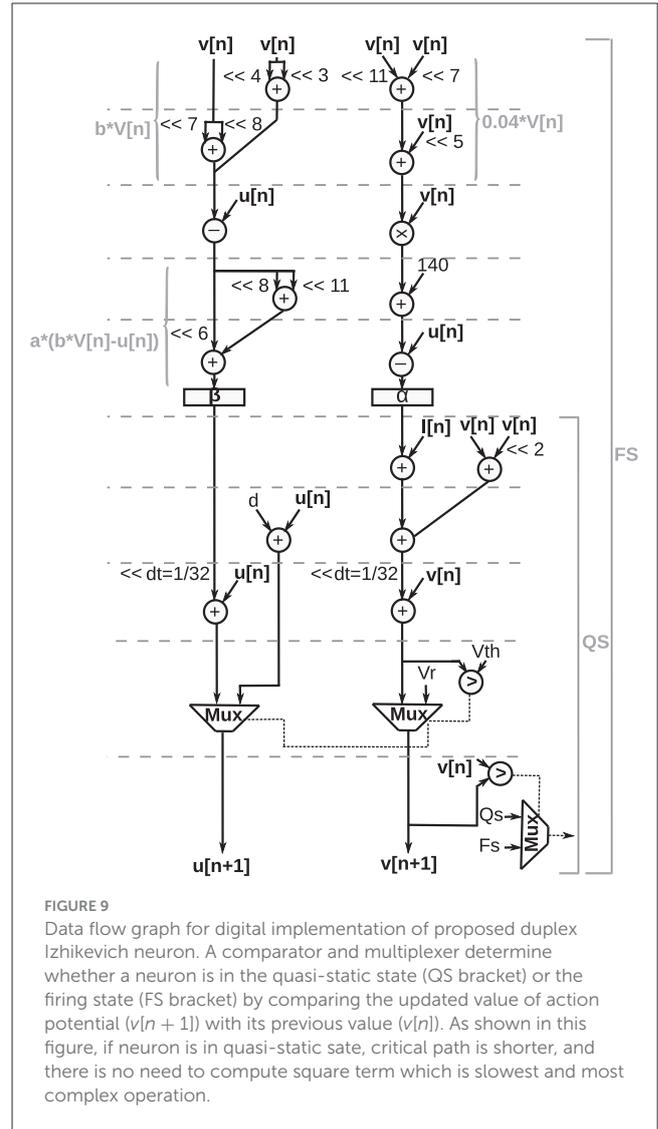
In this illustration, addition, subtraction, and multiplication operations are represented as circles. A comparator and multiplexer determine whether the neuron is in the quasi-static state or the firing state by comparing the updated value of the action potential ($v[n+1]$) with its previous value ($v[n]$). As shown in this figure, if the neuron is in the quasi-static state, the critical path is shorter, and there is no need to compute the square term, which is the slowest and most complex operation.

Values of α and β are stored in two registers (shown as rectangles). The operation units are scheduled in a way that no more than two adders are needed at any time. Arithmetic shift operations are denoted by “ \ll ” symbol, where “ $\ll 2$ ” indicates that data shifted twice to the left. Multiplication with constants is performed using add and shift operations. For instance, $0.04 \times v[n]$ was calculated as follows:

$$0.04 \times v[n] \approx (0.0396)v[n] = (2^{-11} + 2^{-7} + 2^{-5})v[n] \quad (9)$$

$$= (v[n] \ll 11) + (v[n] \ll 7) + (v[n] \ll 5)$$

Such approximation causes a small error; however, it considerably improves the performance of the design. A CORDIC algorithm, as presented in the study by Heidarpur et al. (2019), was used to perform the square function in neuron ODEs. Fixed-point arithmetic was utilized since fixed-point units are considerably cheaper and faster compared with floating-point units. Furthermore, the word length of the design was determined considering the number of integer bits to represent variables in their domain and the number of fraction bits for the minimum required precision. Additional bits were also added to avoid overflow or underflow. Taking these requirements into consideration, a 30-bit word length was specified, comprised of a 14-bit fraction and a 16-bit integer part.

