ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

Simulation and Characterization of Neuron Cell Behaviour in FPGA

[1] J. Meribah Jasmine, [2] N. Habibunnisha, [3] Dr. D. Nedumaran
[1][2][3] Department of Central Instrumentation & Service Laboratory,
University of Madras, Guindy, Chennai, TN, India

***Abstract* - This paper details the code development, implementation and testing of basic behaviour and characterization of the biological neuron model in FPGA environment. In this work, the real behavior and function of biological human neuron was studied and implemented as an artificial neuron in VHDL environment. To develop the biological neuron model, Perceptron algorithm was employed. The artificial biological neuron contains sub blocks such as timer, alpha, addition & comparison and inhibit blocks. Initially, the sub blocks were developed individually and combined together according to the biological neuron model design. Simulation codes for the artificial neuron model were developed using VHDL language. Finally, the developed artificial neuron model was implemented on ANVYL SPARTAN-6 XC6SLX45 CSG484 package FPGA kit and the outputs were exhibited as simulation output in the PC monitor as well as in the LED indicators of the Anvyl Board. This work helps to learn the strategies of developing codes for neuronal behaviour study using FPGA and VHDL language, which will be extended to advanced applications like computational neuroscience and artificial intelligence.**

Keywords - biological neuron; perceptron; action potential; Xililnx ISE; FPGA.

I. INTRODUCTION

The brain is a highly complex system comprises of a large quantity of basic constitutional units called neurons, which are interconnected through synapses [1]. The research endevours on the dynamical behaviors of a single neuron is the fundamental approach to investigate the functional complex neuronal system. But, the neuroscientists proposed some specific models according to the analysis of experimental observations. The dynamical behaviors of a single neuron can be modeled as a set of mathematical differential equations [2]. The investigation of neuron structures is a difficult and complex task and a large number of different approaches have been used to model the nervous system. To investigate the functioning of the brain, it is necessary to perform relevant simulation and implementation of a large-scale network for imitating the real brain [1]. Especially, the neuron firing results in a response called spike and the spike timing is the essential element in neural information processing system, due to the timing between the spikes that carries the neuronal information [2]. A neuron consist of three main parts such as soma, dendrites, and axon. The soma integrates the input and synthesis the neurotransmitter signal. The dendrites looks like a tree like structure, which receive information from sensory organs or from the other neurons. The axon is a fine long fibers originated from neuron bodies, which transmits the output of neural signals to the synapsis [1].For realization of spike trains or spike-train-timing, the behaviors of neuron's spike-based models are considered. However, spike-based models such as Izhikevich, the Integrate-Fire (IF), and Leaky-Integrate-Fire (LIF) models cannot illustrate the dynamic mechanisms physiologically. There are three approaches to consider for the accurate modeling of electrophysiological activity of neurons. Here, we have used integrate fire model. In this model, the neuron receives many spikes from other neurons randomly and therefore the action potential fluctuates as a function of time. Here, the spikes are rare events triggered at threshold. Below the threshold level, it is called as sub-threshold region. In the sub-threshold region, we can find the fluctuation of membrane potential [2]. Another important thing need to consider in modelling the neuron is the spike train. The neuron signals are nothing but a very small electrical pulse, which is otherwise called as action potential or spikes. The amplitude of these spikes is normally in the range of about 100 mV and the duration is about 1-2 ms. The continuous action potential emitted by the single neuron is called spike-train. Since all the spikes looks similar and all impulses will give the same information but practically this is not the case. Hence, it is assumed that these spikes does not carry any information rather the number of spikes and the timing difference between two or more adjacent spikes having the information. The spikes of action potential are well separated and even for very strong inputs. The minimum distance between the spikes gives the absolute refractory period of neuron.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

II. RELATED WORKS

Hines and Carnevale developed a discrete event model for the integration and firing of responses from many cells [3]. The models were simulated with fixed time step or with variable time step method in which the order of response magnitude is more faster than the biological neuron. Bailey et al. developed a neuron libraray for modelling various nervous systems similar to mammalian brain structures in VHDL environment, which was found to be similar to that of the Message Based Event Driven (MBED) model [4]. Borgesa et. al. proposed spike timing-dependent plasticity of neuron and its network behavioural changes due to environment based on the Hebbian rules and random distribution method, respectively. They analysed the synchronization and desynchronization effect with repect to the input level and probability of connections, which exhibited that the transition of synchronization depends on the network architecture and perturbation level [5]. Frnando Pezez-Pena, et. al. have developed the inter-spike-intervals exponential and gama distributions for studying the neuron firing rate using statistical hypothesis [6].

