Spiking Domain Feature Extraction with Temporal Dynamic Learning

Honghao Zheng
Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, USA
zhenghh@vt.edu

Yang Yi
Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, USA
yangyi8@vt.edu

2023 24th International Symposium on Quality Electronic Design (ISQED) | 979-8-3503-3475-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISQED57927.2023.10129326

***Abstract*—Spiking neural network (SNN) has attracted more and more research attention due to its event-based property. SNNs are more power efficient with such property than a conventional artificial neural network. For transferring the information to spikes, SNNs need an encoding process. With the temporal encoding schemes, SNN can extract the temporal patterns from the original information. A more advanced encoding scheme is a multiplexing temporal encoding which combines several encoding schemes with different timescales to have a larger information density and dynamic range. After that, the spike timing dependence plasticity (STDP) learning algorithm is utilized for training the SNN since the SNN can not be trained with regular training algorithms like backpropagation. In this work, a spiking domain feature extraction neural network with temporal multiplexing encoding is designed on EAGLE and fabricated on the PCB board. The testbench's power consumption is $400mW$. From the test result, a conclusion can be drawn that the network on PCB can transfer the input information to multiplexing temporal encoded spikes and then utilize the spikes to adjust the synaptic weight voltage.**

***Index Terms*—SNN, feature extraction, multiplexing, STDP**

I. Introduction

With the capability to closely mimic biological neural systems, spiking neural networks have attracted more and more research attention [1]. Unlike traditional artificial neural networks (ANNs), SNN neurons don't transmit signals to receivers in every clock cycle. They only fire spikes when the membrane potential exceeds a specific threshold voltage [2]. That way, SNNs can save unnecessary operational energy and thus improve power efficiency. Moreover, SNN has the potential for temporal learning since it can transfer temporal information into its signals using temporal encoding schemes. Different encoding schemes introduce temporal information in different forms [3]. For example, in the Time-to-first-spike encoding (TTFS) scheme, information is transferred to the time difference between the onset of the encoding window and the first spike. With temporal spiking neurons, the dynamic range of SNN will be significantly improved. A high dynamic range means that neurons can respond to extreme input signals. With the exponential relation of spiking neurons, the system's dynamic range will be way better than conventional ANNs. A higher temporal resolution feature should be added to systems to enable high-efficiency recognition tasks. SNN has the potential to carry out information processing with higher frequency since the computation complexity is lower than ANNs.

Due to the non-differentiable property of spikes, SNN can not be trained with regular training algorithms. For example, backpropagation is unsuitable for SNNs since the spike cannot be differentiated to get a gradient. Thus, researchers have come up with various training algorithms for SNN, such as spike timing dependence plasticity (STDP) [4], Hebbian learning [5], surrogate gradients [6] and NormAD [7]. Among them, the most commonly used is the STDP learning rule. With the STDP algorithm, the synaptic weight between neurons is updated with the relative time difference of pre-and post-synaptic spikes [4]. Nevertheless, the baseline STDP suffers from several shortcomings. The baseline STDP doesn't support supervised learning. It only allows SNNs to divide data points into groups. In addition, even after addressing this issue, the classification accuracy of baseline STDP is relatively lower than other training algorithms utilized in traditional ANNs.

Trying to enable the mentioned valuable features of SNN, our group has been investigating neural encoding schemes for years. Since the rate encoding doesn't possess the property of temporal learning, high dynamic range, and low latency, we skipped the rate encoding and looked directly into the area of temporal neural encoding. In one of the recent papers, [8], we proposed the integrated circuit (IC) design of a multiplexing temporal encoder that combines multiple temporal encoding schemes with different timescales together. SNN can carry out temporal learning with high robustness and low processing latency with such an encoding scheme.

To verify the function of multiplexing temporal neural encoding more deeply, especially investigate the training performance when the STDP training rule is working with the multiplexing encoding schemes, a hardware prototype of the training neural network needs to be implemented. This network must support temporal and multiplexing encoding and be trained in the STDP rule for verification. With these restrictions and requirements, we have designed and fabricated a printed circuit board (PCB) board with a simple training neural network to verify the temporal learning, high dynamic

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, and Grant CCF-1937487.

range, and low latency features of multiplexing encoding to address the low accuracy issue of STDP. Major contributions of our work are summarized as follows:

- Implementation of the spiking domain feature extraction neural network with dynamic temporal multiplexing encoding on PCB board.
- The training neural network on the PCB board can convert the input information to temporal multiplexing encoded spikes and utilize the STDP training algorithm to adjust the synaptic weight voltage according to pre-and post-spike relation.