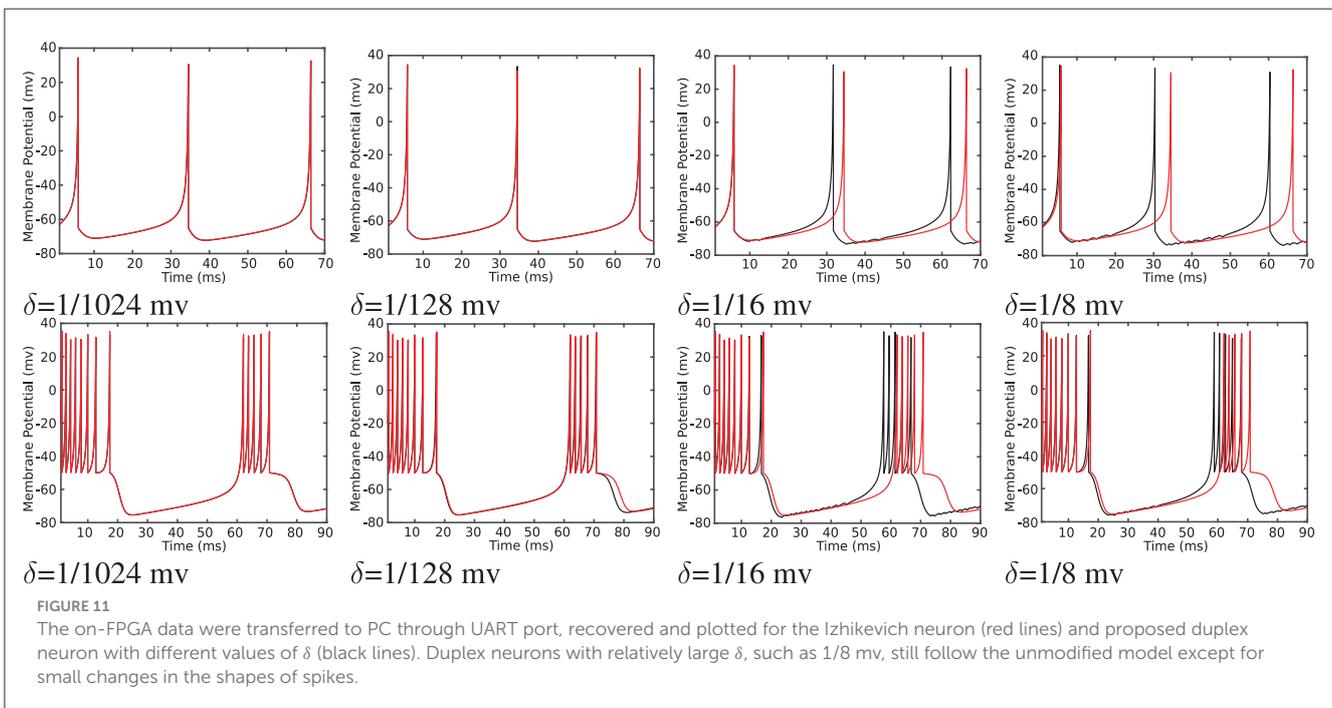
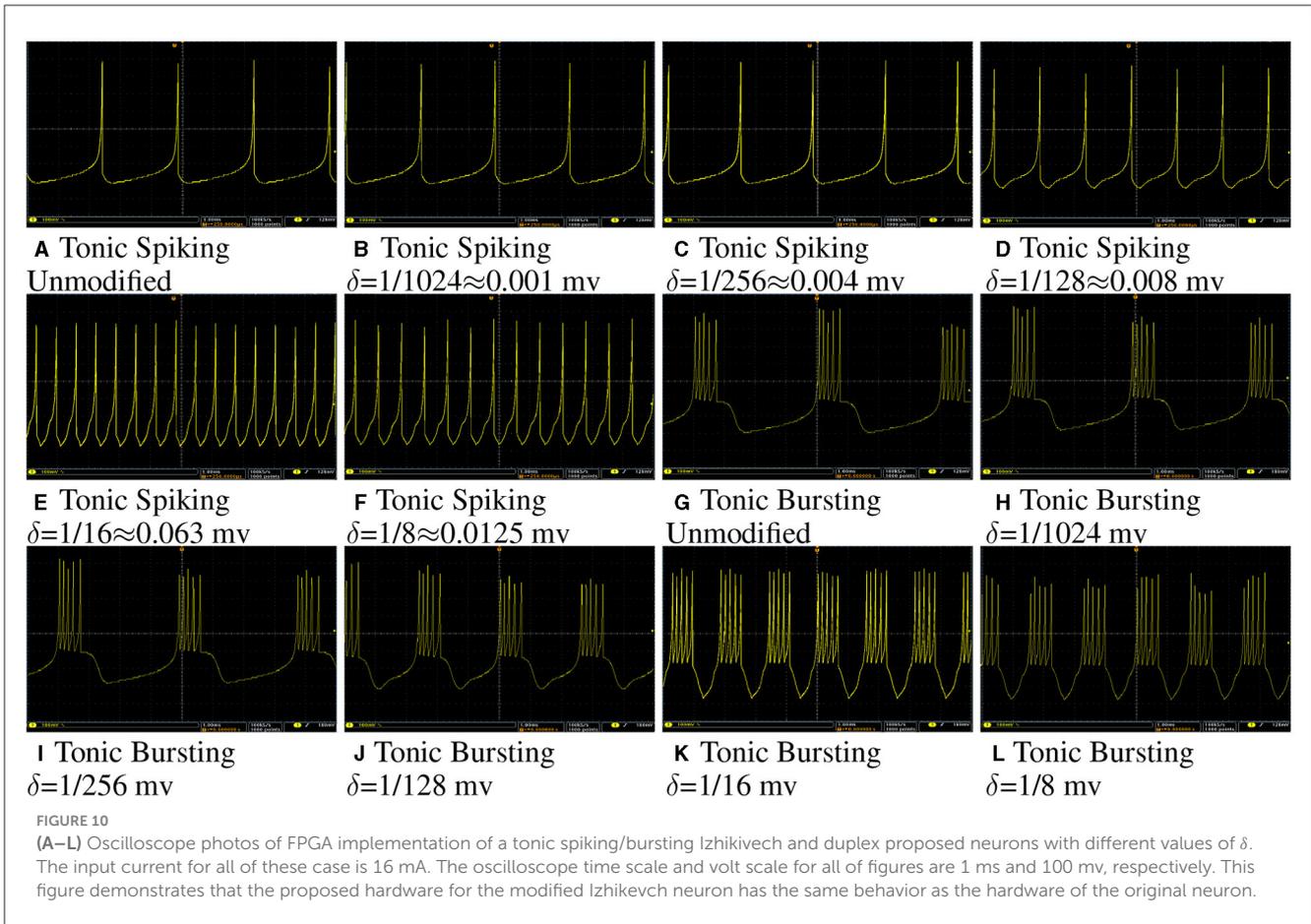


