Implementation of Hardware Model for Spiking Neural Network

Jungmin Choi[1], Minwook Ahn[2], and Jong Tae Kim[1]
[1]Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea
[2]DMC R&D Center, Samsung Electronics, Suwon, Korea

Abstract – *The izhikevich neuron model is well known for mimicking almost all dynamics of the biological neurons like Hodgkin-Huxley neuron models with much less hardware resources. Despite its versatility and biological plausibility, izhikevich neuron model is still not suited for a large scale neural network simulation due to its complexity compared to the simpler neuron models like integrate-and-fire model. In this paper, we implement a Spiking Neural Network (SNN) of the silicon neurons based on the izhikevich neuron model in order to show that it is feasible to simulate a large scale SNN. As a demonstration, we construct our system to simulate a sparse network of 1000 spiking neurons on Xilinx FPGA. During the simulation period (1000ms), the network exhibits a rhythmic activity in delta frequency range around 4Hz. This means that the proposed network can simulate a large scale SNN based on izhikevich neuron model for human cortical system.*

Keywords: Spiking neural network, SNN, Izhikevich model, Biophysical model

1 Introduction

Neural network, mimicking the function of human brain, is widely used for several key applications such as vision processing, speech recognition, and classification. In most studies, the neural network is designed on a large scale to improve the performance of those algorithms. In order to implement the large scale neural networks, there are some approaches. One of the most useful methods is using the neuron model emulation on GPU. This is successful approaches due to the powerful computation resources provided by GPU. However, most CE devices are adapting the approaches via a network-based server, because of its huge computational power. There are some attempts [1] [2] to implement a neural network by hardware in order to overcome such a drawback.

One of the most commonly used algorithms to simulate a large scale neural network and hardware acceleration is the Spiking Neural Network (SNN) models [2]. These models simulate neuronal behavior more similarly than the traditional neural networks. In many researches, these models have designed on FPGAs to examine the feasibility of hardware implementations about the models [3] [4] [5] [6].

In this paper, we implemented a SNN of the silicon neurons based on the izhikevich neuron model, which is known for resembling almost all biological neuron dynamics [7]. Our proposed architecture can simulate randomly-connected SNN network of thousands of spiking neurons with millons of synapses in real-time. The simulated time resolution is configured by Phase Locked Loop (PLL) logic and the parameters of each neuron are stored in on-chip memory.

We explore the feasibility of using FPGA for large scale simulations of SNN based on the izhikevich model. Our experiment shows that our silicon neural network has biological plausibility of spiking behavior and resembles some dynamics of human brains through hardware simulation.

Section 2 of this paper provides background of the SNN and characteristics of the individual spiking neuron models. Section 3 explains the proposed hardware architecture of SNN and their operations. Section 4 discusses the result of this work and analyzes the meaning of data while section 5 concludes this paper.

2 Background

SNN, the third generation of neural network, is characterized the high level of biological realism by using individual spikes [8]. Fig. 1 shows an overview of a spiking neural network. A set of neurons on the left side generates voltage spike or fire. Each spike of neurons goes to the synapse which is connected to the other neurons. Each synapse has a weight and the magnitude of the input spike is scaled with that weight. The scaled spikes are summed to provide the overall input current for the next neuron. After that, the neuron decides its next state using the input current and its current state. According to the changed state, the neuron determines whether or not the generation of voltage spike.

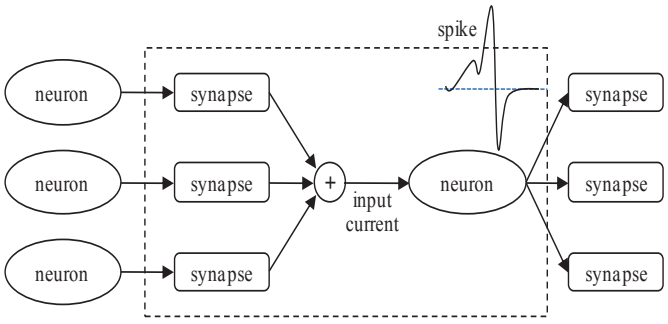

Fig. 1. Overview of a spiking neural network

The SNN include the concept of time into the operating models with spatial information, like real neurons do. The concept is that neurons in SNN do not fire every cycle, but it fires only when a membrane potential exceeds a certain value. When a neuron fires, it provides a signal to other neurons connected by synapses.