II. Spiking Neuron Model

For understanding the functionality of spiking neural networks, the most fundamental component of the system that needs to be carefully investigated is the neuron model. Since the beginning of neuroscience, scientists have been focusing on neurons and its property [9]. A neuron has four main parts, dendrites, soma, axon, and synapse. The function of a neuron is to receive a stimulus and output an impulse. When stimulus arrives at dendrites, it will be transformed into the soma, the central computing unit in the biological neural network. Each neuron has a specific threshold voltage, and the neuron fires a spike to the axon only when the input exceeds the threshold. Finally, the synapse will convey the impulse to the subsequent neurons. After a spike is fired, the membrane potential will be reset to the resting stage until the arrival of the following input signal [10].

Researchers have developed several neuron models that can mimic the functionality of biological neurons partially or entirely. Several primary and standard models will be discussed in this paper, including the integrate-and-fire (IF) model, the leaky integrate-and-fire (LIF) model, the Hodgkin-Huxley (HH) model, and the FitzHugh-Nagumo (FHN) model [11].

The IF model is one of the simplest neuron models [12]. This is a basic neuron model that can fire spikes. As illustrated in Fig. 1(a), the IF neuron is built of a capacitor and a resistor in parallel. In this simple structure, the input current, the membrane capacitance, and voltage have a relation as:

$$I_{ex} = C_m \frac{dV_m}{dt} \tag{1}$$

where I_{ex} is the input current, also known as excitation current, C_m and V_m are the membrane capacitance and voltage, respectively. When the voltage across the membrane capacitor reaches a certain threshold voltage, a spike will be fired. However, after the firing process, the voltage across the membrane capacitor will not be reset to the initial level because of the lack of a resetting mechanism. Therefore, the IF model is not able to operate as real biology neurons without resetting mechanism.

Another neuron model is the Hodgkin-Huxley model [13]. It was first introduced in 1952 by Alan Hodgkin and Andrew Huxley, inspired by the ionic mechanism of firing and transmitting neural pulse in the squid giant axon. This

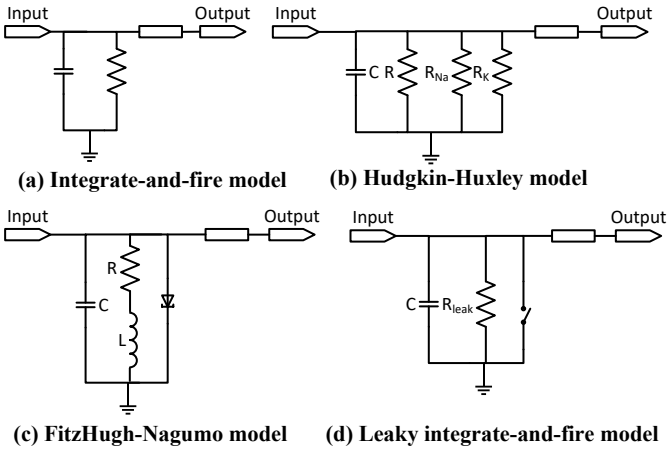

Fig. 1. Simplified neuron models.

model describes the biology and chemistry operation in the neuron. The basic idea of the model is illustrated in Fig. 1(b). The three resistors represent three ion channels, in which R represents the unspecific channel and R_{Na} represents the sodium channel, and R_K represents the potassium channel. The mathematical equation of the HH model can be written as:

$$C_m \frac{dV_m}{dt} = -\sum I_k(t) \tag{2}$$

where $I_k(t)$ is the ion currents through different channels. It can be observed that this model can mimic real biological neurons more accurately. However, because of the complexity of the HH model, it isn't easy to implement this model in hardware. Therefore, a more simplified model than the HH model is required. As shown in Fig. 4(c), the FHN model is a simplified version of the HH model by removing several parameters in the HH model [14]. The mathematical relationship between the parameters within this model is denoted as:

$$\frac{dV_m}{dt} = V_m - \frac{V_m^3}{3} - W + I_{ex} \tag{3}$$

where W is the recovery variable. It is worth noting that the FHN model is still mathematically too complicated to be realized in hardware, which leads our eyesight to the LIF neuron model.

