

Spiking Neural Network For Energy Efficient Learning And Recognition

Wang Ning Lo¹, Yan Chiew Wong^{2*}

Abstract: Nowadays, people are confronted with an increasingly large amount of data and a tremendous change of human-machine interaction modes. It is a challenging and time-consuming task for traditional computing system to deal with the content of information. The use of applications consumes energy and hard to perform through standard programmed algorithms. Spiking neural networks have emerged that achieve favourable advantages in terms of energy and time efficiency by using spikes for computation and communication as well as solving different problems such as pattern classification and image processing. Therefore, an energy-efficient spiking feedforward computing system is presented to evaluate its performance. Common building blocks and techniques used to implement a spiking neural network are investigated to identify design parameters for hardware-based neuron implementations. Izhikevich neuron, Address-Event Representation system and Spiking-Timing-Dependent Plasticity module are developed by using Vivado software. Demonstration of digit recognition using SNN hardware implementation on FPGA has been performed. The energy consumption of the system is only 136mW and low hardware resource utilization has been observed. This work presents essential properties of a spiking feedforward computing system that emulates the behaviour of biological neural networks, showing the potential for learning and classification in significantly reduced energy resources.

Index Terms: Spiking Neural Network, Neuromorphic, Digit Recognition, FPGA.

1 INTRODUCTION

Nowadays, it is a challenge for the scientific society in this generation to comprehend and replicate the function of the brain with the same characteristics whether in biological or electronic fields. Artificial intelligence (AI) can be known as the simulation of human intelligence processes by machines. Due to discoveries of the biological process, a modern approach of artificial intelligence is attracted attention by a lot of scientists since it can reproduce the neurology of the mammalian brain. For instance, AI is widely used to generate data analysis in real time. Still, current modern computers are inefficient to perform the tasks of recognition, analyzation as well as data classification. Hence, neuromorphic computing is served to overcome this gap by emulating certain aspects of brain functions. The combination of both computation and memory emulating neurons as well as synapses for the brain-inspired architecture are needed for the achievements of next-generation AI systems. Neuromorphic computing technology can be served for integrating algorithms which can support real-time learning with architectures built on new computing hardware for addressing specific user applications. Neuromorphic devices are widely used to mimic the situation of the brain's architecture and dynamics to reproduce their functional abilities in connection with computational power, sturdy learning as well as energy efficiency. It is undeniable that the research community is attracted attention by the bio-inspired systems due to their computational power [1]. Although there is a lack of consensus about the information processing in the brain of the mammals, biological processes have functioned as references for recent computational models.

Artificial Neural Networks (ANNs) can be considered as the biological neural networks simplified versions which in terms of function and structure. In previous research work [2][3][4], the weights and the biases of neural network have been trained with several techniques for various applications such as speech and number recognition. The latest generation of neural networks is named as Spiking Neural Networks (SNNs) are introduced as the more biological realistic method since it produces the spikes as well as incorporates the time and space through neural plasticity and connectivity. SNNs provide optimal characteristics for hardware implementation to achieve online operation. Embedded hardware systems are utilized to enlarge the applications where this neural network can be implemented like a custom VLSI chip [5]. As aforementioned, the energy-efficient spiking feedforward computing system is the primary design in this work. The design and physical implementation of a computing system can replicate the behavioural properties of biological neural networks to perform various functions such as learning and recognition which can associate with the human brain. The designed system will demonstrate its rich dynamic behaviours with low power consumption. Hence, three modules like Izhikevich neuron model, Address-Event Representation (AER) communication system and Spike-Timing-Dependent Plasticity (STDP) training algorithm are developed to emulate the function of the neural networks.

2 SPIKING NEURAL NETWORKS

Spiking Neural Networks (SNNs) have been developed in the past decades in order to transfer information which can mimics in biological neurons [6]. These networks consist of spiking neurons. Figure 1 depicts the connection of two biological neurons. Two biological neurons communicate through sequences of spikes. The main parts of the neuron are the dendrites, the axon and the soma. The presynaptic neuron links with the postsynaptic neuron through the synapses. It also can be represented by the spiking neuron model. The information from the inputs will be processed by the neuron in order to generate a single spiking signal at the output. The probability of generating a spike depends on the inputs. Besides that, one state variable is needed to characterize its dynamics behaviour. A spike will be produced by the model

- Wang Ning Lo¹, Yan Chiew Wong^{2*}
- Centre for Telecommunication Research & Innovation (CeTRI),
Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer (FKEKK),
Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100,
Durian Tunggal, Melaka, Malaysia. Email: ycwong@utem.edu.my

when internal variables achieve a certain threshold. Several neuron models such as Leaky Integrate-and-Fire (LIF) model, Hodgkin-Huxley (HH) model Izhikevich (IZH) model are described in this section. HH model requires large number of parameters and equations to regenerate different type of neurons with good accuracy. Two equations are involved in LIF model to reproduce only a few types of neurons. Hence, IZH model is implemented by altering four parameters to regenerate different families of neurons based on two equations. This model has a key advantage of resource-frugal especially on designing a large CPG network embedded in a single board since additional modules needed for hybrid experiments [7].

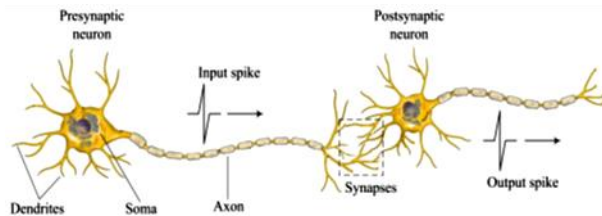


Fig. 1. Connection of two biological neurons.

2.1 Leaky Integrate-and-Fire (LIF) model

Leaky Integrate-and-Fire (LIF) neuron model is a combination of a capacitor and a “leaky” resistor in parallel as shown in Figure 2. Current source as a synaptic current input is applied to charge up the capacitor for producing a potential. A low-pass filter converts the incoming spike which means presynaptic action potential coming from another neuron into a current pulse, $i(t)$ that flows into the postsynaptic neuron. After that, the voltage-controlled switch is used to discharge the capacitor to a resting potential like a biological neuron when potential surpasses a threshold value. A low pass filter represents the synapse. The presynaptic and postsynaptic neurons will communicate through this synapse. The resulting current pulse starts to charge the LIF circuit and increase the membrane potential, $u(t)$. A spike is fired by the neuron at specific time, t^f whenever the membrane potential achieves a specific threshold value. Thus, the neuron state is reset after firing and maintained at the refractory period [8].