There are lots of neuron model in SNN. Several studies compare 11 spiking neuron models base on their biological plausibility and computational complexity [7]. Of these models, current studies of SNN models are generally using the Hodgkin-Huxley, leaky integrate-and-fire and izhikevich models [9].

2.1 Hodgkin-Huxley model

The Hodgkin-Huxley model [10] is one of the most important and biologically accurate models in SNN. It consist of four differential equations (eq. 1-4) and lots of parameters which describe membrane potential, activation of Na and K currents, and inactivation of Na current. The model can exhibit all kinds of spiking patterns and neural dynamics if the parameters are tuned.

The Hodgkin-Huxley model is the most biologically plausible, but one of the main problems is that it is extremely expensive for large scale implementations. Thus, one can use the model only to implement a small number of neurons.

$$\frac{dv}{dt} = (\frac{1}{C})\{I - g_k n^4 (v - E_k) - g_{Na} m^3 h (v - E_{Na}) - g_L (v - E_L)\} \quad (1)$$

$$\frac{dn}{dt} = (n_\infty(v) - n)/\tau_n(v) \quad (2)$$

$$\frac{dm}{dt} = (m_\infty(v) - m)/\tau_m(v) \quad (3)$$

$$\frac{dh}{dt} = (h_\infty(v) - h)/\tau_h(v) \quad (4)$$

2.2 Leaky Integrate-and-Fire (I&F) model

One of the most widely used and simplest models in SNN is the leaky I&F neuron model. It has only one differential equation (eq. 5) where *a, b, c* are parameters of the model, *I* is the neuron current, and *v* is the membrane potential of the neuron. When the *v* reaches its threshold value, the neuron generates spike. Then *v* is reset to value *c* according to the expression (6).

Leaky I&F model is incapable of producing rich spiking patterns or neural dynamics, because it has very simple equation and variables. Though the model is computationally effective and ease with that can be simulated or analyzed, it is unrealistic to implement a biologically plausible SNN.

$$\frac{dv}{dt} = I + a - bv \quad (5)$$

$$if \quad v \geq v_{thresh} \quad then \quad v \leftarrow c \quad (6)$$

2.3 Izhikevich model

The izhikevich model is described by differential equations of the form (Eq. 7-9)

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \quad (7)$$

$$\frac{du}{dt} = a(bv - u) \quad (8)$$

$$if \quad v \geq 30mV \quad then \quad v \leftarrow c,\ u \leftarrow u + d \quad (9)$$

where *a, b, c, d* are parameters of the model. The variable *I* is the neuron current, *v* is the membrane potential of the neuron and *u* is the membrane recovery factor which affects membrane reset. After the membrane potential reaches its threshold (*+30mV*), *v* and *u* variables are reset according to the expression (9).

In our system, we applied the Izhikevich model as a compromise for the biophysical similarity and computational power. The model is close to the Hodgkin-Huxley model in biological plausibility, but it is similar to the leaky I&F model in computational complexity. Table 1 shows that the excellent performance of the izhikevich model than the other models.