4.2 Hardware implementation

To implement the design on FPGA, the architecture, as shown in Figure 9, was modeled in Very High-speed integrated circuit Hardware Description Language (VHDL). The design was verified by simulating and testing using ModelSim software. Subsequently, the HDL description was synthesized and configured for FPGA implementation using Xilinx Integrated Synthesis Environment (ISE) software tool.

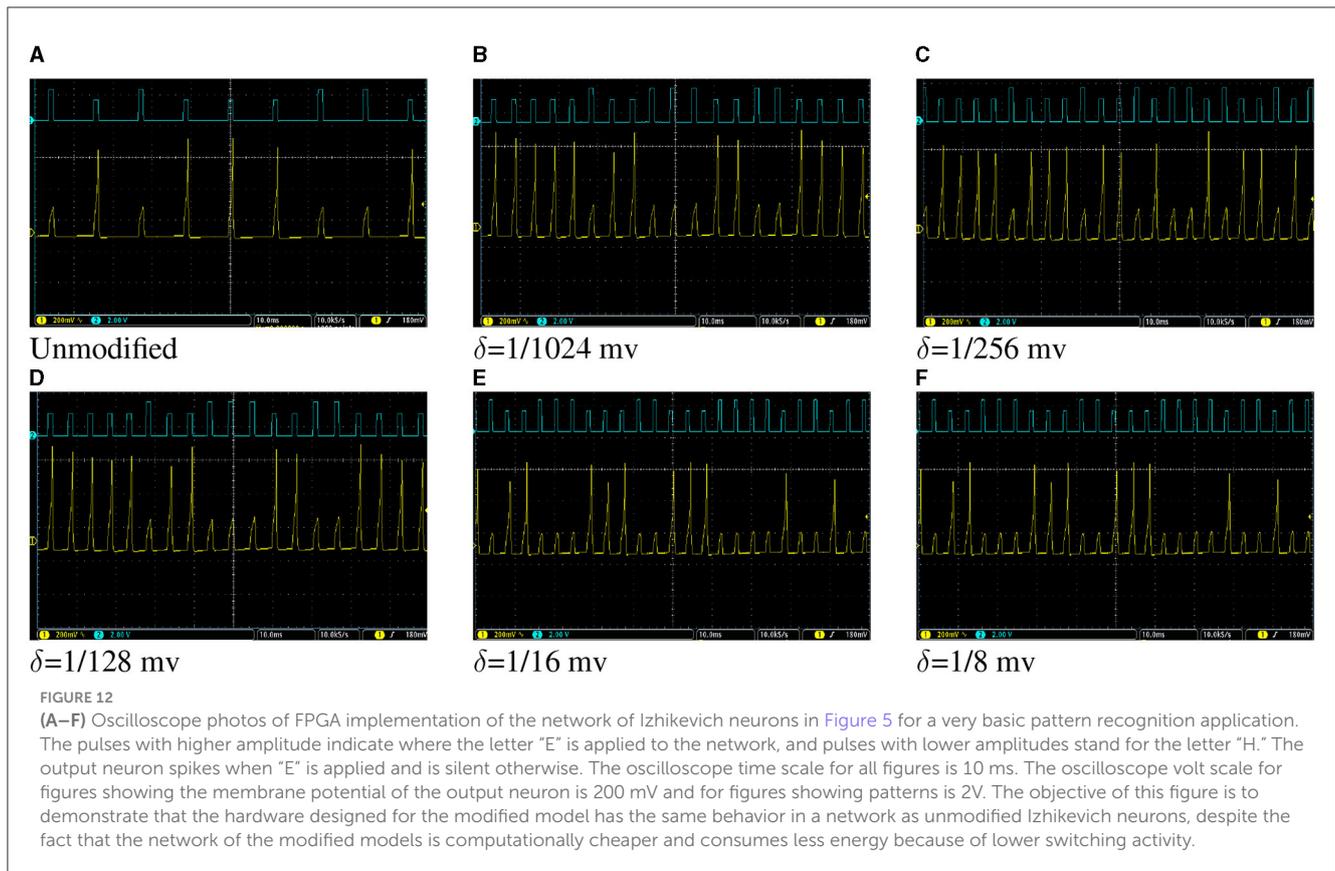
4.2.1 A single neuron

Figure 10 shows membrane potential of a tonic spiking/bursting Izhikevich and a duplex proposed neuron for different values of δ on Spartan 6 XC6LX75 FPGA. The data were converted to analog using a 12 bit Digital to Analog Converter (DAC).



As this figure demonstrates, even for large values of δ , such as 0.128 mv, the duplex model still follows the original model. However, by increasing δ , one can observe some differences in the

shapes of spikes. The input current for all neurons is the same, but spike rate for neurons with larger δ are higher since they have higher throughput.



To transfer data from FPGA to PC for further analysis, a Universal Asynchronous Receiver-Transmitter (UART) module was developed and implemented on FPGA as described in the reference Heidarpur et al. (2019). Transferred data are presented in Figure 11. As it is evident in this figure, the proposed and unmodified model has a similar membrane potential waveform apart from some deviations in the shape of spikes. These implementation results are in agreement with computer simulation results in Computer Simulation Section, as presented in Figure 4 and Table 1.

4.2.2 A network of neurons

The network, as shown in Figure 5, was described in VHDL and implemented on Spartan 6 XC6LX75 FPGA. Figure 12 shows oscilloscope photos of membrane potential of output neuron of the network while tests are applied to the network. Figure 12A shows the result for the network of the unmodified Izhikevich neurons where the output neuron spikes for “E” and is silent for “H.” Thereafter, the same network with proposed duplex neurons was implemented on FPGA. While implementing modified models with different values of δ , we faced two unexpected challenges.

First, in the unmodified Izhikevich neuron, all neurons generate new output after a certain number of the clock cycles and have the same throughput. However, for the duplex neuron, this number could be different for each neuron depending on its input weights. Transferring data from one layer to the next layer requires that all neurons action potentials of that layer to be updated. Therefore, the slowest neuron determines the critical path, and faster neurons must wait in idle state for others to finish.

