Spiking Versus Traditional Neural Networks for Character Recognition on FPGA Platform

Amjad J. Humaidi and Thaer Mohammad Kadhim
Department of Control Systems Engineering, University of Technology, Sina'a St., Baghdad, Iraq
601116@uotechnology.edu.iq

***Abstract*—Spiking Neural Networks (SNN) is considered the third generation of neural networks. This type of neural networks are inspired from biological nature of cortical neuron, and they (SNNs) introduced the concept of time rather than using real-valued inputs and outputs, which is a characteristic feature of Artificial Neural Network (ANN). The purpose of this research work is to design, develop and implementation of character recognition algorithm in a hardware environment (FPGA) based on Artificial Neural Networks and Spiking Neural Networks. Then, a comparison study is made, between these two generations of neural networks, regarding execution time and hardware size on the FPGA platform. Altera Cyclone-II DE2 (development and education) board has been suggested to work with for design and implementation of character learning and recognition. Coding with Verilog Language has been used for developing and synthesising the software structure of SNN and, also, to perform learning and recognition inside Altera Cyclone-II DE2 hardware. The learning method used in ANN is back propagation, while spike time dependent plasticity (STDP) has been employed for learning of SNN.**

***Index Terms*—Artificial Neural Networks (ANN); Spiking Neural Network (SNN); Spike Time-Dependent Plasticity (STDP); Character Recognition; FPGA.**

I. INTRODUCTION

Artificial neural networks (ANNs) are mathematical-based structures which try to mimic the computational ability of the brain and to develop human-like intelligent systems. ANN is based on neurons which characterised by real-valued inputs and outputs [1].

Spiking Neural Networks (SNN) is considered the third generation of neural networks. This type of neural networks is inspired by the biological nature of the brain neurons. The information transmission among brain cortical neurons is performed by action potentials (spikes). This spiking nature of neurons has been reported by Adrian in 1920s. Biological neurons send out spikes or short energy pulses as signals when they receive sufficient input from other neurons. Therefore, spiking neural networks (SNN) is more biologically plausible than classical ones; however, the calculation will account for the concept of spikes rather than real values [2].

Each spiking neuron consists of three computational stages; in the first stage, a sum for all of the neuron's input current is performed, while in the second stage of computation the summation of input current for each neuron will be integrated over time. At the third stage of computation, if the potential difference value between the two sides of the cellular membrane of the neuron (neuron's membrane potential) reaches above a specified threshold, a spike is fired, and the value of membrane potential goes back to a reset value [3]. The schematic diagram describing the model of Spiking Neural Networks is illustrated in Figure 1.

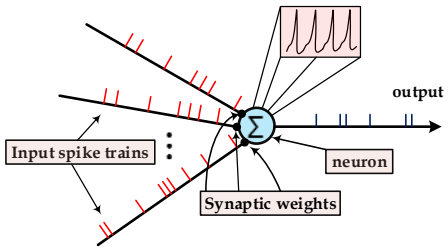

Figure 1: A schematic diagram of SNN neuron model

Performing neural computation is the primary aim of spiking neural networks. The problem is how to express the computation-pertinent quantities in terms of spikes, which are responsible for communicating spiking neurons. In neuroscience, the neural coding is still a premature topic of research, and it has not yet been solved. However, neural information encoding such as binary coding, rate coding, latency coding, fully temporal encoding, predictive spike-encoding and probabilistic spike-coding has been reported as the main encoding techniques used in SNN [3]. The binary coding has been drawn from the observation that physiological neurons tend to transmit pulses at the time they subjected to input sensory stimuli such a light or external electrical stimulus. In binary coding, the neuron is either active or silent. In case of active operation, the neuron emits one or several spikes during the time interval of this mode. Such encoding has been employed in the implementation of current spiking neural network for interpreting the output spike trains of SNN and hence classifying the input spike trains [4, 5]. In the present work, this coding has been adopted in the design and implementation of SNN for the character recognition.

There are many models of SNN neuron and the most commonly used models in the field of pattern recognition are Hodgkin Huxley Model, Izhikevich Model, Spike Response Model, Thorpe Model, Kasabov Model [6]. The model proposed by Izhikevich combines both the biologically plausible dynamics of the Hodgkin-Huxley model and the efficient computational integrate-and-fire model [7]. This neuron mode the l will be considered as the essential element of SNN for the present work.

It has been proved that the weights between a presynaptic neuron and a postsynaptic neuron do not have constant values; instead, they have to change values, which in turn would influence on the amplitude of emitted spikes [8]. The procedure of the weight update is called the learning process,

and it can be divided into two categories: supervised and unsupervised learning.

