# A New Digital Low Power Spiking Neuron

Moshe Bensimon, Shlomo Greenberg, Yehuda Ben-Shimol, and Moshe Haiut

***Abstract*—This article presents an innovative reconfigurable parametric model of a digital Spiking Neuron (SN). The model is based on the classical Leaky Integrate and Fire (LIF) model with some unique modifications, allowing neuron configuration using seven different leak modes and three activation functions with dynamic threshold setting. Efficient hardware implementation of the proposed spiking neuron significantly reduces the area and power cost. The SN model is implementable requiring only 700 ASIC 2-inputs gates and is denser than IBM SN which is composed of 1272 gates. The proposed spiking neuron model can be expanded to support a larger recurrent neural network and is efficiently applied to audio applications. Performance evaluation has been carried out through a simple voice activation system design using 1000 SNs, demonstrating ultra-low power consumption of only 20uW and consuming an area of 0.03 mm2 using 28nm technology. Simulations of the proposed digital spiking neuron also demonstrate its ability to accurately replicate the behaviors of a biological neuron model.**

***Index Terms*—Spiking neuron, digital neuron, recurrent neural network, LIF model, LSTM, low power, voice activation.**

## I. Introduction

In a digital world where the information rate is growing increasingly, the demand for real-time non-linear machine learning methods capable of processing an extensive amount of data is high as ever [1]. Various kind of classical and spiking neural networks-based architectures have been recently proposed to support processing of vast streams of information in real time [2].

Internet of things (IOT) and mobile devices are penetrating our life rapidly and becoming dominant platforms [3]. Therefore, the necessity for low-power and high-performance embedded platforms is increasing [4]-[6]. Spiking neural networks (SNN) which are characterized by low-power consumption are able to substitute classical NN architectures and have the potential to achieve a higher level of energy efficiency [7]. S. Furber et al. [8] present the SpiNNaker manycore hardware architecture model and demonstrate the ability to integrate one billion of spiking neurons in real time. The work of F. Akopyan et al. [9] introduces a reconfigurable hardware architecture which aggregates one million spiking neurons using the TrueNorth platform.

In similar to the recurrent neural network (RNN) and the long short time memory (LSTM) architecture, SN-based neural networks are supposed to be more efficient while processing a serial type of incoming data, utilizing the high correlation between adjacent frames while applied to voice and video applications [10].

This paper presents an innovative reconfigurable parametric model of a digital Spiking Neuron (SN). The proposed neuron model is based on the common Leaky Integrate-and-Fire (LIF) model [11] with some unique modifications, allowing a neuron configuration using seven different leak modes and three activation functions with dynamic threshold setting. The neuron is efficiently implemented in hardware demonstrating ultra-low power consumption.

The proposed spiking continuous time neuron (SCTN)[12] is designed to operate as a basic block in a continuous time recurrent neural network (CTRNN) [13].

This paper is organized as follows. Sec. II presents some background for different neurons types and compares the classical and spiking neurons. Sec. III describes the mathematical model of the proposed spiking neuron. The specifications of the neuron model are presented in Sec. IV and Sec. V conclude the paper.

## II. Back Ground

This section briefly reviews the four types of neuron models: the classical, spiking, analog, and digital neuron models. From an architecture point of view, spiking neurons are very similar to classical neurons. The implementation of spiking neural networks (SNN) can be carried out using both analog and digital circuits.

A classical neuron sums the weighted inputs performing multiplications and additions and using Lookup Table (LUT) to implement a sigmoidal-like activation function. Spiking neurons receive a sequence of spikes, distributed in time. Since spikes are actually binary values, the weighted inputs are accumulated directly into the neuron kernel with no need for hardware multipliers [7].

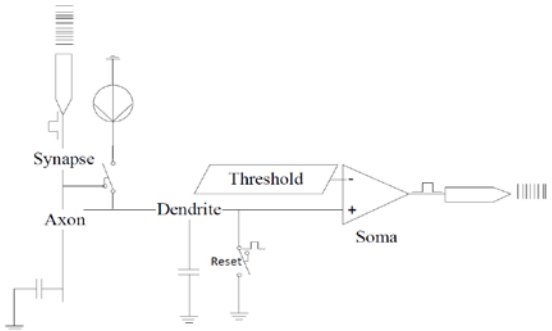


Fig. 1. Digital neuron architecture.