Second, the proposed modification results in discrepancy in timing of the spikes since the duplex neurons are faster than unmodified neurons. On the other hand, SNNs are primarily based on timing. Because of that, the network, initially, did not function properly with the weights calculated based on the original neuron and was not able to recognize all instances of input patterns. For larger values of δ , where the speed up was more dramatic, overall network failure rate was higher. To resolve this issue, the weights were recalculated by applying STDP rule to network of duplex neurons. Figure 12 shows oscilloscope photos of FPGA implementation of the trained network, where two test patterns are applied to network. It could be observed in this figure that the on-FPGA network of duplex neurons, even for large values of δ , has the same behavior as the network of original Izhikevich neurons.

4.3 Results and discussion

The objective of the proposed modification is to reduce the number of the clock cycles required to generate an output. This contributes to increasing speed and reducing power consumption of the circuit. Table 2 compares resource utilization and operation frequency for FPGA implementation of the original and duplex modified Izhikevich neuron. The proposed modification does not result in a notable increase in resources that are needed to implement the neuron hardware on FPGA. Furthermore, resources and frequency are comparable to similar studies. Some studies reported low resources but they use extensive approximations and are not very accurate replicators of Izhikevich neuron. Table 3

TABLE 2 Comparison between resource utilization and frequency of the proposed method and previously published studies.

References	Slice registers	Slice LUT's	Frequency	DSPs	Device
Soleimani et al. (2012)	493	617	241.9	0	Virtex-II Pro XC2VP30
Haghiri et al. (2018)	490	459	240	0	Virtex-II Pro
Yang et al. (2020)	130	119	291.8	-	ZCU102
Jokar et al. (2019)	195	198	285	0	Virtex-6
Grassia et al. (2014)	646	1,048	105	22	Virtex-5 XC5VLX50
Heidarpour et al. (2016)	829	1,221	134.3	0	Spartan-6 XC6SLX9
Shimada and Torikai (2015)	357	1,776	-	-	Zync-7000 XC7Z020
Original Izhikevich	270	469	212.8	0	Spartan-6 XC6SLX75
Duplex Izhikevich	297	532	171.9	0	Spartan-6 XC6SLX75

Objective of this study is not to reduce resource utilization or increase the frequency of design but rather to skip unnecessary computations to save power and reduce latency of the neuron.