Improved from the IF model, the LIF model has an additional leaky coefficient, representing the diffusion across the membrane [15]. As illustrated in Fig. 4(d), this model consists of three critical components in parallel, the membrane capacitor, the leaky resistor, and the voltage-controlled resetting switch. The relationship between these components can be written as:

$$I_{ex} = I_{leak} + C_m \frac{dV_m}{dt}. \tag{4}$$

Like the IF model, the LIF neuron fires an output spike when the membrane voltage exceeds the threshold. And the voltage across the capacitor will constantly leak out through the resistor if the membrane voltage is below the threshold level. Unlike the IF model, the membrane voltage will be reset

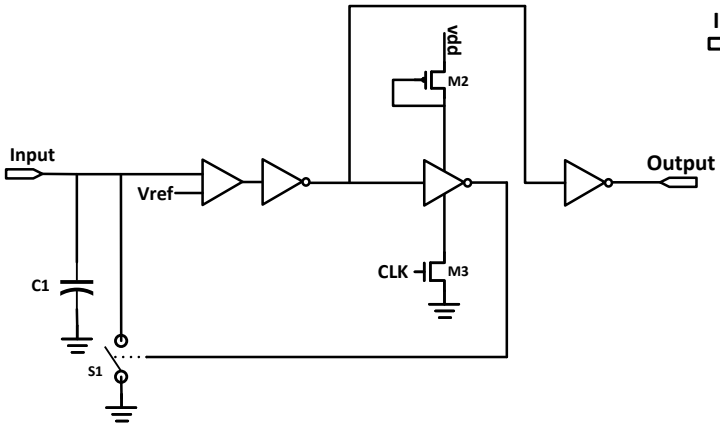

Fig. 2. Spiking neuron circuit schematic.

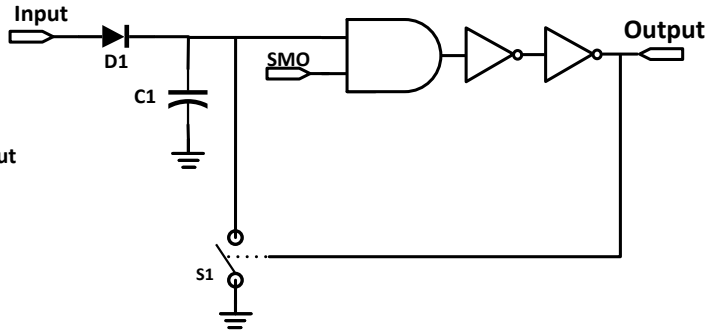

Fig. 3. Gamma alignment block circuit schematic.

to the initial level with the voltage-controlled switch triggered by the spike signal. Therefore, this model has been found to adequately mimic the working mechanism of biological neurons with the simplicity to be implemented in hardware.

From the discussion above, a conclusion has been come up with that the LIF model is most suitable for the latency encoding function for our introduced encoder because it not only can operate like a biology neuron but is also relatively easier to realize at the circuit level.

III. Spiking feature extraction network design

The proposed spiking feature extraction network is designed and optimized using EAGLE. It comprises input neurons, an STDP training circuit, synapses, and output neurons.

A. Spiking Neuron Design

Both the input and output neurons have the same structure. The neurons accept current inputs and give analog spike signal outputs. The neurons are built based on the structure discussed in Section. II, the LIF neuron. The circuit schematic of the spiking neuron is shown in Fig. 2. When the input current is integrated into the membrane capacitor, the membrane voltage is increased with the integration. After that, a comparator compares the membrane voltage with a threshold voltage. The comparator will output digital high voltage when the membrane voltage exceeds the threshold voltage. The digital high voltage will go through a two-inverter formed buffer to stabilize it. Since the output signals of the neuron are spikes, a resetting mechanism is implemented to bring the membrane voltage back to zero. This mechanism is realized by the p-channel transistor, the inverter and the n-channel transistor formed switch stage. The inverter ensures that a triggering voltage will be sent to the analog switch whenever a spike is fired to drag the membrane voltage back to the ground. The nmos transistor ensures that when the CLK signal starts a new sampling window, the analog switch will be triggered to restart the integration process on the membrane capacitor from the ground.

With the latency neuron being implemented, the training neuron and the temporal encoding scheme in the neural network are both achieved. The latency neuron can be used in the training neuron layers. It can also be used as the temporal TTFS encoder.

B. Gamma Alignment

As mentioned in Section. I, to utilize the outstanding feature of temporal learning, high dynamic range, and low latency in the neural network, the multiplexing encoding needs to be implemented on the PCB board. With the TTFS encoder designed, a gamma alignment function block is the only block necessary to achieve multiplexing encoding. As shown in Fig. 3, the gamma alignment block consists of one diode, capacitor, AND gate, inverters, and an analog switch. The diode and the capacitor form a spike detector. When a spike enters the block, the diode will pass the voltage to the capacitor and stop it from decreasing after the spike ends. After that, the AND gate will wait for the subthreshold membrane oscillation (SMO) signal. When the peak voltage of the SMO comes, the AND gate will fire digital high voltage and the two inverters will hold the voltage steady. To reset the spike detector, the analog switch will use the output spike to drag the hold voltage back to the ground, making the block ready to shift another spike.

