



HAL
open science

Real-time pattern recognition implementation on FPGA in multi-SNNs

Yang Xia, Timothée Levi, Takashi Kohno

► **To cite this version:**

Yang Xia, Timothée Levi, Takashi Kohno. Real-time pattern recognition implementation on FPGA in multi-SNNs. ICAROB, Jan 2020, Beppu, Japan. hal-02484022

HAL Id: hal-02484022

<https://hal.science/hal-02484022>

Submitted on 19 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Real-time pattern recognition implementation on FPGA in multi-SNNs

Yang Xia*

*Graduate School of Engineering, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

Timothée Levi

*LIMMS/CRNS-IIS, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

Takashi Kohno

*IIS, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

E-mail: xiayang@sat.t.u-tokyo.ac.jp,

Abstract

By mimicking or being inspired by the nervous system, Neuromorphic systems are designed to realize robust and power-efficient information processing by highly parallel architecture. Spike timing dependent plasticity (STDP) is a common method for training Spiking Neural Networks (SNNs) for pattern recognition. Here, we present a real-time STDP implementation on FPGA in SNN using digital spiking silicon neuron (DSSN) model. Equipped with Ethernet Interface, FPGA allows online configuration as well as data input and output all in real-time. We show that this STDP implementation can achieve pattern recognition task and the connection between multi-SNNs enlarge the scale of networks and application.

Keywords: SNN, STDP, DSSN, FPGA, Ethernet

1. Introduction

Neuromorphic systems are designed by mimicking or being inspired by the nervous system, which aims to realize robust, autonomous, and power-efficient information. There are three common methods to realize the neuromorphic circuits, which are software¹, analog hardware^{2,3} and digital hardware^{4,5}. Software-based cannot achieve real-time processing. Compared to analog circuits, digital implementations generally consume higher power but are more scalable because they are less susceptible to noise and fabrication mismatch. Another

advantages of FPGA devices are configurability, portability, and low-cost.

Silicon neuronal network (SNN) is a neuromorphic circuit that reproduces the electrophysiological activities in the nervous system focusing on the spiking dynamics in the neuronal cells and their transmission via the synapses. Their application includes bio-inspired information processing such as pattern recognition^{6,7} and associative memory⁸ as well as neuro-prosthetic devices⁹

Digital SNNs are expected to achieve a very large-scale network comparable to the human brain in the future

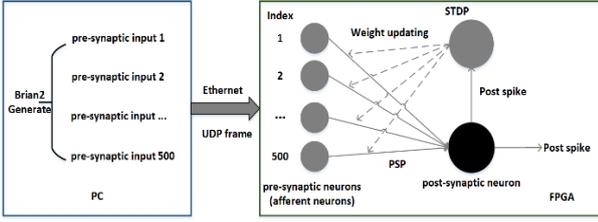


Fig. 1. Overall Architecture of STDP learning.

exploiting the scalability of the digital circuits. In these 5 years, several very large-scale SNNs with one million neurons were developed¹¹.

Spike-timing-dependent plasticity (STDP) is a well-known rule for updating the synaptic efficacy in SNNs, which uses only local information. Many biological experiments found evidence for STDP process in the synapses^{12,13}.

In this paper, we report an implementation of digital SNN with online STDP learning on FPGA. The model and implementation of our SNN are explained in the next section and section 3, respectively. Then results and conclusion follows.

2. Architecture of the network model

2.1. DSSN model

Neuronal models need to be chosen taking into account the balance between the reproducibility of neuronal activity and computational efficiency. Integrate-and-fire (I&F)-based models are able to be implemented by compact hardware, but they lack reproducibility of complex neuronal dynamics. Ionic-conductance models have high-ability of reproducing neuronal activities, but cost massive computational resources. The DSSN model is a qualitative neural model¹⁴, which was designed for efficient implementation in digital circuits. The simplest version of DSSN model supports the Class I and II cells in Hodgkin's Classification¹⁵. The differential equations of DSSN model are as follows.

