Spiking Neuron Model for Embedded Systems

Vladimir Lutkovski
Faculty of RadioPhysics and Computer Technologies
Belarusian State University
Minsk, Belarus
e-mail: lutkovski@bsu.by

Dzianis Sarnatski
Faculty of RadioPhysics and Computer Technologies
Belarusian State University
Minsk, Belarus
e-mail: denisiussarnatski@gmail.com

Serafim Yablonski
Faculty of RadioPhysics and Computer Technologies
Belarusian State University
Minsk, Belarus
e-mail: serafimyablonsky2004@gmail.com

Abstract — *Spiking neural networks (SNN) are used in robotics, particularly on the boards of autonomous vehicles, so the issues related to the hardware implementation of spiking neurons and SNNs is hotly discussed. Significant attention is devoted to the energy efficiency of the models in use. In the frame of the presented project, well-established neuron models have been investigated. As the result the spikes counting model (SCM) enabling real-time operation and attaining high energy efficiency have been developed. The implementation of the developed model in microcontrollers MSP430 family is achieved without the need of floating-point operations (FPO). Moreover, we analyze the issue of transferring and implementing the spikes counting model using alternative platforms.*

Keywords — embedded system, microcontroller, MSP430. spiking neuron, spikes counting, hardware implementation, energy efficiency.

I. Introduction

The first successful attempt of modeling and understanding of human brain processes was reported in 1943. It resulted in creation of simple but quite effective model called “formal neuron” [1]. Later in 1949 Donald Hebb discovered details of self-tuning of biological neural networks [2]. A decade later, in 1957–1958, the modeling of a basic neuronal activity was implemented in the device called perceptron by F. Rosenblatt [3].

In the second generation of artificial neural networks (ANNs), also known as deep learning networks (DNN) more complicated models of neurons were used. DNNs are extremely useful instrument in a variety of human activity – from entertainment and service to industry and medicine. Typical applications of ANN are weakly-formalized tasks, such as pattern recognition, images and sound analysis, etc. [4].

Models of spiking neurons inherited yet more details of biological prototype compared to the previous models. Spike Neural Networks (SNN) are usually regarded as the third-generation neural networks. Among the actual applications of SNN are signals preprocessing, and data encoding, speech and images recognition.

Since last years the topic of hardware implementation of ANN, both classical DNN and SNN, is hotly discussed along with machine learning (ML) applications for embedded systems [5–12]. Significant attention is devoted to the energy efficiency of the used neuron models. Actually, the motivation of this study is driven not only by extremely fast-growing quantity of sources concerning the modeling and hardware implementation of ANN, SNN and ML in general. We intended to design a simple SNN models and integrate it into real embedded systems.

The presented paper is arranged as follows. Firstly, the related works are analyzed. In the contest of the discussed problem we focused our attention on the data representation in SNNs and hardware implementation of the low cost and energy efficient models of spiking neurons. The structure and hardware implementation of the original spikes counting model (SCM) is introduced in the third section. In the forth section the hardware implementation of the developed SCM into the real embedded system based on MSP430 microcontrollers is presented. Finally, possible applications of the model are considered and the author’s contribution in the topic is summarized.

II. Related works

The last decades demonstrated a fast growth of ANN theory and even faster improvement of ANN practice mostly due to new architectures as well as the ideas of deep learning [4]. Unfortunately, current models of DNN such as Convolution Neural Networks (CNN) require huge resources in terms of data sets, high computational costs and energy consumption. Graphics Processing Units (GPUs) allow faster computations but the problem of high energy consumption still remains.

On the one hand, many techniques have been developed for architectures reducing and parameters optimizing such as pruning in order to reduce the ANN complexity and a lot of works is devoted the reduction of power consumption in traditional DNN. On the other hand, development of hardware implementations of SNN is one of the promising ways for the problem solving due to the better modeling of biological neural networks in terms of energy efficiency and abilities for online learning. The fundamental issue in this way is the data representation: accumulated weighted sum of incoming signals is scalar in the ANNs of first and second generations while it is spiking in SNN and biological neural networks (BNNs) at the neuron’s membrane [8], (table I).