Spiking neuron model	Number of firing patterns	Resources
Hodgkin-Huxley	19	8320 (Spartan-3 xc3sd1800a 4-input LUT)
Hindmarsh-Rose	18	831 (Virtex-2 4-input LUT)
Izhikevich	21	195 (Virtex-5 xc5vlx330t LE)
Leaky Integrate-and-Fire	3	366 (Spartan xc3s1500 4-input LUT)
Quadratic Integrate-and-Fire	6	207 (Virtex-5 xc5vl330t 6-input LUT)

Table 1. The performance comparison of each spiking neuron models

3 Hardware Implementation

3.1 Different types of spiking neurons

The cortical neurons in human brain can be classified into several types according to the firing and bursting pattern [11]. All excitatory cortical neurons are divides into the following three classes : RS(Regular spiking) neuron is the most typical neurons in the cortex, IB(Intrinsically Bursting) neuron fires a stereotypical burst of spikes followed by repeated single spikes, and CH(Chattering) neuron can fire stereotypical bursts of nearly spaced spikes.

And all inhibitory cortical neurons are divided into the following two classes : FS(Fast Spiking) neuron can fire periodic trains of action potentials with extremely high frequency without any adaptation, and LTS(Low-Threshold Spiking) neuron can also fire high frequency trains of action potentials, but with a marked spike frequency adaptation.

In order to simulate accurately neuron dynamics in our system, we used these all kind of models by adjusting the parameters of izhikevich neuron model.

3.2 Structure

We construct our system to simulate a sparse network of 1000 spiking neurons with 1 million synaptic connections in real time (1*ms* resolution) using verilog on Xilinx ISE design suite. For efficient use of hardware resources, we design the actual logic so that only a single neuron is operated per 1 clock cycle. And we could obtain same results by the clock rate 1000 times faster. This system is shown in Fig. 2

Each neuron is randomly connected to 1000 other neurons with an excitatory-inhibitory ratio at 4:1. And we set each parameter (*a, b, c, d*) for modeling the rich spiking neuron structures (RS, IB, CH, FS, LTS) [11] described above are uniformly distributed per each time resolution. A RS model corresponds to c = -65mV (deep voltage reset) and d = 8 (large after-spike jump of u). An IB model corresponds to c = -55mV (high voltage reset) and d = 4 (large after-spike jump of u). A CH model corresponds to c = -50mV (very high voltage reset) and d = 2 (moderate after-spike jump of u). A FS model corresponds to a = 0.1 (fast recovery). And a LTS model corresponds to b = 0.25. There are synaptic connection weights which can scale the magnitude of input spike from previous neurons. And each neuron receives a noisy thalamic input, besides the synaptic input. These parameters and weights would be floating-point in software, but this is infeasible in hardware. Thus we use fixed-point for variables, constants and all arithmetic operations.

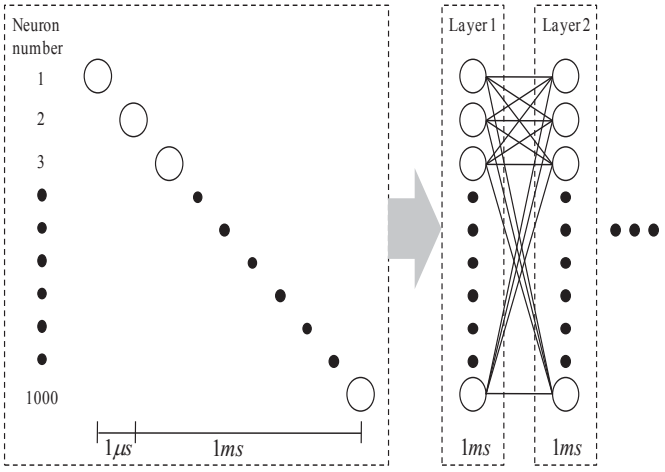

Fig. 2 The hardware structure of proposed SNN

3.3 Operation

To explore the feasibility of the proposed architecture and spiking neuron model explained in the previous section, we implemented the SNN structure on a Xilinx FPGA (part XC7Z020). Fig. 3 shows the overall design of this system. The system repeats the arithmetic operation to simulate the SNN model in time steps. According to izhikevich differential equations and synapse weights, each time step requires calculation of neuron states and summation of scaled input spike.