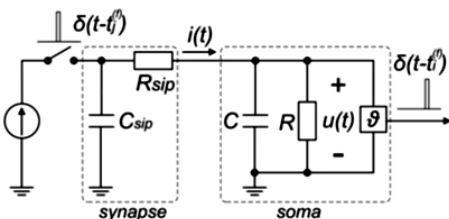


Fig. 2. Leaky Integrate-and-fire (LIF) neuron model.

2.2 Hodgkin-Huxley (HH) Model

In 1952, HH model had been described to define the mechanisms of the ions in the squid giant axon underlying the initiation and propagation of action potentials [9]. Data extracted from the giant axon of a squid is compared with HH model, it demonstrates the highest similarity to the biological neuron [10]. HH neuron model demonstrates the neuron membrane potential which the dynamic behaviour of the different ion channels of the dendrites and soma[11]. Figure 3 describes the equivalent circuit of HH neuron model. The membrane capacitance is associated with the current across

the membrane as well as the ionic movement through resistive membrane channels. Thus, the ionic current, I_{ion} in this model is divided into three components which are small leakage current, I_L , potassium current, I_K and sodium current, I_{Na} that is mainly conveyed by chloride ions.

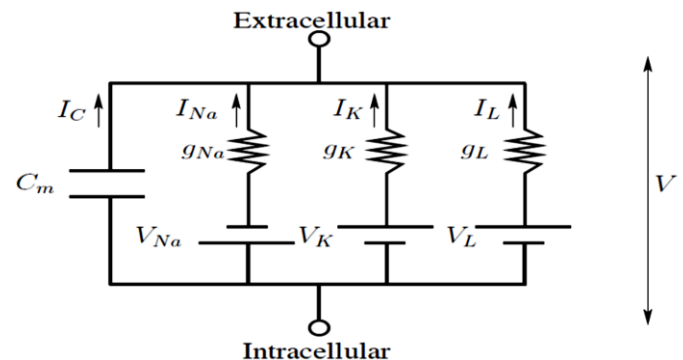


Fig. 3. Hodgkin-Huxley (HH) neuron model.

2.3 Izhikevich (IZH) Model

In this work, Izhikevich (IZH) model is developed to build the large-scale models of brain impulses by neural networks [12]. In 2003, IZH model was introduced by Eugene M. Izhikevich for spiking neural networks. Figure 4 shows Izhikevich model. This model is biologically same as Hodgkin-Huxley model and can compute efficiently as Integrate-and-Fire model. The accuracy of physical and biological models of HH neurons have been deduced into two-dimensional system of ordinary differential equations [13]:

$$v' = 0.04v^2 + 5v + 140 - u + 1 \quad (1)$$

$$u' = a(bv - u) \quad (2)$$

with the auxiliary after-spike resetting

$$v \geq 30mV, \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (3)$$

v and u are variables while a , b , c and d represent parameters. All are dimensionless. v is membrane potential of the neuron and variable u is the membrane recovery. u emulates the inactivation of sodium ionic currents and activation of potassium ionic currents. Then, it gives negative feedback to v . The value of the membrane potential will be increased by positive synaptic currents, I from the other neurons. If these currents are not sufficient to produce spike, the voltage membrane will be reset. Besides that, if the spike is produced more than or equal to +30 mV due to sum of their input current, v and u are reset based on the Equation 3. The remaining membrane voltage in this model depends on the parameter b . The voltage is between -60 mV to -70 mV. In addition, this model only depends on the previous value of membrane potential before the threshold potential since it does not have a specific threshold for the spike generation can be as high as -40 mV or as low as -55 mV. The following factors need to be considered for the use of the parameters which are a , b , c and d displayed in this model. Parameter a means the time-scale recovery of u . The smaller the values slowing the recovery. The representative value for a is 0.02. In addition, b means the sensitivity of u to the subthreshold fluctuations of v . The large values for recovery membrane and

membrane potential will increase threshold oscillations and low-threshold spiking dynamics. The representative value for b is 0.2. Next, c depicts the after-spike reset value of the membrane potential caused by the fast high-threshold K^+ conductance. The typical value for c is -65 mV. Furthermore, d represents the after-spike reset of the recovery variable caused by slow high-threshold Na^+ and K^+ conductance. The typical value for parameter d is 2.

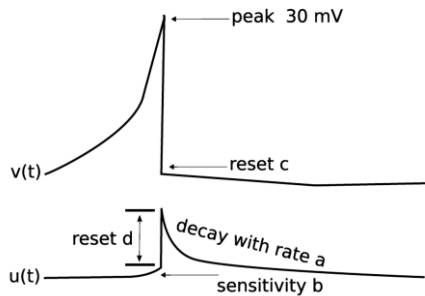


Fig. 4. Izhikevich (IZH) model.

3 COMMUNICATION SYSTEM

Two systems will be described in this section which are Address-Event Representation (AER) and Networks-on-Chip (NOCs). AER is more suitable for spike-based computations since it can construct large-scale networks with arbitrary and configurable synaptic connectivity compared with NOCs.

3.1 Address-Event Representation (AER)

Misha Mahowald first proposed an event representation system to transmit the neurons' pulse on a chip to the appropriate destination. It is done in the array of neurons in another chip [14]. Based on Figure 5, the mechanism of Address-Event Representation (AER) system is described. AER system can be divided into encoder and decoder. An encoder provides unique address from the neuron which produces a spike. After that, these addresses will be transmitted to a decoder through the bus in order to choose the suitable spike's location. The efficiency of this AER system is able to prevent the occurrence of bottlenecks when the data requires to be switched in a system consisted of a massively interconnected component such as SNN models. Besides that, some factors require to be addressed to implement AER communication system effectively. The first factor is about the step used for selecting the events to be delivered. If many events are generated at one period, the system are required to make the decision about the order of the events to be delivered through the address bus since it only assigns one address at one clock cycle. Another factor is the access technique applied for this system. The bus with various components works asynchronously and independently, thus an appropriate procedure must be set up to overcome the collisions. Moreover, some techniques [15], including arbitration access or ALOHA-based access as well as sequential scanning have been suggested but they do not provide proper guidelines for specific applications.