Spike Proportional, Remote Supervised and Modified Remote Supervised methods are examples of supervised learning, while Hebbian learning algorithm and Spike-Timing-Dependent Plasticity (STDP) are classified as unsupervised learning. STDP is adopted here for learning SNN.

Some of the researchers have been explored the subject of character recognition based on spike neural network. Other researchers utilised Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC) hardware to implement the other structures of the neural network. The following works interview some of the relevant topics:

Mariam Bokeria has introduced character recognition utilising the spiking neural network. The work has used the Izhikevich model, which had been implemented within the MATLAB environment [9]. Nikolai Jefimov has designed and implemented image recognition based on deep neural network structure. The recognition is dedicated to recognising Latin letters with dimensions of 3×3 and 5×7 pixels. The design, implementation of image recognition algorithm and proving of learning algorithms for the deep neural network have been performed using MATLAB [10]. Yuan Jing has used FPGA to implement the Optical Character Recognition (OCR) system. The work presented tradition Artificial Neural Network (ANN) for training and recognition [11]. S. Chaturvedi, A. A. Khurshid and Nirja Karlewar has designed and implemented SNN-based character recognition using ASIC (Application-Specific Integrated Circuit). The researchers have suggested the Izhikevich model for network neurons and, also, used Spike-Time-Dependent Plasticity (STDP) algorithm as a training rule [12]. Marzieh Moradi et al. have focused on FPGA implementation of recognition for Farsi handwritten digit system using traditional neural network [13]. Ankur Gupta and Lyle Long have introduced spiking neural networks for image recognition application based on pulse coding. The proposed model could successfully recognise a set of 48 characters with 3×5 pixel size. For recognition purpose, the work made use of active dendrite and dynamic synapse model; while the spike timing-based plasticity (STDP) has been used for the training process. Both training and recognition processes have been implemented using MATLAB [14].

No previous study has presented the FPGA-based design and implementation of SNN for character learning and recognition; especially developing the algorithm with Verilog code. Also, there is no previous study had made a comparison study between SNN and ANN on the platform of FPGA. In the present work, a comparative study between two generations of neural networks has been established for character recognition using FPGA. One can summarise the primary objectives of the work as follows;

i. Design, development and implementation of the character recognition algorithm based on spiking neural network and traditional neural network using FPGA hardware (Altera Cyclone-II DE2). SNN and ANN are synthesised and coded for learning and recognition of characters using Verilog code.
ii. The Spike-Time-Dependent Plasticity learning method is used in SNN, while Supervised Back-propagation algorithm is suggested as a learning method for ANN.
iii. A comparison is made between SNN and a previous study based on ANN [11]. Both structures have been designed and implemented in the environment of FPGA. The comparison is assessed regarding hardware size occupation and speed of character recognition.

II. Izhikevic Model

This model has been proposed by Izhikevich to combine both the biologically plausible dynamics of Hodgkin-Huxley model and the efficient computational integrate-and-fire model. The Izhikevich model is characterised by second-order differential equations as indicated by the following equations [15, 16],

$$dV/dt = 0.04\,V^2 + 5\,V + 140 - u + I \quad (1)$$

$$du/dt = a\,(b\,V - u) \quad (2)$$

where the variable v describes the voltage potential of the membrane, and the variable u describes the membrane recovery, activation and deactivation of potassium currents sodium currents, respectively. The design parameter a represents the time scale of the membrane recovery u, where high values of a leads to faster recovery. The parameter b stands for the sensitivity of the membrane recovery u in response to sub-threshold fluctuations of the membrane potential V. The variable I refers to the total synaptic current. The portion $0.04\mathrm{V}^2 + 5\mathrm{V} + 140$ has been acquired by fitting the spike initiation dynamics of cortical neurons such that the unit of variable V correspond to mV such that the time unit correspond to ms.

Once the voltage V exceeds a threshold value of $30mV$, both V and u are reset to the specified value. This condition can be formulated as;

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

where c describes the reset value of the membrane potential V, resulting from fast high-threshold conductance of potassium (K), and d depicts the after-spike reset value of the recovery variable u due to the high-threshold conductance of sodium (Na) and potassium (K). By suitable adjustment of fixed parameters (a, b, c, and d), Izhikevich has shown that his model can produce the responses of most types of neurons seen in biological tests.