TABLE I. Properties of biological neural networks, ANNs, and SNNs (according to [8])

Properties	BNNse	ANNs	SNNs
Data Representation	Spikes	Scalars	Spikes
Training	Dynamics of Synapses	Gradient Learning	Under discussion
Platform	Brain	VLSI	SpecialVLSI

Many attempts have been made for modeling and hardware implementation of spiking neurons and SNNs [6–11]. Computational complexity of the model is usually evaluated in numbers of variables and floating points operations (FPO) required for the modeling. Usually one needs from 5 to 1200 FPO in order to simulate one spiking neuron. Parameters of selected models are listed in table II.

Among the simplest models of spiking neuron, Izhhikevich’s model is attractive for hardware implantation and applications in signal processing systems [9–12]. For example, it was applied for encoding the temporal radar signal into spikes for the radar interference detection [9]. Parameters of the selected spiking neurons models (according to [6] and [13]) are presented in the table II.

TABLE II. PARAMETERS OF SPIKING NEURONS SELECTED MODELS (ACCORDING TO [6,13])

Model	Number of FPO	Number of Variables	Complexity
Integrate-and-Fire	5	1	Very Low
Resonate-and-Fire	10	2	Very Low
Izhhikevich (2003)	13	2	Very Low
Spike Response	50	1	Very Low
Hodkin-Huxley	1200	1	Very High

Tuning of SNN differs from learning of first and second generations ANN in some details. It was shown that input patterns can be encoded in the synaptic weights by local Hebbian delay-learning of spiking neurons where, after learning, the firing time of an output neuron reflects the distance of the evaluated pattern [12]. In fact, the such algorithms as Hebbian Learning, Genetic Algorithms may be used in usual manner for SNN learning [12,13].

Recently the original spike-based learning rule for deep SNN training was introduced. Differing from other spike-based learning rules, the spike count of each neuron is used as the surrogate for gradient backpropagation requiring much less memory and computations [15].

Spiking neurons could be implemented in hardware on the basis of special neuromorphic VLSI [8] or microcontroller unit (uCU). For example, a spiking neuron models for neuroscience teaching were implemented in ATMega328 uCU on the basis of Arduino platform [11].

III. THE SPIKES COUNTING MODEL

Neurons in BNN perform a wide range of functions – from simple data encoding to convolution and feature extraction [6, 8]. Here we intended to develop a simple resonant model for data encoding. A lot of such neuron models are based on the differential equation presented in the next complex form:

$$x' = (-d + iw_0)\, x + I,$$

where $(x,\ x')$ – neuron state space vector (x' is the derivative), d– decay constant, $\omega_0 = 2\,\pi\, f_o$ – neuron's resonant frequency, I – input signal.

Searching the numerical algorithm to integrate the above equation, we try to minimize the number of FPO. In fact the most of microcontrollers has counters and timers on the chip and it allows us to perform summing of spikes without using the central processing unit (CPU).

The developed spikes counting model (SCM) was simulated in SciLab and Matlab/Similink environments. The structure of developed model is presented in Fig. 1. Parameters of State-Space and Timer blocks are tuned depending on the input signal under processing. Functions of the other blocks are clear from subscriptions. There are two outputs: encoded signal from the counter (Sum of Elements) and single spike for the next synaptic connection.

The designed SCM has some features of resonate-and-fire neuron [9] and Izhhikevich's model [10], nevertheless the implementation of it on the board of microcontroller is achieved without the need of floating-point operations (FPO). It is achieved using spikes counting at the successive time intervals determined by the Timer as it is shown at the Fig. 1.

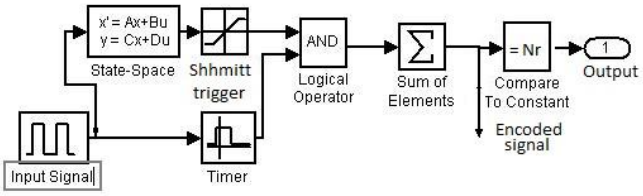

Fig. 1. The structure of SCM

The presented model was simulated and tested both on the basis of CMOS CD400 serial logical elements and ultra-low power microcontrollers MSP430 (presented in the next section).