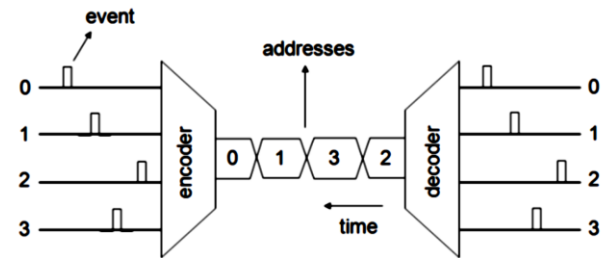


Fig. 5. The mechanism of AER system.

3.2 Networks-on-Chip (NOCs)

Networks-on-chip (NOCs) are implemented for spike communication [16]. They can receive and deliver digital information through a time-multiplexed shared bus. NOCs are required to implement for large-scale chips because connectivity in a silicon fabrication process has limitation of flexibility in the third dimension. However, the on-chip connectivity still cannot match the three-dimensional connectivity found in the brain. In [17], energy efficiency for SNN-based applications have been demonstrated towards biofidelic implementations. Due to limited connectivity and constrained bus bandwidth, NOCs require further investigation for spike communication.

4 LEARNING METHODS

Synaptic plasticity means the modification of the synaptic strength due to their connections. It arguably contributes to memory and learning in biological neural networks. Hence, two methods such as Spike-Timing-Dependent Plasticity (STDP) and Supervised Hebbian Learning (SHL) are discussed in this section. STDP is for unsupervised learning while SHL is for supervised learning. STDP is self-learning technique which works with unlabeled data. The output is based on the collection of perceptions. It can be executed in real-time analysis and less computational time for training compared with SHL.

4.1 Spiking-Timing-Dependent Plasticity

Spike-Timing-Dependent Plasticity (STDP) is an approach to establish artificial neural networks for performing complex computational operations [18]. STDP can process biologically to modify the synapses and connections of the neurons in SNN. It can be used to enhance or weaken the connection of each neuron according to the degree of synchronous firing. A method of unsupervised learning is applied to strengthen synapsis that contribute to the output spike generation. Hence, for those do not have contribution, those spikes that generate after the output spike will be weakened [19]. Based on presynaptic i and postsynaptic j neurons, the equation that depicts the changes in synaptic weight as:

$$\Delta w_j = \sum_{k=1}^N \sum_{l=1}^N w(t_j^l - t_i^k) \quad (4)$$

The function below determines the degree of add or reduce of the synaptic weight. It is based on the spiking time between pre-synaptic and post-synaptic neurons can be represented as:

$$W(x) = \begin{cases} A_+ \exp(-\frac{x}{\tau_+}) & \text{if } x > 0 \\ A_- \exp(-\frac{x}{\tau_-}) & \text{otherwise} \end{cases} \quad (5)$$

t_j^l is the activation time l^{th} of the neuron j , and t_j^k is the activation time k^{th} of the neuron in Equation 4. In Equation 5, A_+ and A_- can be defined as the two constants that represent the variation in the synaptic weight at $t = 0_+$ and $t = 0_-$ respectively. Besides that, τ_+ and τ_- represent the time constants of the exponential decrease in the variation of synaptic weight.

Figure 6 describes the graphical representation of the STDP learning rule [18]. From Equation 5 of relative weights changes according to the period between the spikes which are pre- and post-spikes of the synapsis among two neurons. Reduction of synaptic weight will be occurred when the postsynaptic neuron fires before the presynaptic neuron and vice versa.

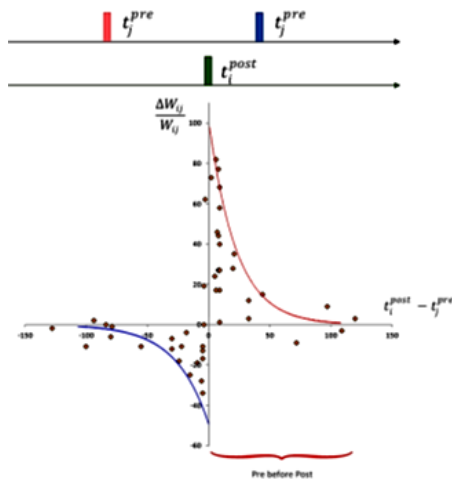


Fig. 6. Graphical representation of the STDP learning rule [18].

4.2 Supervised Hebbian Learning

Supervised Hebbian Learning (SHL) is known as the most plausible supervised SNN learning algorithm in biological neural networks [20]. With the additional of ‘teaching’ signal, an output neuron can be fired at the specific time by using SHL. Supervision may be envision as supervision by other neurons due to the intracellular synaptic currents in ‘teaching’ signal. Method SHL suffers from the limitations that even after goal firing pattern has been reached and the weights are changed. In order to ensure stability, the learning rule must involve the constraints, but it is difficult to set the constraints. The weights will increase after every training period and unstable network will be occurred or at least to generalize weakly in the testing phase [21]. In [20], the first spike-based methods similar with SHL approach has been proposed. The monosynaptic excitation is defined in this learning rule. Two pre-synaptic and one post-synaptic spikes are produced during every learning period. The first presynaptic spike as input signal at the time t_1^{in} , whereas the target firing time for the postsynaptic neuron represents the second presynaptic spike at $t_2^{in} = t^d$. The learning rule is shown at Equation 6 where t^d represents the actual time of the postsynaptic spike and $\eta > 0$ represents the learning rate.

$$\Delta w = \eta(t^{out} - t^d) \quad (6)$$

5 SPIKING FEEDFORWARD COMPUTING SYSTEM

5.1 Simulation of IZH Neuron Model

The designed module for the Izhikevich neuron consists of seven entries. Besides that, it has a small RAM. Various synaptic weights connection with the other neurons are stored in RAM. Figure 7 shows the digital block of IZH neuron. A clock signal CLK coordinates the various operation of the neuron. Activation signal EN plays a role to launch the capability of the neuron. Next, the internal RAM of the neuron can be written by using the signals synaptic weight Weight, write enable WE and address Addr. Furthermore, the input of AER_Bus enables the neuron to read which neuron generated a spike. An output Spike_out represents whether a spike is generated by the neuron.