### A. Digital vs. Analog Neuron

While an analog implementation requires less area (factor of 5) and consumes less power (factor of 20), the digital

Manuscript received September 13, 2018; revised January 29, 2019.

The author are with the Ben-Gurion University, Israel (e-mail: bensimmo@post.bgu.ac.il, shlomog@bgu.ac.il, benshimo@bgu.ac.il, Moshe.Haiut@dspg.com).

doi: 10.18178/ijfcc.2019.8.1.534

model is simply characterized and is more appropriate for large-scale circuits, VLSI and mass production [14]. Fig. 1 illustrates a typical digital neuron implementation. Each incoming spike (across the axon) is sampled by a latch (synapse) and increments a digital counter (dendrite). Once the accumulated number of pulses crosses a given threshold (membrane potential) a pulse is provided in the neuron output.

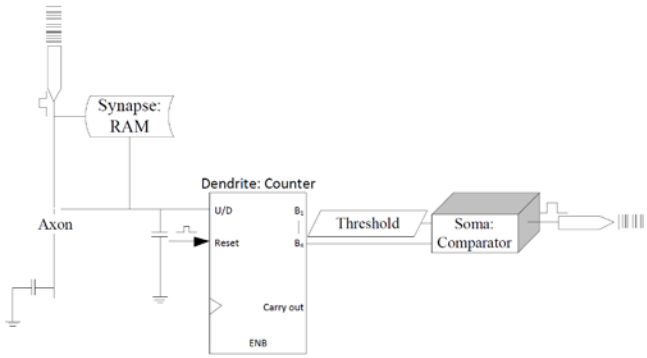


Fig. 2. Analog neuron architecture.

Fig. 2 depicts a general analog interpretation of the neuron architecture. The dendrite potential is represented by a capacitor which is responsible for integrating the incoming pulses. A spike is triggered out by the analog comparator (soma), in case the capacitor voltage reaches a given threshold and the membrane potential is resetting to an initial value by discharging the capacitor [15].

### B. Classical vs. Spiking Neuron

A comprehensive comparison between classical and spiking neurons in term of silicon area and power consumption is given in [7]. Low energy consumption is believed to be an inherent feature of SNN while efficiently implemented in silicon. The authors present a comparative analysis, considering two levels: individual neuron and a full system level. For a fair comparison, it is assumed that a synaptic connection of a spiking neuron is equivalent to one input of a classical neuron, and the network structure for both neuron types are similar. Table 1 is derived from [7] to demonstrate the comparison results for the individual neuron in term of area and power consumption at 65nm. The power consumption for the full system comparison is given in Table 2 for digital and analog architectures at 65nm and 28nm.

TABLE I: CLASSICAL VS. SPIKING NEURON [7]

| Program | Tech [nm] | Area [mm$^2$] | Power[uW] |
|---|---|---|---|
| Classical neuron [7] | 65 | 18,643 | 8 |
| Spiking neuron [14] | 65 | 538 | 4 |

TABLE II: A FULL SYSTEM COMPARISON

| Program | architecture | Tech [nm] | Benchmark energy, J |
|---|---|---|---|
| Classical [16] | Digital | 65 | $10^{-8}$ |
| Spiking | Analog [7] | 65 | $10^{-7}$- $5{\cdot}10^{-6}$ |
| | Digital [4] | 28 | $5{\cdot}10^{-6}$ |

A comparison of analog and digital implementations [7], [14], [17] shows that the analog design consumes only 2PJ while the digital design consumes 41PJ. Classical digital neurons consume of tens of pJ per one synapse activation while spiking neurons consume only a few PJ [7].

## III. THE PROPOSED SPIKING NEURON MODEL

The mathematical model of the suggested neuron offers a more complex representation comparing to the basic LIF model. First, we briefly describe the basic LIF model and then present the SNN model, followed by an examination of our neuron model.

### A. The Leaky Integrate and Fire Neuron Model

The LIF model is a simple common spiking neuron model. In this model, the membrane potential depends on the incoming synaptic inputs that arrive from other neurons, each weighted by the respective synaptic strength. These inputs are modeled by a change in the membrane conductance, for which the summation of the synaptic input is nonlinear [11]. Since the membrane potential decays over time, the neuron is considered to be leaky. When the membrane potential crosses a predefined threshold $V_{th}$ it is reset to an initial value $V_{reset,}$ and an output spike is generated.