IV. HARDWARE IMPLEMENTATION OF THE SPIKING NEURON

The developed model was firstly implemented on the basis of CMOS CD400 serial logical elements (Fig.2). In this case, the analyzed signal is connected to the analogue signal preprocessing unit consisting of a selective amplifier and logical AND gate with Schmitt's trigger function (CD4083A). The Timer (NE555P) determines the duration of the time window allowing pulses from the signal preprocessing unit pass to the integrated counter (CD4520). The LED display is used for signals selection and control.

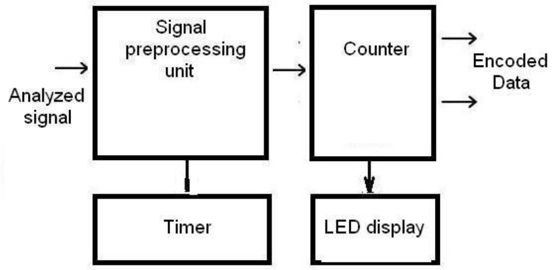

Fig. 2. The hardware implementation of spikes counting neuron model

The microcontroller based experimental setup for the model testing is presented in Fig. 3. Functions of timing and spikes counting are performed in uCU (MSP430G2553). The signal preprocessing unit is the same as above.

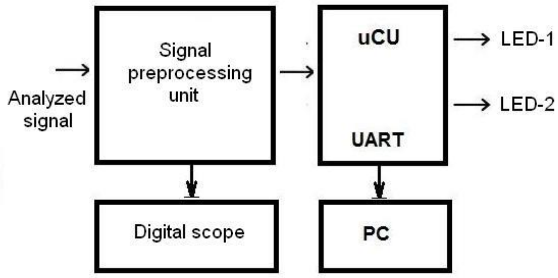

Fig. 3. MSP430G2553 based experimental seup

The developed model of SCM could be easily implemented in the MSP430 family microcontrollers based on the performed modeling.

Code Composer Studio was used as the IDE for MSP430G2553 embedded code programming and debugging. The flowchart of an algorithm for spikes counting is presented below in Fig. 4. Different low power modes (LPMs) were used for the better energy saving.

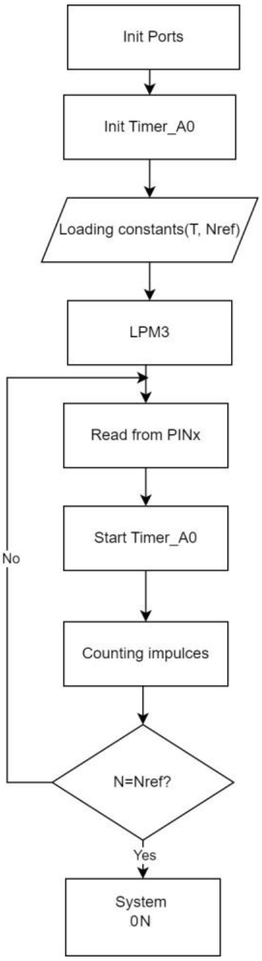

Fig. 4. Flowchart for CSM simulation in MSP430

On switching the power, RESET signal is formed, ports and timers are initiated, constants T (time window duration) and Nref (expected quantity of spikes within the time window T) depending on the input signal under processing are loaded, microcontroller LPM3 is switched. On arriving any signal from the signal preprocessing unit to pin P1.4 of MSP430 the spikes pulses are counted. Finally, we have the quantity of counted spikes within the selected time interval T or output signal if the quantity of counted spikes is equal to the preinstalled Nref.

Besides the educational purpose of demonstrating the SCM model, the algorithm could be used in practice for voice activation (i.e. switching on) any embedded system. We consider the switching on an embedded system on arriving selected voice command as the useful application of developed SCM model. In another words, the main embedded system may be in the energy saving standby mode, while the SCM model in MSP430 is active any time performing the analysis of input audio signals. A signal for the main system activation is generated if the quantity of counted pulses is in the range corresponding the selected voice command. The calculated parameters of the neuron were loaded into the uCU memory on performing signals analysis for the chosen limited set of voice commands.

Sounds NUL, RAZ, DWA, TRI (in Russian) were recorded with the help of digital scope and used for this procedure testing. Parameters of SCM were tuned for sound NUL (Fig. 5 up), selected for system switching in the next way: Nref = 9 for the time window T=0.1 s. It was shown that the model recognized the NUL and there was no reaction on the sound TRI (Fig. 5 bottom) and the other sounds.