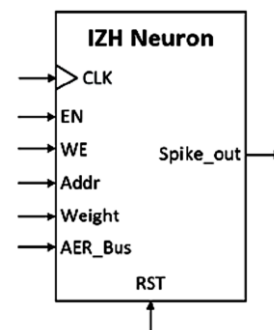


Fig. 7. Digital block of a neuron.

In order to lower a level of neuron implementation, there are various combinational and sequential blocks that manipulate the activities of the neurons shown in Figure 8 [22]. Block diagram of the neuron consists of internal RAM and two registers of the voltage recovery and membrane potential. The two combinational blocks based on the execution of differential equations and a sequential block for the Input Align of weight synaptic are also implemented in this block diagram.

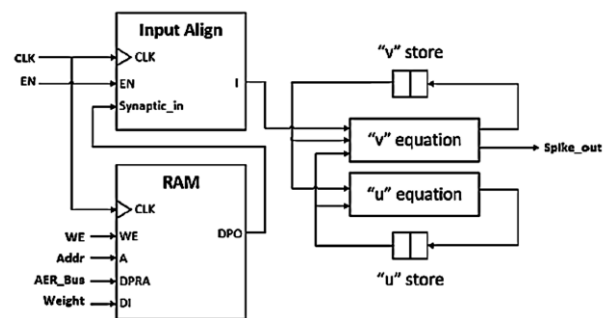


Fig. 8. Block diagram of the neuron [22].

Figure 9 shows the response of the IZH neuron based on the input step of 12 mV. Synaptic weight Weight in the testbench has been set to 120 which represents 12 mV. It refers to the strength or amplitude of a connection between two nodes, which influences the firing of the neurons. v_3 signal depends on

Spike_out based on the synaptic weight to perform its own functionality. The WE signal is activated at 100 ns and the neuron '1' with synaptic weight of 12 mV is written inside the internal RAM. Then, AER bus locates the address of the neuron '1' after a few clock cycles. Then, the neuron is firing at constant state and therefore it generates the input step of 12 mV for the emulated neuron. At the end, the neuronal impulses will be generated by the neurons to the output signal Spike_out. Figure 10 and 11 show the response of the IZH neuron based on the input step of 30 mV and -15 mV, respectively. The input step of 30 mV will generate a higher frequency of spikes if compared to the response based on the input step of 12 mV. For the inhibitory input step of -15 mV, the neuron cannot produce any spikes. Through the simulations, IZH neuron emulates the behaviours displayed by the original model in [12].

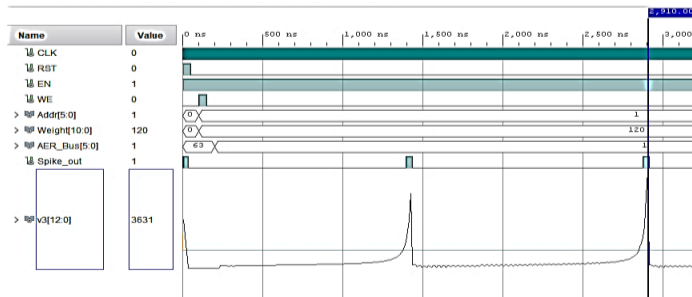


Fig. 9. Response of the IZH neuron based on the input step of 12 mV.

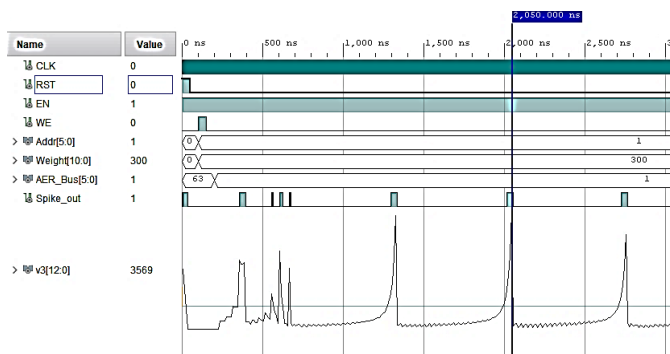


Fig. 10. Response of the IZH neuron based on the input step of 30 mV.

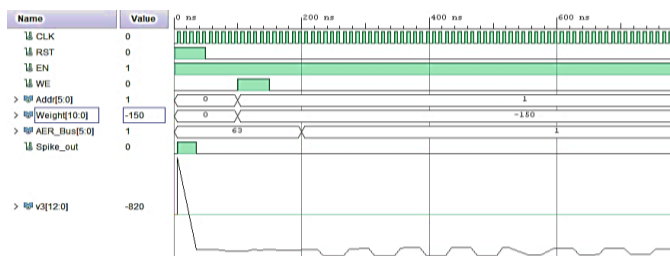


Fig. 11. Response of the IZH neuron based on the input step of -15 mV.

5.2 Simulation of AER System

Figure 12 shows the design of AER system block diagram. AER system consists of two outputs and two inputs. The two inputs include spikes vector Spikes and clock signal CLK. The outputs are the AER communication bus and the neurons' activation signal EN_Neuron. A whole AER system involves

four components such as comparator, priority encoder, multiplexer, and FIFO (First in, First out).

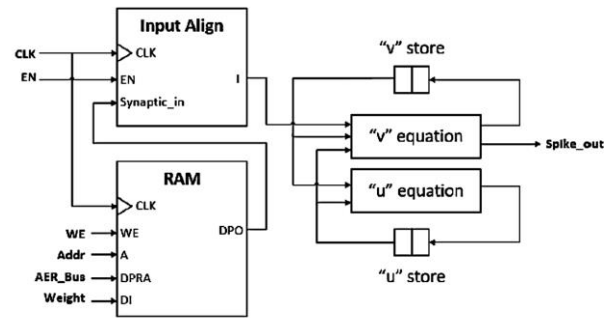


Fig. 12. Block diagram of AER system [22].