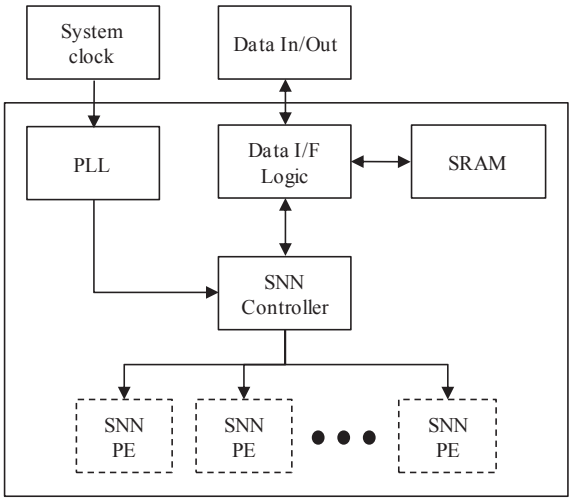

Fig. 3 Overall SNN design on FPGA

Generally, FPGA has several megabytes on-chip memory to store and load frequently accessed data in specific area. In our system, the internal RAM stores the parameters and weights which will be assigned to each neuron. And by request of SNN controller, it loads the parameters for Processing Elements (PEs).

A PLL can adjust the time resolution of SNN controller by dividing a system clock. It is useful for reconfiguring the SNN operating clock or time resolution and for improving the flexibility of system.

The main modules on FPGA consist of the following three components.

1) A SNN controller. This module receives a clock frequency from the PLL and constitutes the operations of the SNN PEs. It provides the equation parameters or synaptic weights to the SNN PEs in each clock-cycle from internal RAM while it delivers the final output data of SNN PEs to the data interface logic. This module also controls the output indices of the fired neurons and postsynaptic current which is accumulated from the previous synapses.

2) A SNN PE. This module is a set of izhikevich neuron models. It implements computations given by equation (7-9) and generates a local spike based on this set of neurons. They are connected together by the SNN controller which manages related parameters and schedules the operation of PEs.

3) A data interface logic. This module can take the neuron parameters and configurable values of network from external memory or test bench. Then it distributes these data to the internal RAM or the SNN controller. Also, it provides parameters of neuron to SNN controller from internal RAM when SNN controller requests these data.

4 Results

We developed a randomly connected SNN architecture on FPGA and the resource utilization of the FPGA implementations is 54% (LUTs).

To examine the performance of system, we use the Xilinx ISE hardware simulation which is shown in Fig. 4. As we intended, the local neuron generates a spike in 1*us* intervals. Also, the figure indicates that the system changes parameters of izhikevich equation to model excitatory or inhibitory cortical neurons.

In order to estimate the spiking or firing rate of neuron models, we extracted the simulated wave. Then, we analyzed the data by using MATLAB, as shown in Fig. 5. In the simulation period (1000ms), we could find that the network exhibits rhythmic activity in delta frequency range around 4Hz. The delta wave is one of the four fundamental types of brain waves, called deep sleep waves [12].