TABLE 3 On FPGA dynamic power, total number of clock cycles and total time required for unmodified and proposed DX Izhikevich neurons with different values of δ (mv) to generate a spike.

	Unmodified	$\delta = 1/1,024$	$\delta = 1/256$	$\delta = 1/128$	$\delta = 1/16$	$\delta = 1/8$
Dynamic power (mW)	11	6	6	6	6	6
Number of clocks	122,400	120,788	114,704	73,780	37,312	32,200
Total time (ms)	2.45	2.42	2.30	1.47	0.74	0.65

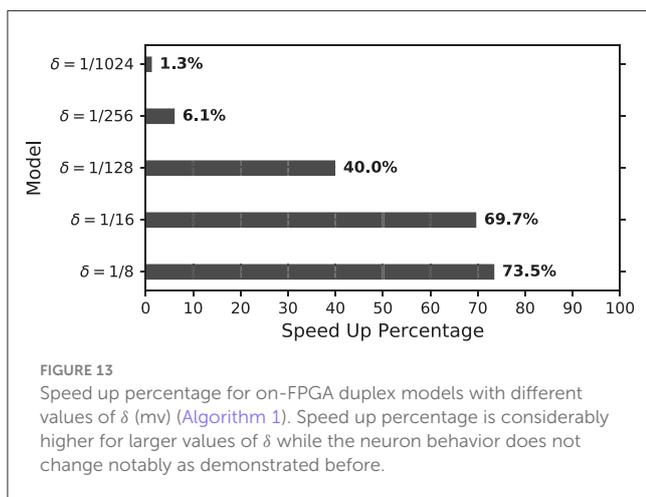


FIGURE 13 Speed up percentage for on-FPGA duplex models with different values of δ (mv) (Algorithm 1). Speed up percentage is considerably higher for larger values of δ while the neuron behavior does not change notably as demonstrated before.

shows on-FPGA dynamic power consumption reported by Xilinx Power Estimator (XPE), and the number of the clock cycles and time is required to generate a spike.

4.3.1 Speed

As data in Table 3 indicate, proposed modification, especially for larger values of δ , considerably decreases total computation time per spike. Furthermore, speed up percentage was calculated according to Equation 8 and compared as follows

$$Speed\ up = \frac{T_{Un} - T_{DX}}{T_{Un}} \times 100 \tag{10}$$

where T_{Un} and T_{DX} are total computation time per spike for unmodified and duplex Izhikevich neuron. This percentage was

calculated and compared for different values of δ , as shown in Figure 13. As this figure indicates, models with $\delta = 1/8$ mv and $1/16$ mv require almost 70% less computation time to generate a spike.

4.3.2 Power dissipation

Reported by XPE, original Izhikevich model dynamic power on FPGA is almost double of those of the modified models. These powers were calculated for operation frequency of 50 MHz, which is the frequency of on-board oscillator. The total energy consumption per spike (E_{ps}) could be calculated as follows:

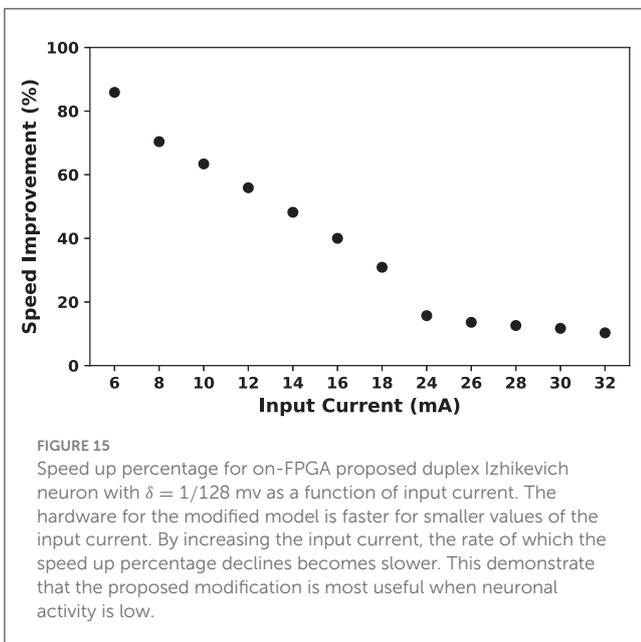
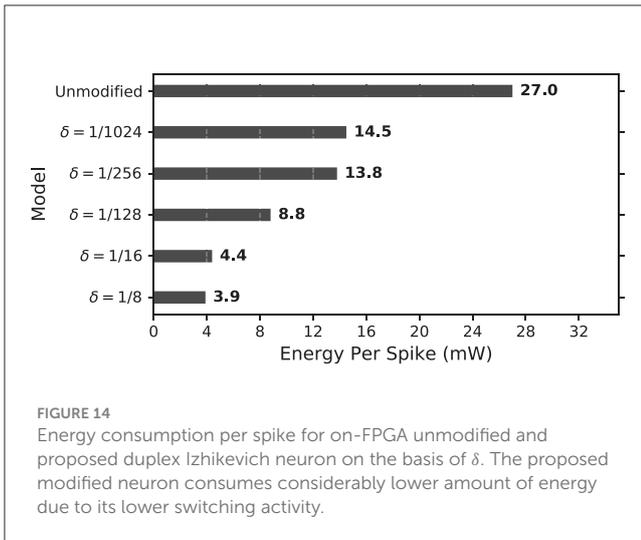
$$E_{ps} = P_d T_s \tag{11}$$

where P_d is dynamic power and T_s is total time to generate an spike.