Figure 13 displays the behaviour shown this communication bus corresponding to the spike's generation. Initially, it writes all their bits to '0' since there is no spike detected by the AER bus. In this situation, it shows the number '31' since this bus is able to display up to 31 addresses. It starts from the address of 0 to 30. A spike is produced at the neuron '0' at 10ns. In the next clock cycle, AER bus will write the '0' address in its output. Several spikes are produced by neurons '0', '1', '3', and '4' at 40ns. The priority encoder starts to function then the EN_Neuron will not be activated in order to end up the action of all the neurons. Hence, it can write the address of the firing neurons in every clock cycle one by one. The spikes neurons '1' and '2' are fired at 60 ns. FIFO is used to store the corresponding spikes vector. When the priority encoder completes the transmission of spikes for neurons '0', '1', '3', and '4', the addresses of neurons '1' and '2' which generate at 60ns will be written in AER bus to show its own functionality.

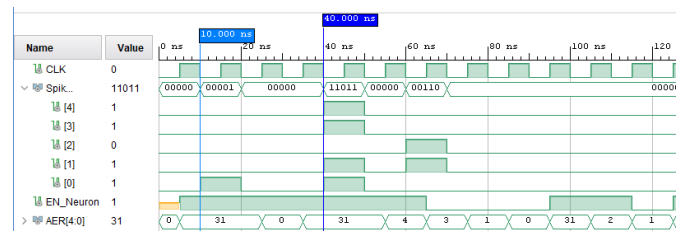


Fig. 13. Timeline of the AER behaviour.

5.3 Simulation of STDP Learning Module

Spiking-Timing-Dependent Plasticity (STDP) is a training system to modify the weights of all the connections of the neural network. Figure 14 shows the digital block of the STDP module. It consists of six inputs and three outputs. The clock signal coordinates the various operations of the learning module. EN_Addr is used to change the connection when STDP rules are implemented while the module learning is activated by an activation signal EN. When the STDP module is connected respectively, Pre_Spike and Post_Spike can read the fired previous neurons and the output spike of the neurons. Three output signals such as WE, Addr and Weight allow to write in neuron's RAM. In addition, reset and enable signals permit to restore all the synaptic weights of the neuron's RAM.

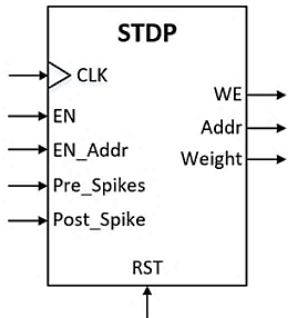


Fig. 14. Digital block of STDP module.

The STDP module is composed of some interconnected sequential blocks and combinational blocks for it to work. Figure 15 depicts the block diagram of the STDP module with counter address (Addr cnt), incrementor decrement link selector, (I/D Sel.), and synaptic weight counter (Weight cnt). Besides that, digital logic of equations is implemented by using a set of combinational blocks from the STDP learning rule. The address counter can choose on which synaptic connects with the module rule. The link selector will activate the relevant signal whether the pre-spike occur within the connection of post-spike. The synaptic weight counter can be used to store and modify the synaptic weight of all the neuron's connection.

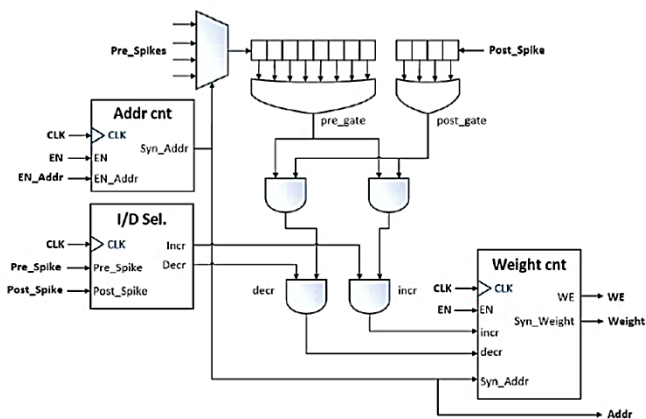


Fig. 15. Block diagram of STDP module [22].

Figure 16 displays different behaviours shown by STDP module. The inputs and outputs along with internal signals of the STDP module were constructed in order to understand the inner function of the module. The neuron obtains the same tendency of spikes generation from the other neurons which is connected to approximately every 200 ns. Hence, the synaptic connection can be altered by the STDP module for every moment that the spikes are received. Initially, the first synapsis which relates to the first bit of Pre_Spikes vector is renewed with a weight value of 6. In order to operate the second synapsis, the EN_Addr signal is activated since there is no spike from it. EN_Addr signal will be triggered again for applying the STDP learning in the third synapsis. The weight value will be renewed to 4 due to the time difference among the two spikes.

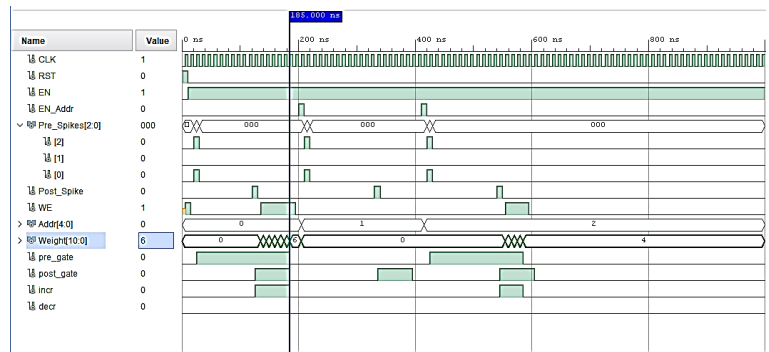


Fig. 16. Timeline of inputs and outputs of the STDP learning module.