The LIF mathematical model is described by (1).

$$\begin{aligned} &Y(t) = Y(t-1) + \sum_{i=1}^{N} s_i(t) w_{ij} \\ &Y(t) = Y(t) - \mu_j \\ &If\ \ Y(t) = Y_{threshold} \rightarrow Spike \text{ and set } Y(t) = Y_{reset} \end{aligned} \tag{1}$$

where, $V(t)$ is the membrane potential at time $t$, $S_i(t)$ is the binary synaptic inputs (stream of spikes), and $W_{ij}$ represents the corresponding inputs weights. The neuron leakage is described by the constant parameter $\mu_j$ .

### B. The Neuron Mathematical Model

Fig. 3 illustrates the mathematical model for the proposed SNN neuron. The proposed spiking neuron structure is composed of four main components: 1) an adder which performs the weighted input summation, 2) a leaky integrator with a time-constant that is controlled by a "memory" parameter α, 3) a random generator which is used to implement the sigmoid activation function, and 4) a comparator that checks if the membrane potential reaches the threshold. The spiking neuron fires a pulse in case the weighted input summed by the adder exceeds the current random value (threshold). Synapses weights are kept in an internal neuron dedicated memory. The membrane potential $Y(t)$ is given by (2).

$$Y(t) = a\left(Y(t-1) + \sum_{j=1}^{N} I_j \cdot w_j\right) + Bias \tag{2}$$

where α is the neuron leakage parameter representing the oblivion factor, and the bias (theta) is the neuron offset value. The weighted inputs (axons) are accumulated directly into the adder in an iterative process, while adding the membrane potential from the previous iteration. The leakage rate is

controlled by the α parameter. The membrane potential is represented by an internal Y register. A pseudo-random generator RNG is used to generate the activation function. In case the accumulated value (including a bias) in Y exceeds the current rate of RNG the neuron fires a pulse in its output.

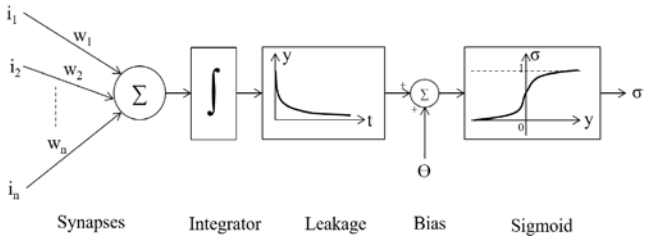


Fig. 3. The SN mathematical model.

## IV. The Neuron Specification and Parameters

The innovative neuron is configurable, and its operation is controlled by several parameters, allowing to control and modify the neuron behavior. The neuron model provides seven different leak modes and three activation functions with dynamic threshold setting. The synapses weights for the neuron cells inputs, as well as the neuron parameters (α and $\theta$) are kept in an internal dedicated memory unit. A unique neuron leaky parameter α represents the oblivion factor which enables controlling the rate of "remembering" or "forgetting" of previous states of the neuron. Therefore, by changing α different behaviors of the neural cell over time may be obtained. The incoming signals can be tuned by two parameters which define the signal gain and delay. Thus, the neuron effectively supports processing of serial input data and can be efficiently applied to voice applications. The binary values the weighted inputs are accumulated directly into the neuron kernel with no need for hardware multipliers implementation. Moreover, the multiplication of the accumulated weighted inputs by α (Eq. 2) may be performed using only shift and subtract operations. Table 4 shows the neuron reconfigurable parameters. These parameters may be adjusted to control the neuron behavior and accurately replicate the behaviors of a biological neuron model.

### A. The Neuron Parameters

*1) Neuron Leakage Factor:* The Leakage factor (LF) represents the integrator leakage rate and actually effects the neuron gain and delay. The LF parameter defines the constant α which multiplies the weighted inputs sum as indicated by Eq. (2). Since the multiplication is performed using shift operations LF is actually the number of right shifts that are applied to the value stored in the Y register that represents the membrane potential.

*2) Neuron Leakage Period:* The leakage period (LP) defines the rate of integrator leakage operation and effects the neuron gain and delay. A value of zero represents a full rate, meaning that a leakage operation is executed every pulse cycle. A value of N defines a leakage operation once every N+1 pulse cycle.