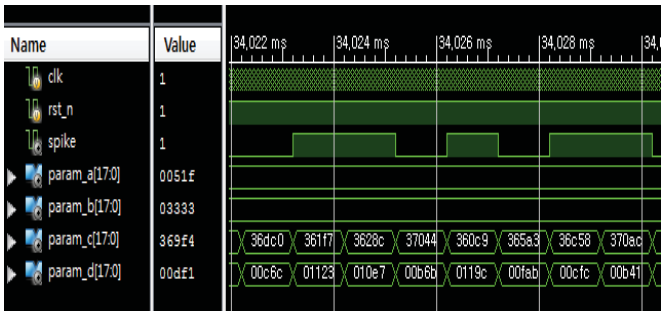

Fig. 4. Result of hardware simulation

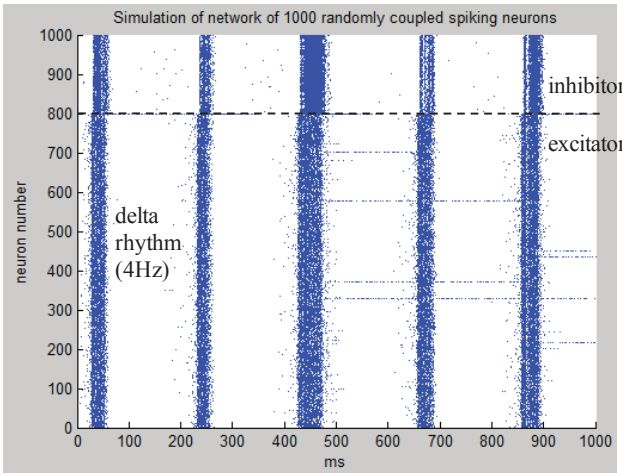

Fig. 5. Simulation of a SNN of 1000 randomly coupled spiking neurons

5 Conclusions

In this paper, we implement the izhikevich neuron model based on FPGA (part XC7Z020) to simulate a large-scale spiking neural network of 1000 spiking neurons with 1 million synaptic connections in real time.

Our SNN structure reproduces the behavior of biological neurons and we could find that our system has biological plausibility. Also, our results indicate that the proposed neural network is suitable for large scale izhikevich models based on cortical neurons while using computational resources efficiently.

6 References

[1] David B. Thomas and Wayne Luk. "FPGA accelerated simulation of biologically plausible spiking neural networks" *IEEE Symposium on Field Programmable Custom Computing Machines*, 2009

[2] Kenneth L. Rice, Mohammad A. Bhuiyan, Tarek M. Taha, Christopher N. Vutsinas and Melissa C. Smith. "FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition" *International Conference on Reconfigurable Computing and FPGAs*, 2009

[3] H. Shayani, P.J. Bentley, and A. M. Tyrrell, "Hardware Implementation of a Bio-plausible Neuron Model for Evolution and Growth of Spiking Neural Networks on FPGA"*, Conference on Adaptive Hardware and Systems 2008 (AHS '08), pp. 236-243, Jun.* 2008.

[4] A. Cassidy, S. Denham, P. Kanold, and A. G. Andreou. "FPGA Based Silicon Spiking Neural Array," *IEEE International Workshop on Biomedical Circuits and Systems (BIOCAS'2007), Montreal, pp. 75-78, Nov*. 2007.

[5] M. La Rosa, E.Caruso, L. Fortuna, M. Frasca, L. Occhipinti, and F. Rivoli, "Neuronal dynamics on FPGA: Izhikevich's model,' *Proceedings of the SPIE*, *vol. 5839, pp. 87-94,* 2005.

[6] M. Mokhtar, D. M. Halliday, and A. M. Tyrrell, "Hippocampus-Inspired Spiking Neural Network on FPGA," *Proceedings of the 8th International Conference on Evolvable Systems: From Biology to Hardware*, *pp. 362-371,* 2008.

[7] Eugene M. Izhikevich. "Which model to use for cortical spiking neurons?" *IEEE Transaction on neural network*, vol. 15, no. 5, Sep, 2004

[8] Jilles Vreeken. "Spiking neural networks, an introduction." *Institute for Information and Computing Sciences, Utrecht University*

[9] Mohammad A. Bhuiyan, Rommel Jalasutram, and Tarek M. Taha, "Character recognition with two spiking neural network models on multi-core architecture" *International Conference on Reconfigurable Computing and FPGAs*, Dec, 2009

[10] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and application to contribution and excitation in nerve" *J. Physiol., vol. 117, pp.500-544,* 1954.

[11] Eugene M. Izhikevich. "Simple model of spiking neurons" *IEEE Transaction on neural network*, vol. 14, no.6, Nov, 2003

[12] Eugene M. Izhikevich . "Polychronization: Computation with Spikes" *Massachusetts Institude of Technology*, 2005