III. Spike Time-Dependent Plasticity (STDP)

In STDP, the weight changes Δw of a particular synaptic connection is assumed to be related to the following expression [8];

$$\Delta w = \begin{cases} +A_+\, e^{(-s/\tau_+)},\ if\ s > 0 \\ -A_-\, e^{(s/\tau_-)}\ ,\ if\ s \leq 0 \end{cases} \quad (4)$$

where $+A_+$, $-A_-$, τ_+ and τ_- are constants with positive real values. The variable s represents the delay between the presynaptic and postsynaptic firings and it is described by $= t_{pre} - t_{post}$.

In this learning method, the change in weights happens if the pre-synaptic spike occurs before the post-synaptic spike. In this case, the synapse is strengthened; otherwise, it is weakened. This will guarantee that the occurrence of the next

post-synaptic spike (following the present training iteration) will be closer to the pre-synaptic spike. The weights will be changed according to the relaxation rule [8]:

$$\Delta w_{new} = \begin{cases} w_{old} + \eta\, \Delta w(w_{max} - w_{old}), & if\ \Delta w \geq 0 \\ w_{old} + \eta\, \Delta w(w_{old} - w_{min}), & if\ \Delta w < 0 \end{cases} \quad (5)$$

Here, η is the learning rate. For excitatory synapses $w_{min} = 0$ and $w_{max} = 1$, whereas for inhibitory synapses $w_{min} = -1$ and $w_{max} = 0$. If there is no pre-synaptic spike, the weight decays with a rate decay η_{decay}. This is similar to the decay of η_{decay} synaptic strength in real neurons.

IV. Character Recognition Algorithm

In what follows, the structures of SNN and ANN are described. The suggested networks have been trained to recognise 48 different input images corresponding to 48 different characters as indicated in Figure2. These 48 images correspond to upper case (A-Z) letters, eight Greek letters, ten numerals (0-9) and four symbols.

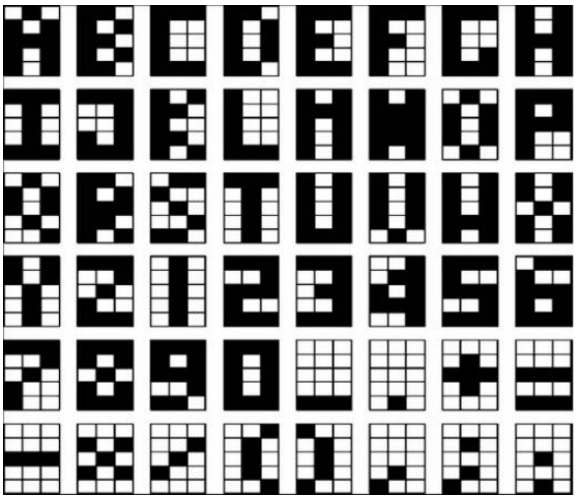

Figure 2: Training images

A. Spiking Neural Network

In this algorithm, a two-layer spiking neural network structure based on the Izhikevich model has been used for character recognition [16]. The first layer of the network works as input neurons, while the second layer represents the output neurons.

The characters corresponding to input images are given to the first layer of neurons (designated as level 1), such that each pixel of the image is presented to its corresponding individual input neuron [16]. Therefore, the number of pixels in the input image is equivalent to the number of neurons in level 1. Moreover, the number of training images is equal to a number of neurons at the second layer, which is referred to as level 2. This is because each neuron of the second layer (level 2) has been encoded to fire only when it recognises its corresponding image. Each neuron of level 1 is linked to every neuron of level 2. A prototype of this neural network structure is depicted in Figure 3.

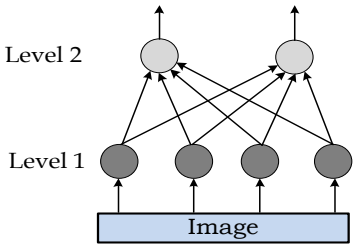

Figure 3: The Neural Structure and Connection between the Neurons of Two Levels.

Membrane potential V is evaluated by input current *I* of each neuron. If this potential of membrane reaches a certain threshold level through the cycle, the neuron is promptly fired. Concerning the neuron of level 1, the input current I of a particular neuron is set to zero if its corresponding pixel in the input image is "off ". However, a constant current is fed to the input neuron if its corresponding input pixel of the image is "on".