$$\frac{dv}{dt} = \frac{\phi}{\tau} (f(v) - n + I_0 + I_{stm}), \quad (1)$$

$$\frac{dn}{dt} = \frac{1}{\tau} (g(v) - n), \quad (2)$$

$$f(v) = \begin{cases} a_{fn}(v - b_{fn})^2 + c_{fn}(v < 0) \\ a_{fp}(v - b_{fp})^2 + c_{fp}(v \geq 0) \end{cases}, \quad (3)$$

$$g(v) = \begin{cases} a_{gn}(v - b_{gn})^2 + c_{gn}(v < r_g) \\ a_{gn}(v - b_{gn})^2 + c_{gn}(v \geq r_g) \end{cases}. \quad (4)$$

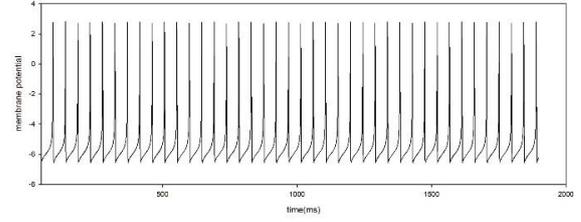


Fig. 2. Stimulation of a DSSN neuron by a constant input current (0.308).

Here v represents the membrane potential, n is a variable that reflects the activities of hyperpolarizing ionic channels. a_{xy} , b_{xy} , and c_{xy} ($x = f, g$ and $y = n, p$) are parameters. The Parameter I_0 is a bias constant and I_{stm} represents the input stimulus.

The only nonlinearity in the DSSN model is a quadratic function. Thus, solving this model using Euler's method needs one multiplication operation per step if the parameters are carefully selected^{8,16}. As multiplication operation requires relatively large resources in digital circuits, the DSSN model is suitable for digital silicon neuronal networks.

2.2. Synaptic model

Postsynaptic current (PSC) is a current inserted to the postsynaptic cell that induces temporal change in its membrane potential. The PSCs generated by a pulse stimulus to chemical synapses are able to be modeled by the alpha function with double-exponential generalization¹⁷. In our network model, the PSP model was simplified as follows.

$$\frac{dx}{dt} = \frac{-x}{\tau}. \quad (5)$$

Here x represents PSC generated by a chemical synaptic transmission. It is reset to w , the connection strength of the synapse, when the membrane potential of the presynaptic neuron exceeds 0. The initial value of w is 6×10^{-10} . The PSCs in the synaptic connections to a postsynaptic are summed up as follows.

$$I_{stm} = \sum x_i. \quad (6)$$

Here i represents index of presynaptic neurons.

2.3. STDP algorithm and architecture

Synaptic plasticity means the magnitude of synaptic efficacy to change in response to the activities of pre and postsynaptic neurons.

The STDP is a biological rule that adjusts the strength of synaptic connections (w) based on the relative timing of the spikes in a pre and postsynaptic neurons. Recently it

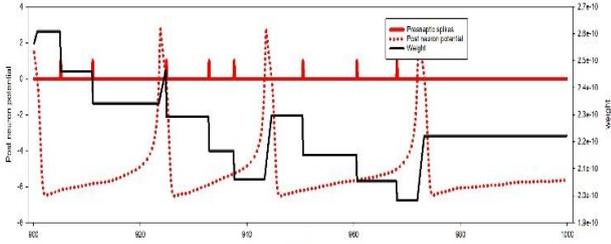


Fig.3. Waveform of STDP method implementation.

has been shown how STDP rule play a key role by detecting repeating patterns and generating selective response to them¹⁸.

The STDP rule is a most common form of learning rules used in SNNs. Here is the standard exponential STDP equations.

$$w_{new} = w + \Delta w, \quad (7)$$

$$\Delta w = \begin{cases} a^+ \cdot \exp\left(\frac{t_j - t_i}{\tau^+}\right), & \text{if } t_j \leq t_i \text{ (LTP)} \\ -a^- \cdot \exp\left(-\frac{t_j - t_i}{\tau^-}\right), & \text{if } t_j > t_i \text{ (LTD)} \end{cases} \quad (8)$$

Here Δw is the modification of the synaptic weight. When a post-synaptic spike arises after a pre-synaptic spike ($t_j \leq t_i$), the connection is reinforced (long-term potentiation (LTP), $\Delta w > 0$), whereas in the opposite case it is weakened (long-term depression (LTD)).