6 SIMULATION AND HARDWARE IMPLEMENTATION FOR DIGIT RECOGNITION

6.1 Simulation of Spiking Feedforward Computing System for Digit Recognition

This section will discuss on the simulations of spiking feedforward computing system for digit recognition implementation by using Vivado software to demonstrate its operation. Figure 17 and 18 illustrates the complete simulation of the spiking feedforward computing system for digit recognition of digits zero to five and digits four to nine represented by output neurons N41 to N46, respectively. From Figure 17, images from digits 0 to 5 have been instructed to neurons N41 to N46, respectively. The pulses of EN_STDP signal will show the output for digit recognition. Six learning phases are implemented that each digit represents each output neuron of the SNN. The image vector '000000001' corresponds to digit 0 until '0000100000' represents digit 5. The pattern of digit 0 is chosen along with the training neuron that relates to N41 for first phase of the training. Initially, the output neuron does not perform for spiking. After the training period, the synapses that dedicate to its firing are changed. Hence, N41 stops learning and starts to produce spikes for the selected pattern. During the fourth phase of the training for digit 3, the training neuron of the output neuron N44 is chosen. If N41 is firing when the fourth phase is starting, the training phase will stop firing and only N44 can generate spikes for the chosen pattern. It same goes to the other phases of the training process until all the training is completed.

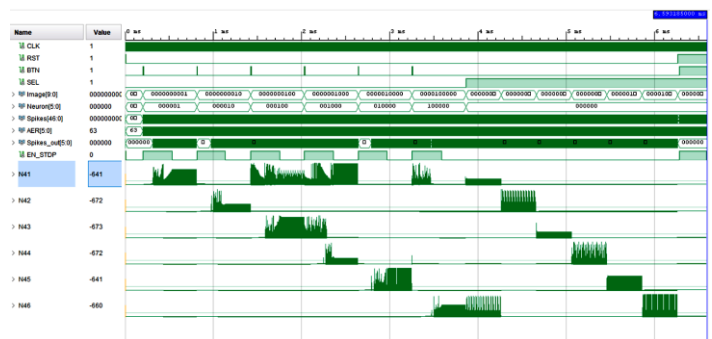


Fig. 17. Simulation of spiking feedforward computing system for digit recognition digits 0 to 5.

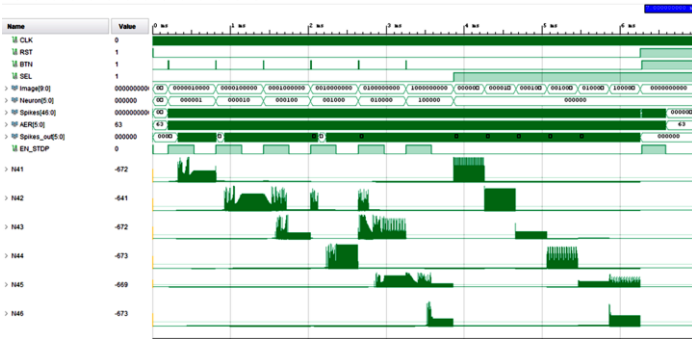
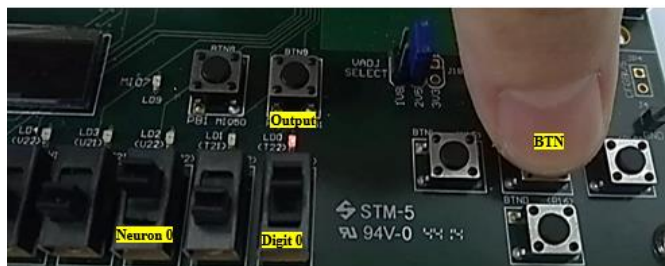


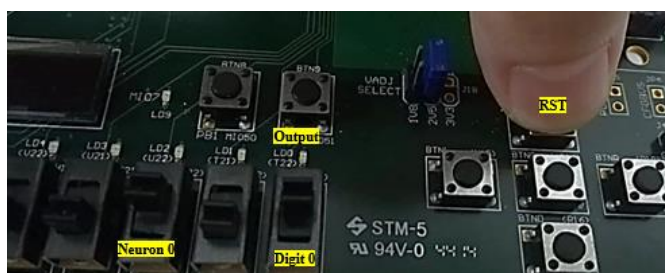
Fig. 18. Simulation of spiking feedforward computing system for digit recognition digits 4 to 9.

6.2 Hardware Implementation for Digit Recognition on FPGA

The zedboard Zynq-7000 is used to implement for digit recognition. Two buttons (BTN, RST), four switches (SW0 to SW4) and two led outputs (LD0 and LD1) were applied for digit recognition. BTN button was set to perform the training pulse for input stimuli; RST button was set to reset all the synaptic weights. SW0 and SW1 represented the stimulus for the input layer of the neural network of digit 0 and digit 1. SW2 and SW3 represented neuron 0 to neuron 1 which allowed to select the training neuron to the introduced digit for the training. When the switch for the selected digits and neurons are turned on, BTN is pressed to activate the recognition. The output of LED will be lighted up which means the spike is generated for the selected pattern. Figure 19 and 20 shows recognition for digit 0 and 1 when (a) when BTN button is pressed and (b) when RST button is pressed.

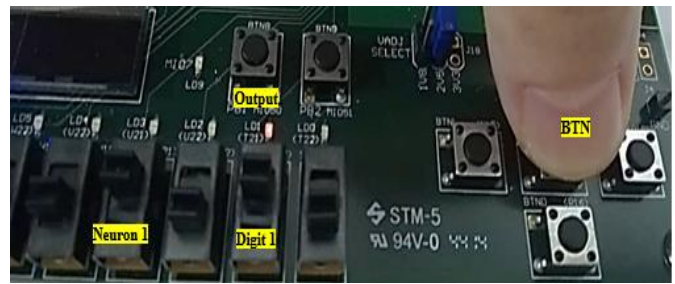


(a)

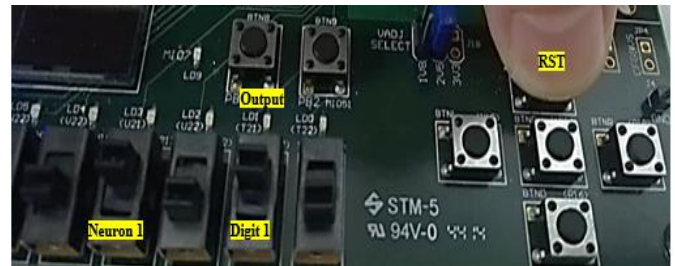


(b)

Fig. 19. Recognition for digit 0 when (a) when BTN button is pressed and (b) when RST button is pressed.