On the other hand, the input current to a certain neuron at level 2 is equal to the sum of all individual currents coming from the connected neurons at level 1. The expression of input current I (j) for neuron j at level 2 is described by Equation (1).

$$I(j) = \sum_i w(i,j)\ \ f(i) \quad (6)$$

where *w* is a weight matrix whose entries w(i,j) is the input weight from neuron i (at level 1) to neuron j (at level 2). The function f(i) shown in Equation (4) is described as a firing vector whose value is equal to zero if the i^{th} neuron of level 1 does not fire, and it is equal to one if the i^{th} level 1 neuron does fire. Training process is responsible for determining the entries of the weight matrix w immediately after the process of driving the set of training images sequentially to the input neurons. The training process would produce a weight matrix, which cab utilized to evaluate the input current of each output neurons.

During the recognition phase, an input image is delivered to neurons of the level 1, and there is one output neuron will be fired after passing a certain number of cycles; indicating that the input image is identified. Based on the input image, the neurons at level 1 are evaluated during each cycle such that the firing vector is updated to specify which neurons are fired at that cycle. At the same cycle, the firing vector resulting from the last cycle is utilised to determine the value of input current I coming into each neuron at level 2. Based on the input current values of neurons at level 2, their membrane potentials *V* can be determined.

B. Artificial Neural Network

An artificial neural network consists of the perceptron, organised into three layers: an input layer, a hidden layer, and an output layer. Perceptrons (j) in a given layer are individually connected to each of the perceptrons (i) from the previous layer, with a weight of w (i, j) placed on the connection. Since each perceptron is a nonlinear function acted on the weighted sum of outputs from other perceptrons, by changing the weight on each connection in an ANN, any smooth function can be approximated. Figure 4 shows the structure of the artificial neural network used for comparison to SNN [11].

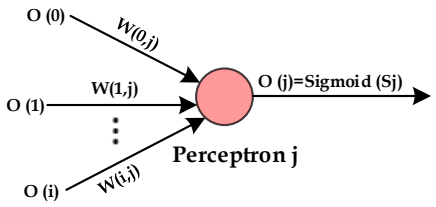

Figure 4: Structure of a perceptron

V. FPGA Implementation of English Character Recognition Using Spiking Neural Network

It is preferably in the design of SNN to be more generic; that is, it is easy to reconfigure the number of Izhikevich neurons per layer such as to meet the demand of a particular application. The values assigned individually to all connection weights are stored in a matrix of size (I×O) array, where I and O are defined as the number of Izhikevich neurons in the input layer and output layer, respectively.

In the present work, an SNN has been designed and written in Verilog code and, then, implemented on an Altera DE2 board to perform Character recognition on a 5×3 binary grid. The inputs into the Pattern Recognizer are the first 15 switches on the Altera DE2 board. Each group of 3 is one row of the grid as indicated in Figure5. In the present work, only four characters (’A’, ’B’, ’C’, and ’D’) have been considered.

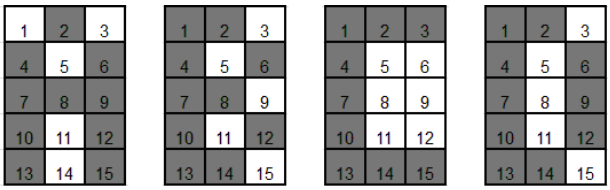

Figure 5: Input switches character

The top-level design entity contains several modules. The most important of these are Iz_neuron, synapse, LCD, Seven segments. Register Transfer Level (RTL) for top-level design is indicated in Figure6. Figure 7 shows the Iz_neuron model, which simulate the Izhikevich neuron in Verilog. It is clear from the figure that the module has three inputs (current, clock, and reset) and two outputs (voltage and spike indicate).

Register Transfer Level (RTL) of Synapse is illustrated in Figure8. The module collects the output of the neuron from the input layer and produces current input to the neuron in the output layer. Figure9 depicts Register Transfer Level (RTL) of LCD control, and logical control for display content. Figure 10 depicts Register Transfer Level (RTL) of an LCD model, which comprises an LCD communication module. Figure 11 depicts Register Transfer Level (RTL) of STDP module.

In the FPGA implementation, the four input patterns of characters are configured using toggle switches implanted on the FPGA board. During hardware implementation, Verilog code corresponding to spike neural network is processed for recognising the presented characters. For this case, the elements of network structure consist of 15 input neurons and 12 output neurons. Figure 12, Figure13, Figure14, and Figure15 show the FPGA responses for the character 'A', 'B', 'C', and 'D', respectively.