*3) The Neuron Delay:* Each neuron cell incorporates a leaking integrator with a time constant that is controlled by a predefined α parameter. The neuron may postpone incoming signals according to an expected leakage time constant, i.e. a delay which is given by (3).

$$T_{PulseCycle} \cdot 2^{LF} \cdot (1+LP) \tag{3}$$

*4) The Neuron Gain:* The incoming pulses are accumulated in the neuron cell with respect to the expected gain. The expected gains for the Identity and Sigmoid activation function are given in Eq. (4).

$$Gain = \begin{cases} LF > 2: \ 2^{2 \cdot LF-3} \cdot (1+LP) \\ LF < 3: \ 2^{2 \cdot LF} \cdot (1+LP) \end{cases} \tag{4}$$

### B. The Neuron Activation Functions

The neuron activation function defines the average spike rate at the neuron output as a function of the integrator value and the bias. The proposed neuron supports common three different types of activation functions: Identity, Binary and Sigmoid.

*Identity:* This activation function is linear to the input and is based on a linearly distributed random number.

*Binary:* This activation function produces an output pulse if and only if the neuron integrator value is positive.

*Sigmoid:* This activation function is approximately a classical sigmoid function. It is based on the sum of eight linearly distributed random numbers. The probability density function of the sum of random numbers is approximately a Gaussian function [17].

Table 3 shows some internal neuron parameters which are transparent to the user.

TABLE III: The Neuron Variables

| Variables | Symbol | Format |
|---|---|---|
| Membrane potential | $Y_j$ (t) | Sign int |
| Time step | t | Unsign int |
| Input stream of spikes on $i^{th}$ axon | $M_i$ (t) | {0,1} |

TABLE IV: The Neuron Reconfigurable Parameters

| Neuron Parameters | Symbol | Format |
|---|---|---|
| Synapse value ($i^{th}$ axon, $j^{th}$ neuron) | $S_{ij}$ | {0,1} |
| Synaptic weight | $w_{ij}$ | Sign int |
| Leakage factor | LF | Unsigned int |
| Leakage period | LP | Unsigned int |
| Activation function | AF | {0-Identity,1-Binary,2-Sigmoid} |
| Leakage timer | LT | Unsigned int |
| Bias | $\theta$ | Signed int |
| Reset membrane potential | $V_0$ | Sign int |

### C. The Random Number Generator

Efficient implementation of a random number generator (RNG) is used to produce the three different activation functions. This hardware implementation of a unique RNG module avoids the common use of LUT for storing the appropriate function. Therefore, the RNG implementation saves memory space, and reduce memory accesses and power consumption. Moreover, a better approximation of the activation functions is achieved.

## V. The Neuron Operation Description

The neuron operation flow may be better understood from the pseudo code and the flow diagram as depicted in Fig. 4. The neuron operation is based on an iterative process. For each iteration the adder performs a weighted sum, which is provided to the leaky integrator. Then, the integrator output is added to a bias and compared against a random value that is generated by a random generator. Finally, an output spike is generated if the sum exceeds the current random value. The neuron cell "output level" may be defined linearly by the spike rate in its output.

***SN Operation Flow - Pseudocode***

```
m = Layer number
n = Number of neurons in layer (m-1)
SUM = Y(t-1)
For ii = 0: n
    If FF (ii) ==1   then
        Weight_ij = Memory_Unit (ii)
        SUM = SUM + Wieght_ij
    End If
End For
α = Memory_Unit(ii+1)
y(t)= α · SUM;
RNG=Activation function: Identity, Binary or Sigmoid
If Activation function=2 and SUM > 0
    Neuron fires a pulse
Else
    Neuron output=0
End If
If Activation function=1 or 3
    If RNG < SUM
        Neuron fires a pulse
    Else
        Neuron output=0
    End If
End If
```

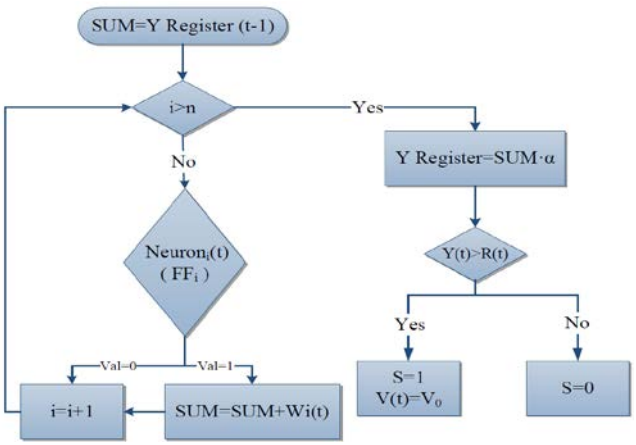


Fig. 4. The SN flow diagram.