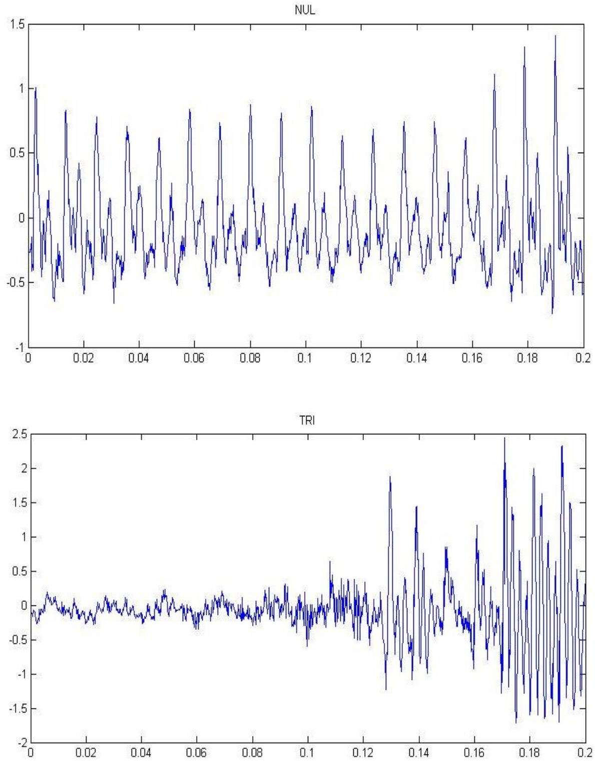

Fig. 5. Tested signals: NUL (up) and TRI (bottom)

There are different strategies for using the SCM neuron in a voice command analyzing:

i) counting the pulses from the signal preprocessing unit within selected time window using onboard Timer_A0;

ii) using the onboard analog-to-digital converter (ADC), where interrupt from ADC will determine the value of the input source signal any time.

Strategies i) and ii) are effective in terms of energy saving, although the first method will be preferable for greater efficiency as the LPM3 mode in MSP430G2553 requires only 1.8 V power supply. In practice it will be easier to parallelize the task with an ADC by sending a signal to other pins, changing only the program and not affecting the modification of the hardware (especially since some MCUs, for example MSP430G2553, implement fast continuous reading). This allows us to increase accuracy paying a little cost.

The loss in power consumption may be not significant as the MSP430 microcontrollers have their own internal oscillator for the ADC. It means that it does not have any of the clock signals, which also allows the MCU to be used in one of the low power modes.

In fact, the model could be incorporated into any microcontroller containing counters or ADC on the board. Nevertheless, there are strong arguments for using microcontrollers from the MSP430 family: it is possible to achieve the better energy consumption parameter using Energy Trace Technology while debugging embedded software with the help of the Code Composer Studio IDE.

Scalability is one of the remarkable properties of the MSP430 family. It allows us developing and testing the prototype based on the simple and cheap microcontroller and later migrate with tested prototype to the more powerful boards in terms of the speed and quantity of timers on the uCU crystal (Table III).

TABLE III. FEATURES OF THE SELECTED MICROCONTROLLERS

uCU	Features		
	Timers on board	Max Speed	Energy consumption
MSP430G2211	1	8 Mips	0.5μA*
MSP430G2553	2	16 Mips	0.5 μA *
MSP430FR2311	3	16 Mips	1.0 μA *
ATMega328P	3	16 Mips	1.0 μA *
STM32F030F4	11	48 Mips	2.6 μA*
Raspberry Pi 4	2	1500Mips	0.5 A

*|) Energy consumption – in Low Power Mode (LPM)

V. DISCUSSION AND CONCLUSION

In the paper the spikes counting model of neuron has been designed and tested. It could be implemented into embedded systems using a microcontroller or CMOS IC logical elements CD4000 (analog of K561) serial.

The implementation of the proposed model in microcontrollers MSP430 family is achieved in the real time without the need for floating-point CPU operations.

The issue of transferring and implementing the discussed model onto alternative platforms was also considered. In fact, any microcontroller may be used if the energy efficiency is no importance, but we consider the proposed approach for microcontroller-based simulation of spiking neurons as preferable. The remarkable features of the developed models are the low cost and the lower energy consumption compared to the other hardware implementations of spiking neurons. The energy consumption of the SCM model in MSP430 microcontroller is twice less in comparison to the hardware model based on ATMega328 uCU [11].