III. METHODOLOGY

A. Perceptron algorithm

The perceptron is a simple neuron model as shown in Figure 1, which takes spike patterns as input and the ouput is coded as synaptic-weighted vectors of inputs. The output *Y* is determined by

$$Y = f(w) = f(\bar{s}.\bar{x}) = f\left(\sum_{j=1}^{n+1} s_j x_j\right) = f\left(\sum_{j=1}^{n} s_j x_j - \alpha\right) \quad (1)$$

where *w* is the weighted sum of inputs, (i. e., dot product of weight and input vectors) and f is the activation function. If there are *n* inputs to the perceptron, then the input is given as *(n+1)* that will be fixed to -1 for the associated weight of $s_{n+1}=\alpha$ Here, α is called the *excitation threshold* [7].

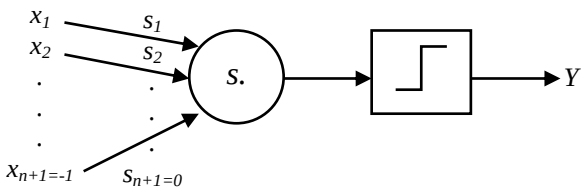


Fig.1. Schematic illustration of the perceptron mode

B. Proposed VHDL neuron model

For digital implementation of neuron model, the behaviours are experessed in numerical form in order to directly implement the model in digital systems or Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL). Since VHDL is a concurrent language, the neuron behaviour can be used to simulate in a parallel fashion.

Further, the dendrites input changes dynamically whenever its input sensitivity changes, which can be accommodated very easily in VHDL simulation. Generally, the simulation objects spend more time in waiting for the input, which can be very well processed in VHDL simulation.

In this work, a simple neuronal system was developed, which comprises of a neuron with three dendrites as inputs. Further, the numbers of simultaneous spikes considered to excite a neuron was fixed as 50 in a short period of time. The weights associated with the dendrites can be varied to fire the neuron to simulate the expected output [8, 9].

C. Neuron model with three dendrite input

In artificial neuronal networks, the learning process of the neuron is represented by how weights of the connections between the neurons are taken for processing. The learning process will make some changes in these connections and subsequently, the neuronal network learns and modifies the values of the weights of the network [5]. The usage of weights is and important factor, since weights are associated with the synapses and this can increase or decrease the signals that arrive to the synapse. In this work, three variable weights are given as the inputs. Every weight is associated with a dendrite input. The proposed system comprised of the following input/output combinations:

- 3 inputs referred to the dendrites from others neurons.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

- 1output referred to the axon.

When the output is fired, the output generates a pulse, which is propagated to others neuron through the axon. Subsequently, a clock signal is required to synchronize the digital signals with the global system.

D. Dendrite input with weight

The proposed neuron model comprises of are three inputs with the associated weights and a clock signal for synchronization. The neuron receives these weighted inputs and compares with the threshold level for producing the spike or action potential. If the acton potential is above the threshold, it will generate a output pulse and that will pass through the axon.

E. Functions of artificial neuron

The dendrite action is considered as the biopotential generator. When an action potential is received, a pulse starts to increase and all the individual potential are added and compared with the threshold. Figure 2 shows the functions of a neuron for simulation purpose.

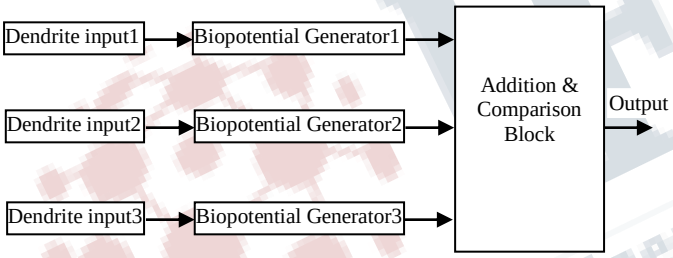


Fig 2. Functions of Artificial Neuron

The functional processor unit of a neuron is used to generate a potential according to the strength of the spikes. The functional process unit also evaluates the threshold potential to generate an output potential.