(a)



(b)

Fig. 20. Recognition for digit 1 when (a) when BTN button is pressed and (b) when RST button is pressed.

6.3 Resources Utilization and Power Consumption

Figure 21 shows the resources utilization of the system. The utilization of Lookup Table (LUT) elements are 1661. The flip-flop (FF) used for this system is 1251. Available Input/output (IO) is 200. Utilized digital signal processing (DSP) blocks is 6. Figure 22 shows the report power consumption for spiking feedforward system. The power consumption for the system is 136mW. Device static power is about 0.106W while dynamic power is 0.03W. Dynamic power can be saved by reducing the speed clocks, voltages or cutting the design activity in order to achieve the energy efficiency of the designed system. Table 1 shows the comparison with previous works by using different AI method and applications. In general, SNN is superior to CNN in power consumption. In [23], Memristive Crossbars Array (MCA) technology is used for deep SNN which is challenging in fabrication although it could achieve good power consumption. Besides that, low power consumption in [24] due to ASIC technology used compared to generic FPGA used in this work. The ASIC design is hard to be reconfigured to train the system. In short, this system shows a more favourable advantage in term of energy efficiency compared with other previous works.

Resource	Utilization	Available	Utilization %
LUT	1661	53200	3.12
LUTRAM	228	17400	1.31
FF	1251	106400	1.18
DSP	6	220	2.73
IO	18	200	9.00

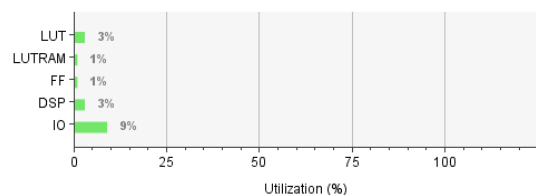


Fig. 21. Resource utilization for SNN system.

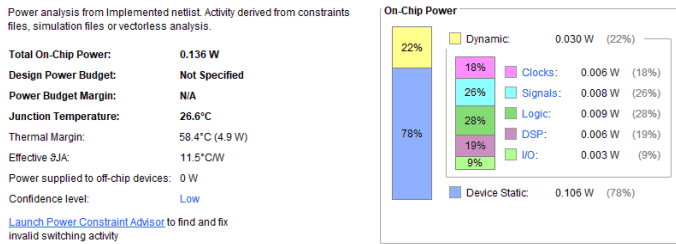


Fig. 22. Power consumption for SNN system.

TABLE 1

Comparison power consumption with previous works by using different AI method and applications.

Specifications	Power	AI Method	Technology	Application
[25]	28W	CNN	Virtex-7	Image Recognition
[26]	3.32W	CNN	Zynq-7000	Character Recognition
[27]	3.2W	SVM	Vertix-7	Image Recognition
[28]	1.69W	SVM	Zynq-7000	Melanoma Recognition
[23]	35.1mW	SNN	Memristor	Digit Recognition
[29]	1.355W	SNN	Virtex-6	Pattern Recognition
[30]	4.6W	SNN	Zynq-ZCU102	Pattern Recognition
[24]	2.8mW	SNN	ASIC	Character Recognition
This work	136mW	SNN	Zynq-7000	Digit Recognition

7 CONCLUSION

In this work, fundamental building blocks for neuromorphic computing i.e. neuron models, communication system and learning module have been presented. IZH neuron module is used to develop large-scale models of brain impulses by neural networks. AER system transmits the neurons' pulse on a chip to the appropriate destination in an array of neurons in another chip. STDP is a training system to modify the weights of all of the neural network. Energy-efficient spiking feedforward computing system has been developed and implemented to demonstrate the signal flows from neuron to classification steps. The system has been applied for hardware implementation for digit recognition. With only 136mW of power is needed to complete the task. Very low hardware resource utilization which is less than an average of 10 is observed. Neuromorphic computing demonstrates the capability of learning and efficiency in computing power, showing a promising future in realizing an autonomous cognitive system.

ACKNOWLEDGMENT

The authors acknowledge the technical and financial support by Universiti Teknikal Malaysia Melaka (UTeM).

REFERENCES

[1] N. Kasabov, Deep Learning in Spiking Neural Networks for Brain-Inspired Artificial Intelligence. 2018, doi: 10.1145/3274005.3274006.
 [2] Y.C. Wong, Y.Q. Lee, "Design and development of deep learning convolutional neural network on an field programmable gate array," J. Telecommun. Electron.

Comput. Eng., vol. 10, no. 4, pp. 25–29, 2018.