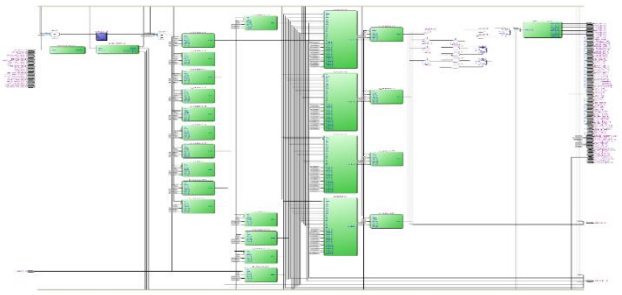

Figure 6: RTL for top-level design

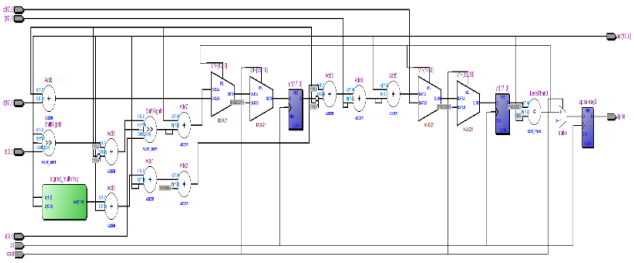

Figure 7: RTL for Iz_neuron model

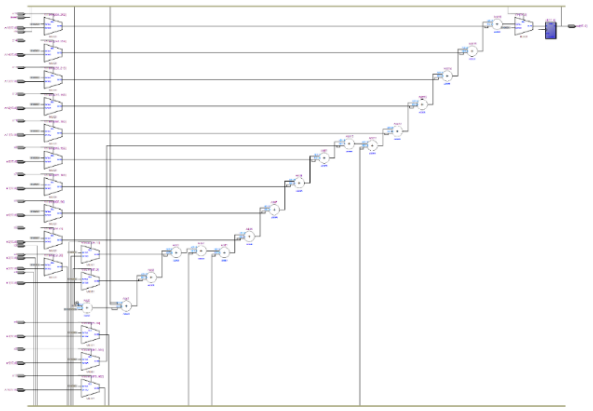

Figure 8: RTL for Synapse

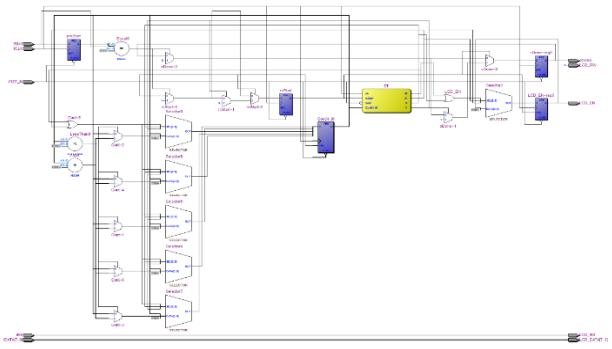

Figure 9: RTL for LCD controller

Figure 10: RTL for LCD

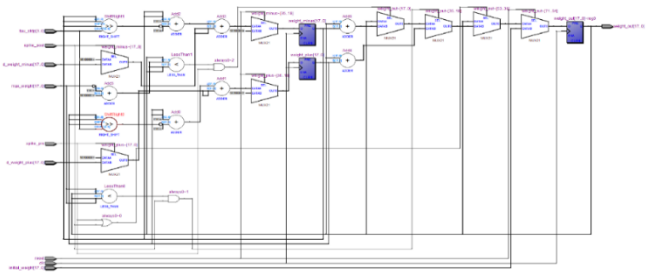

Figure 11: RTL for STDP module

Table 1 shows the hardware usage for this work in terms of logic and memory element.

Table 1
Hardware Utilization of Altera Cyclone-II DE2

Components	Used	Available	Utilization
Total logic element	2,785	33,216	14%
Total combinational functions	4,607	33,216	14%
Dedicated logic registers	639	33,216	3%
Total pins	429	475	90%
Total memory bits	0	483,480	0%
Embedded multiplier 9-bit elements	30	70	54%
Total PLLs	0	4	0%

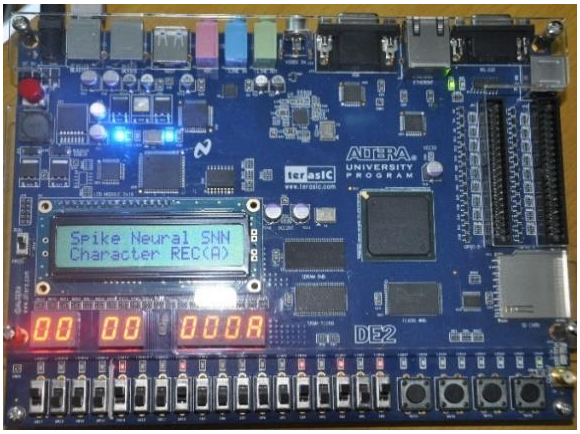

Figure 12: FPGA response for character 'A'