E_{ps} for unmodified and proposed duplex Izhikevich models with different values of δ is presented in Figure 14. As results indicate, proposed models considerably reduce energy consumption per spike. This is because proposed modification not only decreases switching activity of the circuit but it also reduces total computing time. This applies to the trained network as well. As shown in Figure 12, the network with $\delta = 1/8$ mv can process patterns almost 2.5 times faster, which also results in less switching and energy consumption as well.

4.3.3 Impact of input current

To calculate the total number of clocks required to generate a spike and energy consumption measurements, it was assumed that input current of the neuron is 16 mA. Changing the input current will affect the results that presented in the previous section. As discussed in the previous sections, increasing input current results



in a higher spike rate and decreasing ratio of quasi-static state period to the total period of a spike. This, in turn, leads to a lower speed up percentage.

Figure 15 shows on-FPGA speed up percentage (Equation 10) for the duplex neurons as a function of the input current. As this figure demonstrates, SUP is considerably higher for smaller input currents. This denotes that the proposed modification is most useful when neuronal activity is low.

5 Conclusion

In this study, a novel modification to neuron differential equations was presented to avoid unnecessary computation while simulating neurons on computers or implementing them on

hardware. The proposed method is inspired by biology and benefits from the fact that most biological neurons are either silent or fire with a very slow rate. First, the proposed models were simulated for validation both as a single neuron and as a part of a network of neurons trained using STDP. Second, the impact of the proposed modification was studied on computer simulation performance in terms of the total time required to simulate neurons. The results show that the proposed modification can avoid unnecessary computations from 20 to 90 % and speed up simulation time from 2 to 18 % depending on the value of δ , input current, and time step. Furthermore, hardware was designed, described in VHDL, and simulated and implemented on FPGA. Implementation results indicated that the proposed modification accelerates the speed of hardware up to 70 %. The results for energy consumption also revealed that the proposed models can reduce energy consumption per spike from 50 % to 1/7th of its value for an unmodified neuron.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