- [3] T. Mittal and R. K. Sharma, "Speech recognition using ANN and predator-inuenced civilized swarm optimization algorithm," Turkish J. Electr. Eng. Comput. Sci., vol. 24, no. 6, pp. 4790–4803, 2016, doi: 10.3906/elk-1412-19310.3906/elk-1412-193.
- [4] Y.C. Wong, L.J. Choi, S.S.S. Ranjit, H. Zhang and A.R. Syafeeza, "Deep Learning Based Racing Bib Number Detection and Recognition," Jordanian J. Comput. Inf. Technol., vol. 5, no. 3, pp. 181–194, 2019, doi: 10.5455/jjcit.71-156274772810.5455/jjcit.71-1562747728.
- [5] R. J. Vogelstein, U. Mallik, E. Culurciello, G. Cauwenberghs, and R. Etienne-Cummings, "Saliency-driven image acuity modulation on a reconfigurable silicon array of spiking neurons," Adv. Neural Inf. Process. Syst., no. May 2014, 2005.
- [6] W. Gerstner, "Spiking Neuron Models," Encycl. Neurosci., pp. 277–280, 2009, doi: 10.1016/B978-008045046-9.01405-4.1016/B978-008045046-9.01405-4.
- [7] J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov, "Spiking Neural Networks and online learning: An overview and perspectives," Neural Networks, vol. 121, pp. 88–100, 2020, doi: 10.1016/j.neunet.2019.09.004.1016/j.neunet.2019.09.004.
- [8] S. Dutta, V. Kumar, A. Shukla, N. R. Mohapatra, and U. Ganguly, "Leaky Integrate and Fire Neuron by Charge-Discharge Dynamics in Floating-Body MOSFET," Sci. Rep., vol. 7, no. 1, pp. 1–7, 2017, doi: 10.1038/s41598-017-07418-y.10.1038/s41598-017-07418-y.
- [9] F. Santamaria and J. M. Bower, "Hodgkin-Huxley Models," Encycl. Neurosci., pp. 1173–1180, 2009, doi: 10.1016/B978-008045046-9.01413-3.1016/B978-008045046-9.01413-3.
- [10] C. Zhao, W. Danesh, B. T. Wysocki, and Y. Yi, "Neuromorphic encoding system design with chaos based CMOS analog neuron," 2015 IEEE Symp. Comput. Intell. Secur. Def. Appl., 2015, doi: 10.1109/CISDA.2015.7208631.1016/CISDA.2015.7208631.
- [11] M. G. Johnson and S. Chartier, "Spike neural models (part I): The Hodgkin-Huxley model," Quant. Methods Psychol., vol. 13, no. 2, pp. 105–119, 2017, doi: 10.20982/tqmp.13.2.p105.10.20982/tqmp.13.2.p105.
- [12] E. M. Izhikevich, "Simple model of spiking neurons," IEEE Trans. Neural Networks, vol. 14, no. 6, pp. 1569–1572, 2003, doi: 10.1109/TNN.2003.820440.1016/TNN.2003.820440.
- [13] Y. Çakir, "Modeling of time delay-induced multiple synchronization behavior of interneuronal networks with the Izhikevich neuron model," Turkish J. Electr. Eng. Comput. Sci., vol. 25, no. 4, pp. 2595–2605, 2017, doi: 10.3906/elk-1606-8110.3906/elk-1606-81.
- [14] M. Mahowald, "VLSI Analogs of Neuronal Visual Processing: Thesis by," Technology, vol. 1992, no. May, 1992.
- [15] E. Culurciello and A. G. Andreou, "A Comparative Study of Access Topologies for Chip-Level Address-Event Communication Channels," IEEE Trans. Neural Networks, vol. 14, no. 5, pp. 1266–1277, 2003, doi: 10.1109/TNN.2003.816385.1016/TNN.2003.816385.
- [16] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing,"

- Nature, vol. 575, no. 7784, pp. 607–617, 2019, doi: 10.1038/s41586-019-1677-210.1038/s41586-019-1677-2.
- [17] M. Davies et al., “Loihi: A Neuromorphic Manycore Processor with On-Chip Learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018, doi: 10.1109/MM.2018.11213035910.1109/MM.2018.112130359.
- [18] F. Christophe, T. Mikkonen, V. Andalibi, K. Koskimies, and T. Laukkarinen, “Pattern recognition with Spiking Neural Networks: A simple training method,” *CEUR Workshop Proc.*, vol. 1525, pp. 296–308, 2015.
- [19] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Front. Comput. Neurosci.*, vol. 9, no. AUGUST, pp. 1–9, 2015, doi: 10.3389/fncom.2015.0009910.3389/fncom.2015.00099.
- [20] B. Ruf and M. Schmitt, “Learning Temporally Encoded Patterns in Networks of Spiking Neurons,” *Neural Process. Lett.*, vol. 5, no. 1, pp. 9–18, 1997, doi: 10.1023/A:100969700868110.1023/A:1009697008681.
- [21] A. Kasiński and F. Ponulak, “Comparison of supervised learning methods for spike time coding in spiking neural networks,” *Int. J. Appl. Math. Comput. Sci.*, vol. 16, no. 1, pp. 101–113, 2006.
- [22] E.-G. Merino Mallorquí, “Digital system for spiking neural network emulation,” *Universitat Politècnica de Catalunya*, 2017.
- [23] A. Ankit, A. Sengupta, P. Panda, and K. Roy, “RESPARC: A Reconfigurable and Energy-Efficient Architecture with Memristive Crossbars for Deep Spiking Neural Networks,” pp. 1–6, 2017, doi: 10.1145/3061639.306231110.1145/3061639.3062311.
- [24] S. Chaturvedi and A. A. Kurshid, “ASIC implementation for improved character recognition and classification using SNN model,” *Procedia Comput. Sci.*, vol. 62, no. Scse, pp. 151–158, 2015, doi: 10.1016/j.procs.2015.08.42810.1016/j.procs.2015.08.428.
- [25] Y. Duan, S. Li, R. Zhang, Q. Wang, J. Chen, and G. E. Sobelman, “Energy-Efficient Architecture for FPGA-based Deep Convolutional Neural Networks with Binary Weights,” *Int. Conf. Digit. Signal Process. DSP*, vol. 2018-Novem, pp. 1–5, 2019, doi: 10.1109/ICDSP.2018.863159610.1109/ICDSP.2018.8631596.
- [26] Gan Feng, Zuyi Hu, Song Chen, and Feng Wu, “Energy-efficient and high-throughput FPGA-based accelerator for Convolutional Neural Networks,” in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Oct. 2016, pp. 624–626, doi: 10.1109/ICSICT.2016.799899610.1109/ICSICT.2016.7998996.
- [27] O. Elgawi, A. M. Mutawa, and A. Ahmad, “Energy-Efficient Embedded Inference of SVMs on FPGA,” *Proc. IEEE Comput. Soc. Annu. Symp. VLSI, ISVLSI*, vol. 2019-July, pp. 164–168, 2019, doi: 10.1109/ISVLSI.2019.0003810.1109/ISVLSI.2019.00038.
- [28] S. Afifi, H. GholamHosseini, and R. Sinha, “A system on chip for melanoma detection using FPGA-based SVM classifier,” *Microprocess. Microsyst.*, vol. 65, pp. 57–68, 2019, doi: 10.1016/j.micpro.2018.12.00510.1016/j.micpro.2018.12.005.
- [29] Y. Li, “Energy Efficient Spiking Neuromorphic Architectures For Pattern Recognition,” no. May, 2016.
- [30] Ju, X., Fang, B., Yan, R., Xu, X., & Tang, “An FPGA Implementation of Deep Spiking Neural Networks for Low-Power and Fast Classification,” *Neural Comput.*, vol. 32, pp. 1–23, 2019, doi: 10.1162/neco_a_0124510.1162/neco_a_01245.