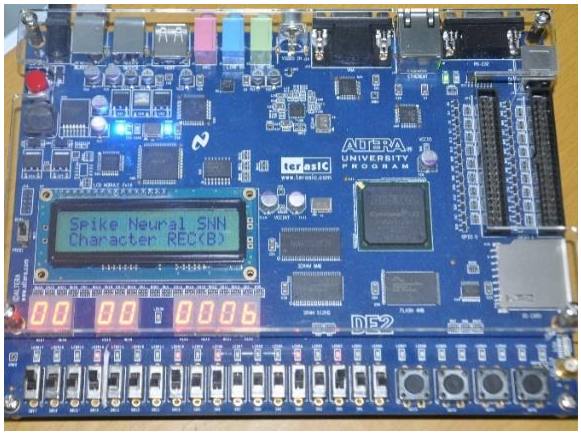

Figure 13: FPGA response for character 'B'

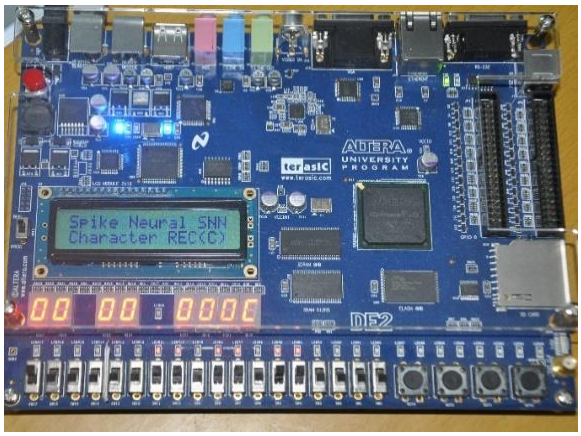

Figure 14: FPGA response for character 'C'

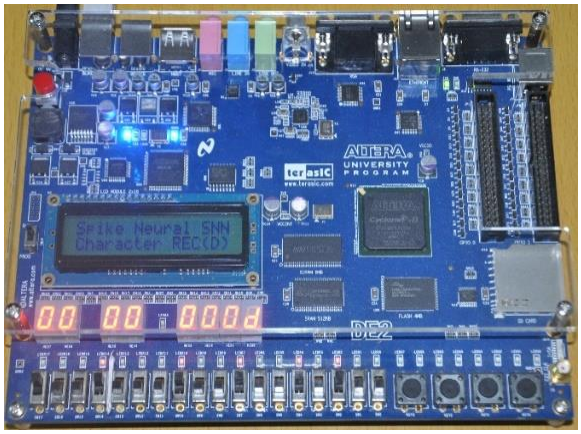

Figure 15: FPGA response for character 'D'

VI. FPGA Implementation of Character Recognition Based on Traditional Neural Network

As has been mentioned earlier, the traditional neural network consists of the perceptron, organised into three layers: an input layer, a hidden layer, and an output layer. The number of neuron in input layer equal the number of pixels in the image. The network has been trained using supervised backpropagation algorithm.

In the FPGA implementation, the four input patterns of characters are configured using toggle switches implanted on the FPGA board. Figure 16 to 19 show the FPGA responses for character 'A', 'C', 'D', and 'F', respectively.

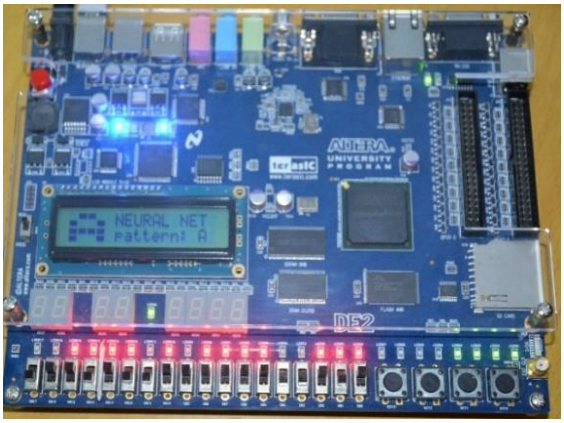

Figure 16: FPGA response for character 'A'