MH: Writing – original draft, Writing – review & editing. AA: Conceptualization, Supervision, Validation, Writing – review & editing. MA: Conceptualization, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., et al. (2015). Truenorth: design and tool flow of a 65 mw 1 million neuron programmable neuromorphic chip. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 34, 1537–1557. doi: 10.1109/TCAD.2015.2474396
- Barchi, F., Urgese, G., Siino, A., Di Cataldo, S., Macii, E., and Acquaviva, A. (2019). Flexible on-line reconfiguration of multi-core neuromorphic platforms. *IEEE Trans. Emerg. Topics Comput.* 9, 915–927. doi: 10.1109/TETC.2019.2908079
- Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J. M., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* 102, 699–716. doi: 10.1109/JPROC.2014.2313565
- Brette, R. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.* 94, 3637–3642. doi: 10.1152/jn.00686.2005
- Christophe, F., Mikkonen, T. J., Andalibi, V., Koskimies, K., and Laukkanen, T. (2015). “Pattern recognition with spiking neural networks: a simple training method,” in *14th Symposium on Programming Languages and Software Tools* (Tampere).
- Covi, E., Brivio, S., Fanciulli, M., and Spiga, S. (2015). Synaptic potentiation and depression in al: Hfo2-based memristor. *Microelectron. Eng.* 147, 41–44. doi: 10.1016/j.mee.2015.04.052
- Devadas, S., and Malik, S. (1995). “A survey of optimization techniques targeting low power vlsi circuits,” in *Proceedings of the 32nd annual ACM/IEEE Design Automation Conference* (New York, NY: ACM), 242–247. doi: 10.1145/217474.217536
- Elnabawy, A., Abdelmohsen, H., Moustafa, M., Elbediwy, M., Helmy, A., Mostafa, H., et al. (2018). “A low power cordic-based hardware implementation of Izhikevich neuron model,” in *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)* (Montreal, QC: IEEE), 130–133. doi: 10.1109/NEWCAS.2018.8585485
- EPFL (2024). *Home*. Available online at: <https://www.epfl.ch/en/> (accessed February 15, 2024).
- Farsa, E. Z., Ahmadi, A., Maleki, M. A., Gholami, M., and Rad, H. N. (2019). A low-cost high-speed neuromorphic hardware based on spiking neural network. *IEEE Trans. Circuits Syst. II Express Briefs* 66, 1582–1586. doi: 10.1109/TCSII.2019.2890846
- Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511815706
- Grassia, F., Levi, T., Kohno, T., and Saghi, S. (2014). Silicon neuron: digital hardware implementation of the quartic model. *Artif. Life Robot.* 19, 215–219. doi: 10.1007/s10015-014-0160-2
- Haghiri, S., Zahedi, A., Naderi, A., and Ahmadi, A. (2018). Multiplierless implementation of noisy Izhikevich neuron with low-cost digital design. *IEEE Trans. Biomed. Circuits Syst.* 12, 1422–1430. doi: 10.1109/TBCAS.2018.2868746
- Heidarpour, M., Ahmadi, A., and Rashidzadeh, R. (2016). A cordic based digital hardware for adaptive exponential integrate and fire neuron. *IEEE Trans. Circuits Syst. I: Regul. Pap.* 63, 1986–1996. doi: 10.1109/TCSI.2016.2598161
- Heidarpur, M., Ahmadi, A., Ahmadi, M., and Rahimi Azghadi, M. (2019). CORDIC-SNN: On-FPGA STDP learning with Izhikevich neurons. *IEEE Trans. Circuits Syst. I: Regul. Pap.* 66, 2651–2661. doi: 10.1109/TCSI.2019.2899356
- Heidarpur, M., Ahmadi, A., and Kandalaf, N. (2017). A digital implementation of 2D hindmarsh-rose neuron. *Nonlinear Dyn.* 89, 2259–2272. doi: 10.1007/s11071-017-3584-0
- Hishiki, T., and Torikai, H. (2011). A novel rotate-and-fire digital spiking neuron and its neuron-like bifurcations and responses. *IEEE Trans. n Neural Netw.* 22, 752–767. doi: 10.1109/TNN.2011.2116802
- Hodgkin, A. L., and Huxley, A. F. (1990). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Bull. Math. Biol.* 52, 25–71. doi: 10.1016/S0092-8240(05)80004-7
- Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.* 17, 211–221. doi: 10.1109/TNN.2005.860850
- Izhikevich, E. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440
- Izhikevich, E. M. (2007). *Dynamical Systems in Neuroscience*. Cambridge, MA: MIT Press.
- Jokar, E., Abolfathi, H., and Ahmadi, A. (2019). A novel nonlinear function evaluation approach for efficient FPGA mapping of neuron and synaptic plasticity models. *IEEE Trans. Biomed. Circuits Syst.* 13, 454–469. doi: 10.1109/TBCAS.2019.2909043
- Karim, S., Harkin, J., McDaid, L., Gardiner, B., Liu, J., Halliday, D. M., et al. (2017). “Assessing self-repair on FPGAs with biologically realistic astrocyte-neuron networks,” in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (Bochum: IEEE), 421–426. doi: 10.1109/ISVLSI.2017.80
- Kohno, T., and Aihara, K. (2016). A three-variable ultralow-power analog silicon neuron circuit. *Neural Netw.* 3, 6. doi: 10.34385/proc.48.A3L-G-1
- Kueh, S. M., and Kazmierski, T. (2017). “A dedicated bit-serial hardware neuron for massively-parallel neural networks in fast epilepsy diagnosis,” in *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)* (Bethesda, MD: IEEE), 105–108. doi: 10.1109/HIC.2017.8227595
- Lee, Y. J., Lee, J., Kim, Y. B., Ayers, J., Volkovskii, A., Selverston, A., et al. (2004). “Low power real time electronic neuron vlsi design using subthreshold technique,” in *2004 IEEE International Symposium on Circuits and Systems*, Volume 4 (Vancouver, BC: IEEE), IV–744. doi: 10.1109/ISCAS.2004.1329111