3. Implementation

The overall architecture of our SNN is shown in Fig.1. The DSSN, STDP, and PSC blocks were implemented on a FPGA chip. In this chip, single postsynaptic neuron is connected with 500 input afferent neurons. Stimulus spike trains were generated by Brian2 on PC and sent to FPGA via Ethernet connection which are explained in 3.4 and 3.5.

3.1. Implementation of DSSN

The DSSN model's differential equations are solved by Euler's method and the value of dt is 0.1s. The solver circuit has time division multiplexing (TDM) and pipelined architecture.

3.2. Implementation of PSC

As introduced in 2.2, we use a simplified exponential decay to approximate the PSCs. The connection between a postsynaptic neuron and the 500 pre-synaptic neurons was calculated efficiently by using TDM and two-stage pipeline as in the DSSN solver circuit. Operations for the 500 sets of PSCs were finished in 1000 clock cycles, by

using single multiplier and single adder. The time step for the summation of PSCs is within 0.1ms if the system clock is faster than 100MHz.

3.3. Implementation of STDP learning

Based on the STDP rule, each presynaptic (postsynaptic) spike, induces an LTD (LTP) where the synaptic weight is updated according to Eq. (7). An example waveform of PSC is shown in Fig.3, where the solid and dotted red curve represent the presynaptic and postsynaptic spikes, as well as the black curve is the strength of synaptic connection. For simulating 500 afferents, 500 clock cycles were consumed. Calculation of exponential function and update of synaptic efficacy are executed by TDM.

3.4. Ethernet on FPGA

Ethernet is a computer networking technology commonly used in local area networks (LAN). Devices equipped with Ethernet interface are connectable to LAN as well as the internet by supporting common communication protocol, for instance, TCP/IP and UDP. For neuronal network implementation on FPGAs, transmission of the spikes is always an issue particularly when the number of neurons is large.

By implementing Ethernet interface and a full hardware protocol stack including IP, UDP and ARP protocol, our SNN on FPGA is capable of receiving input spike patterns from a PC by Ethernet connection and sending report frame back to the PC for monitoring the SNN working status at any time.

The configuration and parameter setting are also possible by the Ethernet connection, which contribute to take advantage of FPGA in flexibility and portability.

4. Pattern recognition

Spatiotemporal spike pattern recognition tasks proposed by Masquelier¹⁹ were performed on our network. The stimulus pattern required in them were generated by superimposing a spatiotemporal spike pattern at many time points on a background random spike pattern. The former spike patterns (ones to be detected) were generated by PoissonGroup Function in Brian2 (a Python library) on PCs. The background random spike patterns were generated using the same library. The stimulus patterns were stored in files and sent from the PC to the FPGA via the Ethernet connection. An example of learning process is shown in Fig.6. The postsynaptic

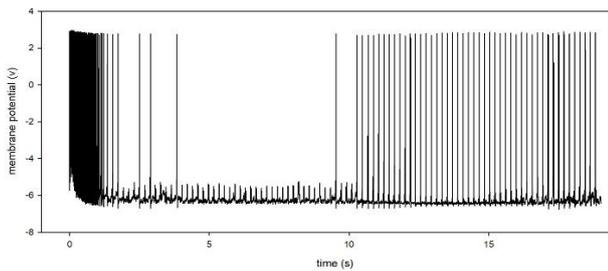


Fig. 4. Waveform of STDP learning process. The learning is finished after around 11s.

neuron started to detect the superimposed patterns at 11s. Our circuit performed this task in real-time. FPGA's system clock was 100MHz.

5. Conclusion

In this paper, a SNN on an FPGA with STDP learning capability was reported. Stimulus spikes were conveyed in real-time to the FPGA chip via Ethernet connection. It was proved that our SNN is capable of spatiotemporal spike pattern recognition in real-time.