The choice of components and embedded system's structure was made with the intention of being sufficient to demonstrate the principles of spiking neurons modeling. Evidently the quantity of recognized voice commands for every uCU is limited by the quantity of timers-counters on the crystal of used uCU. Nevertheless, it is possible to design the advanced embedded system for recognizing wider set of voice commands adding peripherical SCM such as presented in Fig. 2.

The area of possible applications of spiking neurons in general and the developed model in private is much wider than considered above. For example, it is possible to detect the arrhythmia of heart beating signals, perform Fourie transform and the other functions [8, 15]. It also may be applied for sensors connecting and biomedical signals analysis. The presented project is being in the state of integrating the developed models, devices and software into the educational process. It was carried out by students at the Faculty of Radio Physics and Computer Technologies during the courses "Simulation and Statistical Modeling" and "Neural Networks and Deep Learning."

ACKNOWLEDGMENT

The authors gratefully acknowledge the technical and financial support of student’s projects from Andrew Popleteev and Valery Shnitko.

REFERENCES

[1] A logical calculus of the ideas immanent in nervous activity/ W.S. McCulloch, W. H. Pitts // Bulletin of the mathematical Biology. 1990. V. 52. No ½. PP. 99–115.

[2] Hebb, D. Organization of behavior. New York: Wiley, 1949.

[3] Rosenblatt F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.

[4] Golovko, V. Neural Networks Technologies for Data Processing/ V. A. Golovko, V.V. Krasnoproshin. BSU. 2017. 263 p. (In Russian: Головко В.А., Краснопрошин В.В. Нейросетевые технологии обработки данных: Минск.- БГУ. - 2017. –263 с.)

[5] Machine Learning for Microcontriller-Class Hardware: A Review/ Swapnil Sayan Saha et al.//IEEE Sensors Jornal. Vol 22. No 2. 15 November 2022. P.21362 – 21371.

[6] Spiking Neuron Models: A Review/ Ahmed A. Abusnaina et al// Int. Jornal of Digital Content Technology and Applications. Vol.8 No 3. June 2014. P.14 – 21.

[7] FPGA Implementation of Self-Organized Spiking Neural Network Controller for Mobile Robots / Fangzheng Xue,Wei Wang, Nan Li, and Yuchao Yang// Advances in Mechanical Engineering. 2014. P. 1–10. http://dx.doi.org/10.1155/2014/180620

[8] Spiking Neural Networks and Their Applications: A Review/ Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le· // Brain Sci. 2022 Jul; 12(7): 863. Published online 2022 Jun 30. doi: 10.3390/brainsci12070863.

[9] Resonate-and-Fire Neurons for Radar Interference Detection/ Daniel Auge, Etienne Muller. // ICONS '22: Proceedings of the International Conference on Neuromorphic Systems 2022July 2022 Article No.: 2 3. Pages 1–4.

[10] Izhikevich E.M.Simple Model of Spiking Neurons //IEEE Trans. on Neural Networks. 2001. Vol 14 No 6-7 P. 883–894.

[11] Spikeling: a low-cost hardware implementation of a spiking neuron for neuroscience teaching and outreach/ https://www.biorxiv.org/content/10.1101/327502v1.full

[12] Embedded neural controllers based on spiking neuron models/ Laszlo Bako and Sandor Tihamer Brassai //Pollac Periodica 2009. V.3. No4. P. 143 – 154.

[13] Evangelos Stromatias. Developing a supervised training algorithm for limited precision feed-forward spiking neural networks. Dissertation. University of Liverpool . 2011. 107 p.

[14] Deep Spiking Neural Network with Spike Count based Learning Rule/ Jibin Wu et al// arXiv: 1902.05705v1 [cs.NE] 15 Feb. 2019.

[15] Time-coded Spiking Fourier Transform in Neuromorphic Hardware/ Javier López-Randulfe, Nico Reeb, Negin Karimi, Chen Liu, Hector A. Gonzalez, Robin Dietrich, Bernhard Vogginger, Christian Mayr, Alois Knoll// arXiv: 2202.12650v2 [cs.NE] 31 Mar. 2022. 17 p.