- Liu, J., Harkin, J., Maguire, L. P., McDaid, L. J., and Wade, J. J. (2018). Spanner: a self-repairing spiking neural network hardware architecture. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 1287–1300. doi: 10.1109/TNNLS.2017.2673021
- Matsubara, T., and Torikai, H. (2013). Asynchronous cellular automaton-based neuron: theoretical analysis and on-fpga learning. *IEEE Trans. Neural Netw. Learn. Syst.* 24, 736–748. doi: 10.1109/TNNLS.2012.2230643
- Merolla, P., Arthur, J., Akopyan, F., Imam, N., Manohar, R., Modha, D. S., et al. (2011). “A digital neuromorphic core using embedded crossbar memory with 45pJ per spike in 45nm,” in *2011 IEEE Custom Integrated Circuits Conference (CICC)* (San Jose, CA: IEEE), 1–4. doi: 10.1109/CICC.2011.6055294
- Neckar, A., Fok, S., Benjamin, B. V., Stewart, T. C., Oza, N. N., Voelker, A. R., et al. (2019). Braindrop: a mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proc. IEEE* 107, 144–164. doi: 10.1109/JPROC.2018.2881432
- Neil, D., and Liu, S. (2014). Minitaur, an event-driven fpga-based spiking neural accelerator. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 22, 2621–2628. doi: 10.1109/TVLSI.2013.2294916
- Nemo (2024). *Home*. Available online at: <https://nemosim.sourceforge.net/> (accessed February 15, 2024).
- Pastur-Romay, L., Cedron, F., Pazos, A., and Porto-Pazos, A. (2016). Deep artificial neural networks and neuromorphic chips for big data analysis: pharmaceutical and bioinformatics applications. *Int. J. Mol. Sci.* 17, 1313. doi: 10.3390/ijms17081313
- Pfeiffer, M., and Pfeil, T. (2018). Deep learning with spiking neurons: opportunities and challenges. *Front. Neurosci.* 12, 774. doi: 10.3389/fnins.2018.00774
- Rabaey, J. M., Chandrakasan, A. P., and Nikolić, B. (2003). *Digital Integrated Circuits: A Design Perspective, Volume 7*. Upper Saddle River, NJ: Pearson Education.
- Schemmel, J., Brüderle, D., Gribbl, A., Hock, M., Meier, K., Millner, S., et al. (2010). “A wafer-scale neuromorphic hardware system for large-scale neural modeling,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (Paris: IEEE), 1947–1950. doi: 10.1109/ISCAS.2010.5536970
- Shimada, N., and Torikai, H. (2015). A novel asynchronous cellular automaton multicompartiment neuron model. *IEEE Trans. Circuits Syst. II Express Briefs* 62, 776–780. doi: 10.1109/TCSII.2015.2433471
- Soleimani, H., Ahmadi, A., and Bavandpour, M. (2012). Biologically inspired spiking neurons: piecewise linear models and digital implementation. *IEEE Trans. Circuits Syst. I: Regul. Pap.* 59, 2991–3004. doi: 10.1109/TCSI.2012.2206463
- Tao, S., and Rusu, A. (2015). A power-efficient continuous-time incremental sigma-delta adc for neural recording systems. *IEEE Trans. Circuits Syst. I: Regul. Pap.* 62, 1489–1498. doi: 10.1109/TCSI.2015.2418892
- The Brian Simulator (2024). *Home*. Available online at: <https://briansimulator.org/> (accessed February 15, 2024).
- Vaidya, S. P., and Johnston, D. (2013). Temporal synchrony and gamma-to-theta power conversion in the dendrites of ca1 pyramidal neurons. *Nat. Neurosci.* 16, 1812. doi: 10.1038/nn.3562
- Wang, R. M., Hamilton, T. J., Tapson, J., and van Schaik, A. (2014). A mixed-signal implementation of a polychronous spiking neural network with delay adaptation. *Front. Neurosci.* 8, 51. doi: 10.3389/fnins.2014.00051
- Yamashita, Y., and Torikai, H. (2014). Theoretical analysis for efficient design of a piecewise constant spiking neuron model. *IEEE Trans. Circuits Syst. II: Express Briefs* 61, 54–58. doi: 10.1109/TCSII.2013.2285982
- Yang, S., and Chen, B. (2023a). SNIB: improving spike-based machine learning using nonlinear information bottleneck. *IEEE Trans. Syst. Man Cybernet. Syst.* 53, 78527863. doi: 10.1109/TSMC.2023.3300318
- Yang, S., Liu, P., Xue, J., Sun, R., and Ying, R. (2020). “An efficient fpga implementation of Izhikevich neuron model,” in *2020 International SoC Design Conference (ISOCC)* (Yeosu: IEEE), 141–142. doi: 10.1109/ISOCC50952.2020.9333014
- Yang, S., Liu, P., Xue, J., Sun, R., Tan, J., Lei, T., and Linares-Barranco, B. (2023a). Smart traffic navigation system for fault tolerant edge computing of internet of vehicle in intelligent transportation gateway. *IEEE Trans. Intell. Transp. Syst.* 24, 13011–13022. doi: 10.1109/TITS.2022.3232231

Yang, S., Wang, H., and Chen, B. (2023b). Sibols: robust and energy-efficient learning for spike-based machine intelligence in information bottleneck framework. *IEEE Trans. Cogn. Dev. Syst.* 1–13. doi: 10.1109/TCDS.2023.3329532

Yang, S., Wang, H., Pang, Y., Azghadi, M. R., and Linares-Barranco, B. (2024). NADOL: neuromorphic architecture for spike-driven online learning by

dendrites. *IEEE Trans. Biomed. Circuits Syst.* 18, 186199. doi: 10.1109/TBCAS.2023.3316968

Yang, S. and Chen, B. (2023b). SNIB: improving spike-based machine learning using nonlinear information bottleneck. *IEEE Trans. Syst. Man Cybernet. Syst.* 53, 7852–7863. doi: 10.1109/TSMC.2023.3300318