## VI. Power and Area Analysis

The neuron cell occupies relatively small silicon area, runs at low frequency and therefore consumes ultra-low power. The hardware implementation of a single digital neuron cell requires only 700 ASIC gates (with two inputs) and an additional 33 gates per synapse. The IBM spiking neuron [7] is implemented using 1,272 ASIC gates and therefore consumes about twice silicon area and power compared to our proposed spiking neuron.

## VII. Conclusion

This paper introduces a new efficient digital spiking neuron model. The neuron is efficiently implemented in hardware and is well appropriate for ultra-low power applications. The proposed neuron model is configurable and has designated architecture for processing streams of continuous serial data, mainly audio and video. Enhanced performance and fast serial data processing flow is achieved by controlling the neuron parameters. Simulations of the proposed digital spiking neuron also demonstrate its ability to accurately replicate the behaviors of a biological neuron model.

## References


[1] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Heidelberg, Germany: Springer, 2013.

[2] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing*, vol. 74, no. 1-3, pp. 239-255, 2010.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.

[4] P. A. Merolla, *et al*., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668-673, Aug. 2014.

[5] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54-66, 2015.

[6] S. K. Esser, *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, vol. 113, no. 41, pp. 11441-11446, 2016.

[7] S. Dytckov and M. Daneshtalab. (2016). Computing with hardware neurons: Spiking or classical? Perspectives of applied spiking neural networks from the hardware side. [Online]. Available: http://arxiv.org/abs/1602.02009

[8] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker system architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454-2467, Dec. 2013.

[9] F. Akopyan, *et al.*, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537-1557, Oct. 2015.

[10] W. Zaremba, I. Sutskever, and O. Vinyals. (Sep. 2014). Recurrent neural network regularization. [Online]. Available: http://arxiv.org/abs/1409.2329

[11] N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biological Cybernetics*, vol. 95, no. 1, pp. 1-19, Jun. 2006.

[12] DSPG, "Neural Cell and A Neural Network," U. S. Patent 8986-USP, Dec. 14, 2017.

[13] K. Funahashi and Y. Nakamura, "Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks," *Neural Networks*, Vol. 6, pp. 801-806, 1993.

[14] A. Joubert, B. Belhadj, O. Temam, and R. Heliot, "Hardware spiking neurons design: Analog or digital?" in *Proc. of International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1-7.

[15] B. V. Benjamin, *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699-716, May 2014.

[16] F. Conti and L. Benini, "A ultra-low-energy convolution engine for fast brain-inspired vision in multicore clusters," in *Proc. of 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 683-688.

[17] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *IEEE Transactions on Computers*, pp. 31-44, Mar. 1996.

**Moshe Bensimon** is an M.Sc. student at the Department of Electrical and Computer Engineering, Ben-Gurion University. He holds M.B.A and B.Sc. degrees in electrical and computer engineering from the Ben-Gurion University of the Negev, Israel. His current research interests are machine learning and performance enhancement of computer architecture.

**Shlomo Greenberg** received his B.Sc., M.Sc. (with honors), and Ph.D. degrees in electrical and computer engineering from the Ben-Gurion University, Beer-Sheva, Israel, in 1976, 1984, and 1997, respectively. He is currently a staff member at the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel. His primary research interests are computer architecture, wireless communication, image and digital signal processing, computer vision and VLSI low power design.

**Yehuda Ben-Shimo** is a senior lecturer at the Department of Communication Systems Engineering, Ben-Gurion University. He holds Ph.D. (with honors), M.Sc. (with honors) and B.Sc. degrees in electrical and computer engineering from the Ben-Gurion University of the Negev, Israel. His main areas of interest are computer networks, design and analysis and performance evaluation of communication protocols and networks.

**Moshe Haiut** is a senior engineer at D.S.P Group Ltd. Herzelia, Israel. He holds M.Sc (with honors) and B.Sc. (with honors) degrees in electrical engineering from Tel-Aviv University and the Technion, Israel. His main areas of interest are communication, video & audio digital signal processing, digital H/W architecture, and neural networks.