C. STDP Training Circuit

Unlike the conventional artificial neural network, SNNs can not be trained with a traditional algorithm like backpropagation. The STDP is the most commonly used algorithm for SNNs. We have implemented an STDP training circuit block on the PCB board for training the neural network. The structure is inspired by [——] but has been modified and improved for implementation on the PCB board. As shown in Fig. 4, the STDP training circuit consists of four analog switches, six transistors, three capacitors, and one inverter. The inverter is used for flipping the post-neuron spike so that the charge integration on C3 is available. Due to the symmetric mechanism, we will mainly discuss the long-term potentiation (LTP) process. When there is no spike, the voltage at C1 will increase at a rate that Vtp controls so that the pmos transistor M3 is closed. When a pre-neuron spike comes, the analog switch S1 is open and the voltage at C1 will be dragged down for a specific value controlled by Vwp. Thus, M3 is

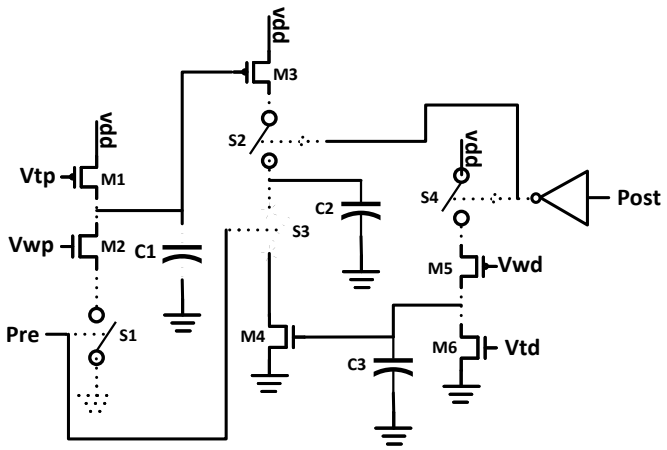

Fig. 4. STDP training block circuit schematic.

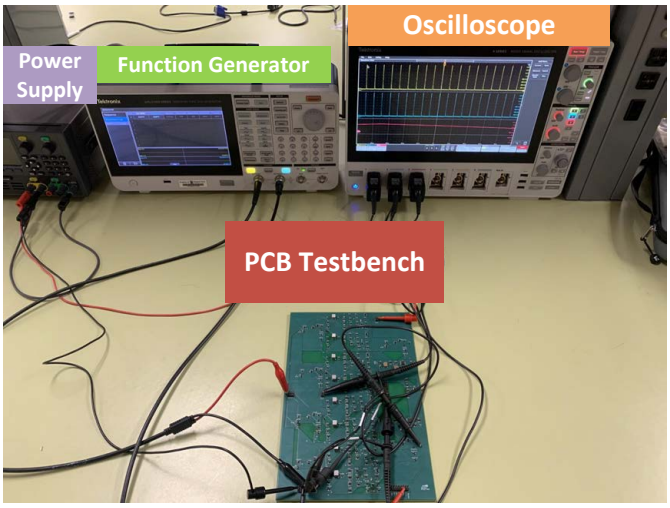

Fig. 5. The testbench setup for PCB spiking feature extraction neural network.

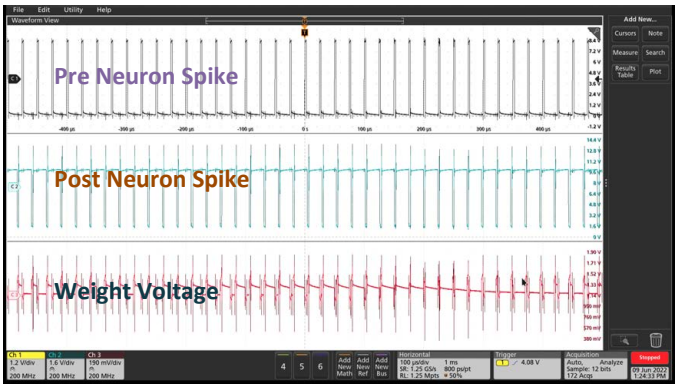

Fig. 6. The PCB spiking feature extraction neural network test result.

open and the potentiation process only needs another post-neuron spike to open the pmos M4. When the post-neuron spike comes, M3 and M4 are both open, so the voltage across the weight capacitor C2 increases. Since the voltage across C1 will increase exponentially after the pre-neuron spike, the closer the post-spike to the pre-spike, the more the weight voltage increases. The long-term depression (LTD) mechanism is similar to the LTP process. The only differences are the voltage across C3 is slowly decreasing after the post-spike and when the pre-neuron spike fires after the post-neuron spike, the weight voltage will decrease.