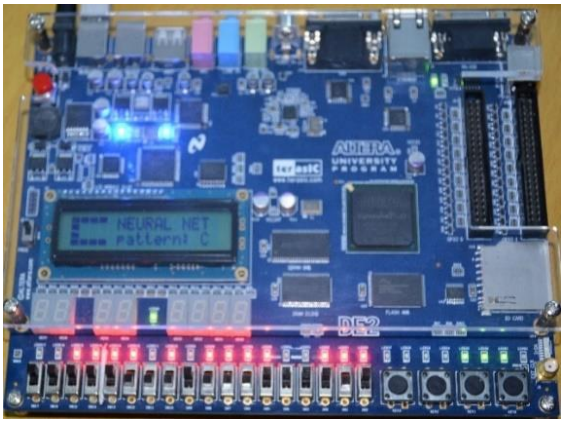

Figure 17: FPGA response for character 'C'

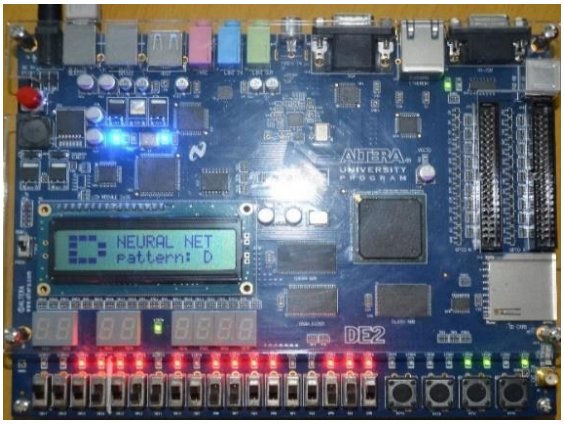

Figure 18: FPGA response for character 'D'

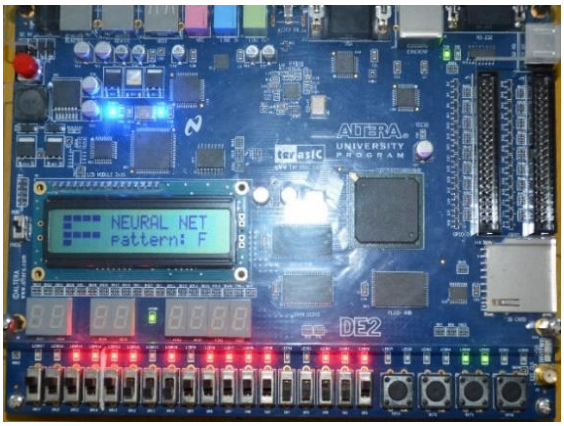

Figure 19: FPGA response for character 'F'

VII. Traditional Neural Networks Versus Spiking Neural Networks Based on FPGA

In this section, the comparison is made between the second and third generation of neural networks for character recognition algorithm. The design platform is based on DE2 educational board (from Altera), and character recognition algorithm is implemented using database binary grid for each character. The characteristics of each type of neural structure are described by the following:

i. The traditional artificial neural network consists of three-layer feed-forward with a perceptron neuron. The supervised learning method used in this type is based on backpropagation
ii. In spiking neural network, two layers feed forward with Izhikevich neuron has been used. Spike Time-Dependent plasticity (STDP) implements the unsupervised learning method.

Table 2 shows the total time required for recognising the character in traditional and spiking neural networks using a stopwatch. The stopwatch timer is a software program designed and developed based on Verilog code. It activates a counter when detecting a sequence change of binary code represented by onboard switches, which are fed to SNN network. The counter is deactivated when a change in LCD output occurs; this indicates the end of counting or reaching the stop condition.

Table 3 and 4 show the hardware usage for traditional NN and spike NN, respectively.

Table 2
Time for Recognition of SNN and ANN based On FPGA

Neural network	Time for recognition
Traditional neural network	36,123 sec
Spiking neural network	0.804 sec

Table 3
Hardware Utilization of the Altera Cyclone-II DE2 for Spiking Neural Network

Components	Used	Available	Utilization
Total logic element	4,588	33,216	14%
Total combinational functions	4,588	33,216	14%
Dedicated logic registers	859	33,216	3%
Total pins	429	475	90%
Total memory bits	0	483,480	0%
Embedded multiplier 9-bit elements	38	70	54%

Table 4
Hardware Utilization of the Altera Cyclone-II DE2 for Traditional Neural Network