In this work, the input patterns were generated on a PC. Perspectives are to generate them in real-time with another SNN. It is also planned to develop an FPGA-FPGA connection bus to expand the scalability of the FPGA-based SNNs, as the scale of the network on an FPGA chip is limited by the amount of on-chip memory used for storing the strength of synaptic connections. An SNN platform with high scalability is advantageous compared to software simulation from the viewpoint of real-time operation and power consumption.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15KK0003.

References

1. M.L. Hines and N.T. Carnevale, *NEURON: a tool for neuroscientists*, *The Neuroscientist* 7, 123-135 (2001).
2. A. Natarajan and J. Hasler, *Hodgkin-Huxley Neuron and FPAA Dynamics*, *IEEE Trans Biomed Circuits Systems*, 12(4), 918-926 (2018).
3. F. Grassia, L. Buhry, T. Levi, J. Tomas, A. Destexhe and S. Saïghi, *Tunable neuromimetic integrated system for emulating cortical neuron models*, *Frontiers in Neurosciences*, 5(134) (2011).
4. M. Ambroise, T. Levi, S. Joucla, B. Yvert and S. Saïghi, *Real-time biomimetic Central Pattern Generators in an FPGA for hybrid experiments*, *Frontiers in Neurosciences*, 7(215) (2013).
5. Grassia F, Kohno T, Levi T, *Digital hardware implementation of a stochastic two-dimensional neuron model*, *Journal of Physiology Paris*, 110(4), 409-416 (2016).
6. N. Qiao et al., *A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses*, *Frontiers in Neuroscience*, 9(141) (2015).
7. P. Merolla et al., *A million spiking-neuron integrated circuit with a scalable communication network and interface*, *Science*, 345(6197), 668-673 (2014).
8. Li J, Katori Y, Kohno T. *An FPGA-Based Silicon Neuronal Network with Selectable Excitability Silicon Neurons*. *Front Neurosci*, 6(183) (2012).
9. F. Broccard, S. Joshi, G. Cauwenbergs, *Neuromorphic neural interfaces: from neurophysiological inspiration to biohybrid coupling with nervous systems*, *J. Neural Eng.*, 14(4):041002 (2017).
10. S. Buccelli et al., *A neuroprosthetic system to restore neuronal communication in modular networks*, *iScience*, 19, 402-414 (2019).
11. M. Davies et al., *Loihi: a neuromorphic manycore processor with on-chip learning*, *IEEE Micro*, 38(1), 82-99 (2018).
12. Markram, H., Lübke, J., Frotscher, M., and Sakmann, B., *Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps*. *Science* 275(5297), 213-215 (1997).
13. Jacob, V., Brasier, D. J., Erchova, I., Feldman, D., and Shulz, D. E., *Spike-timing-dependent synaptic depression in the in vivo barrel cortex of the rat*. *J. Neurosci*, 27(6), 1271-1284 (2007).
14. Kohno T., Aihara K., *Digital spiking silicon neuron: concept and behaviors in GJ-coupled network*, in *Proceedings of International Symposium on Artificial Life and Robotics*, OS-6 (2007).
15. Hodgkin A. L., *The local electric changes associated with repetitive action in a non-medullated axon*. *J. Physiol*, 107(2), 165-181 (1948).
16. Nanami T, Kohno T. *Simple Cortical and Thalamic Neuron Models for Digital Arithmetic Circuit Implementation*. *Front Neurosci*, 10(181) (2016).
17. Amari, S., *The Handbook of Brain Theory and Neural Networks*, MIT press (1995).
18. Grassia F, Levi T, Doukkali E, Kohno T, *Spike pattern recognition using artificial neuron and spike-timing-dependent plasticity implemented on a multi-core embedded platform*, *Journal on Artificial Life and Robotics*, Springer Nature, 23(2), 200-204 (2018).
19. Masquelier T, Guyonneau R, Thorpe SJ, *Competitive STDP-based spike pattern learning*. *Neural computation*, 21(5), 1259-1276 (2009).