IV. Result Analysis of the Spiking Feature Extraction Neural Network

This section will discuss and analyze the testbench setup and the measurement results of the spiking feature extraction neural network. As mentioned in Section. III, the training neural network comprises input neurons, an STDP training circuit, synapses, and output neurons. To verify the efficiency of the network, the power consumption of the spiking feature extraction neural network is also measured. From the measurement, it has been easily tested that the power consumption of this neural network on the PCB board is $400mW$.

As depicted in Fig. 5, besides the PCB board, the first testbench also consists of one power supply, one function generator, and one oscilloscope. Obviously, the power supply is used for the supply voltage. The function generator is utilized for the CLK signals in the neurons. Only with these CLK signals defining encoding windows will the neurons be able to carry out the signal processing and encoding tasks. To verify the dynamic range feature of the proposed neural network, we also utilized the function generator to provide extreme signals to the circuit. From the oscilloscope, it is easy to notice that even with extreme input signals, the circuit still operates properly since that with the variance of the pre-and post-spike relation, the weight voltage changes according to it, as shown in Fig. 6. When the pre-spike arrives before the post-spike, the weight voltage will rise. The weight voltage will decrease when the post-spike arrives before the pre-spike, indicating that the two neurons are not that relative.

With the discussion above, a conclusion can be drawn that the spiking domain feature extraction network can convert the input information to spike signals and then use the STDP training algorithm to adjust the synaptic weight voltage within a large dynamic range.

V. Conslusion

In this paper, we proposed a novel design of the spiking domain feature extraction neural network with the temporal multiplexing encoding scheme. We have also fabricated the hardware prototype of the neural network on the PCB board. A large dynamic range has been achieved with the temporal multiplexing encoding scheme since the temporal multiplexing TTFS-phase encoding can handle more extreme inputs and stay reliable compared with normal encoding schemes. With the SNN functionality, the neural network can extract features from input information and transfer them as spikes. From the test result of the hardware prototype, it is noticeable that using the spikes, the synaptic weights can be trained with the STDP

training algorithm so that the neural network can be utilized for the following classification tasks,

REFERENCES

[1] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International journal of neural systems*, vol. 19, no. 04, pp. 295–308, 2009.

[2] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural networks*, vol. 111, pp. 47–63, 2019.

[3] C. Zhao, B. T. Wysocki, C. D. Thiem, N. R. McDonald, J. Li, L. Liu, and Y. Yi, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 4, pp. 265–276, 2016.

[4] T. Iakymchuk, A. Rosado-Muñoz, J. F. Guerrero-Martínez, M. Bataller-Mompeán, and J. V. Francés-Víllora, "Simplified spiking neural network architecture and stdp learning algorithm applied to image classification," *EURASIP Journal on Image and Video Processing*, vol. 2015, pp. 1–11, 2015.

[5] Y. Munakata and J. Pfaffly, "Hebbian learning and development," *Developmental science*, vol. 7, no. 2, pp. 141–148, 2004.

[6] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[7] N. Anwani and B. Rajendran, "Normad-normalized approximate descent based supervised learning rule for spiking neurons," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[8] H. Zheng, K. J. Bai, and Y. Yi, "Enabling a new methodology of neural coding: Multiplexing temporal encoding in neuromorphic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2023.

[9] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[10] K. Bai, Q. An, L. Liu, and Y. Yi, "A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 1, pp. 62–75, 2019.

[11] C. Zhao, Y. Yi, J. Li, X. Fu, and L. Liu, "Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 8, pp. 2193–2205, 2017.

[12] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[13] M. Nelson and J. Rinzel, "The hodgkin-huxley model," *Bower, JM and Beeman, editors, The book of Genesis*, pp. 27–51, 1995.

[14] E. M. Izhikevich and R. FitzHugh, "Fitzhugh-nagumo model," *Scholarpedia*, vol. 1, no. 9, p. 1349, 2006.

[15] C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz *et al.*, "Generalized leaky integrate-and-fire models classify multiple neuron types," *Nature communications*, vol. 9, no. 1, p. 709, 2018.

[16] F. Nowshin and Y. Yi, "Memristor-based deep spiking neural network with a computing-in-memory architecture," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2022, pp. 1–6.