Components	Used	Available	Utilization
Total logic element	11,388	33,216	34%
Total combinational functions	10,975	33,216	33%
Dedicated logic registers	5,876	33,216	18%
Total pins	423	475	89%
Total memory bits	4,956	483,480	1%
Embedded multiplier 9-bit elements	54	70	77%

VIII. CONCLUSION

A comparison study is made between SNN and ANN to show the main differences in performance. However, based on hardware results, the following conclusions can be drawn:

i. In character recognition, the experimental results show that Spiking neural network is faster than traditional neural network by about 44.92 %.

ii. It is clear from TABLE III and TABLE IV that the number of logic elements dedicated to SNN by Altera DE2 is equal to 4,588 elements while there are 11,388 logic elements assigned to implement the Traditional Neural network. Also, SNN and ANN utilise 38 and 54 elements of 9-bit embedded multipliers, respectively. Therefore, one concludes that SNN occupies a smaller size, regarding logic element than that occupied by ANN.

iii. The spike neural network consists of neurons that are dynamically described by a set of differential equations. On the other hand, the traditional artificial neural network is characterised by neurons which are described by static map (linear or nonlinear) between their input and outputs. Therefore, one can conclude that SNN is more computationally efficient than ANN.

iv. The structure of Spiking neural network is generally smaller in size than ANN. In the present work, the structure of SNN composed of two layers, while there are three layers for ANN. However, SNN could successfully and efficiently recognise the characters.

v. Throughout the design of embedded system code (Verilog) for learning and recognition using SNN and ANN, it is found that the algorithm based on SNN is easy to design and implement than that algorithm based on ANN.

REFERENCES

[1] G.Q. Bi and M.M Poo, "Synaptic modification in cultured hippocampal neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type," *J. Neuroscience*, vol. 18, pp. 10462-10472, Dec.1998.

[2] E. Adrian, *the Basis of Sensation: The Action of the Sense Organs*. W. W. Norton, New York, 1928.

[3] Sergio Davies, "Learning in Spiking Neural Networks," Ph.D., Dissertation, University of Manchester, 2012.

[4] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nature neuroscience*, vol. 9, no. 4, pp. 420–428, 2006.

[5] R. Urbanczik and W. Senn, "A gradient learning rule for the tempotron," *Neural Computation*, vol.21, pp. 340–352, 2009.

[6] Nikola Kasabov, " To spike or not to spike: A probabilistic spiking neuron model," Neural Network, vol.23, pp.16-19, 2010.

[7] Izhikevich, E. M., "Simple Model of Spiking Neurons," *IEEE Transactions on neural networks*, vol.14, no. 6, pp.1569-1572, 2003.

[8] Andrzej Kasinski, Filp Ponulak, "Comparison of Supervised Learning Methods for Spike Time Coding in Spiking Neural Networks," *International Journal Applied Mathematics Computer Science*, Vol. 16 (1), no. 1, pp.101-113, 2006.

[9] Mariam Bokeria, "Character Recognition with Spiking Neural Networks", M.Sc. Thesis, Tallinn University of Technology, Tallinn, Estonia, 2017.

[10] Nikolai Jefimov, "Image Recognition by Spiking Neural Networks", M.Sc. Thesis, Tallinn University of Technology, Tallinn, Estonia, 2017.

[11] Yuan Jing, "An Efficient FPGA Implementation of Optical Character Recognition System for License Plate Recognition", M.Sc. Thesis, University of Windsor, Ontario, Canada, 2016.

[12] S. Chaturvedi, A. A. Khurshid, and Nirja Karlewar, "ASIC Implementation for Improved Character Recognition and Classification using SNN Model", *International Journal of Software and Web Sciences*, Vol. 14, no. 1, pp. 13-17, 2015.

[13] M. Moradi, M. A. Pourmina, and F. Razzazi, "FPGA-Based Farsi Handwritten Digit Recognition System", *International Journal of Simulation Systems Science & Technology*, vol. 11, pp. 17-22, 2010.

[14] Ankur Gupta, Lyle Long, "Character Recognition using Spiking Neural Networks," *International Joint Conference on Neural Networks*, Orlando, Florida, USA, August 12-17, 2007.

[15] Izhikevich, E. M., "Simple Model of Spiking Neurons," *IEEE Transactions on neural networks*, vol.14, no. 6, pp.1569- 1572, 2003.

[16] Amjad J. Humaidi, Thaer M. Kadhim, "Recognition of English Characters Using Spiking Neural Networks," *International Journal of Engineering and Technology*, Vol. 9, no. 5, pp.3494-3503 Oct-Nov. 2017.