F. Real EPSP

In a real-neuron, when a spike arrives inside the soma an excitatory postsynaptic potential (EPSP) is generated. The EPSP gives an exponential function, which can be estimated using the alpha function $f(x) = x * e^{-x}$. The simulation of real EPSP is shown in Figure 3.Therefore each dendrite function can be simulated using a potential generator that comprises of a timer and a alpha function as shown in Figure 4.

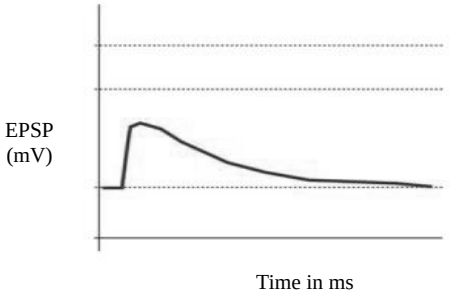


Fig 3. Simulation of real EPSP

G. Splitted potential generator

Due to the weight of the dendrite input, the potential obtained by the alpha function will increase or decrease. This decides the synaptic connection. Therefore, the potential generator is redesigned as splitted potential generator as shown in Figure 4.

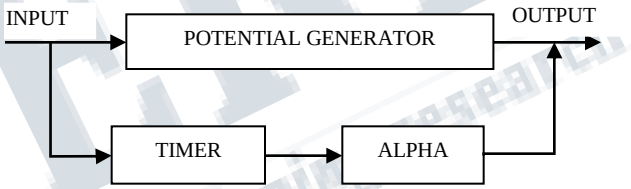


Fig 4. Splitted potential generators

The timer and the alpha block will continuously make the addition of all the input potentials. After the addition process, it compares the result with the threshold value. The critical value (threshold value) for firing a neuron is about 20 mV to 30 mV. Here, we defined the threshold value as 20 mV. If the result is positive, then the neuron will be in an exited state. Otherwise, if the summation of all the input potential is lesser than the threshold value, then the output of the neuron is '0'. For implementation of the threshold function in a neuron model, an inhibit input is incorporated as shown in Figure 5.

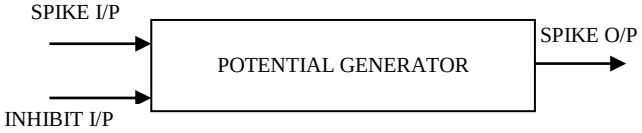


Fig 5. Potential generator with inhibit signal

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

H. Timer block

When the system detects an input pulse (the rise edge), then the timer starts the count. It also have an enable input and when it is active, the timer stops the count, and the count resets so that this does not allow the next input till the completion of the previous process input. When the enable signal is inactive, the timer counts again when it receives the immediate rising edge of the input. The complete timer block is shown in Fig. 6.

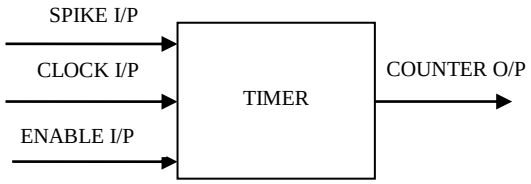


Fig 6. Timer block

I. Addition/Comparison block

The spike potentials generated from the alpha blocks at every instant of time are added and compared with the threshold potentials of 20 mV. This addition and comparision are taken during the reception of the signal from the respective splitted potential generators. If the value of the cumulated spike potential is greater than the threshold, the addition and comparison block generates an action potential, otherwise there is no output. The schematic of the comparison block is shown in Fig. 7.

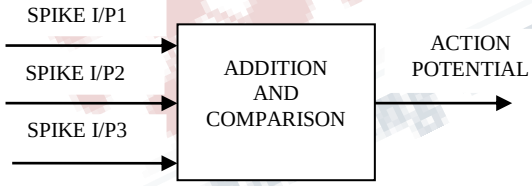


Fig 7. Addition and Comparison block

J. Inhibit block

To simulate the refractory period of a neuron, the model has an inhibit input which decides/rejects the entry of a input to the dendrite based on the refractory response of the neuron. Once the neuron is excited, it unable to generate another spike during the period of excitation time, called the refractory period. The refractory period is about 2 ms. To implement this neuron activity digitally, the inhibit block is used. The inhibit block acts like a timer. When neurons fired, it sends a signal to the potential generators and to the inhibit block. At the same time, it starts a counter to count for the 2 ms refractory period. During the 2 ms period, the counter keeps sending the inhibit signal. When the counter reaches the 2 ms period, it disables the inhibit signal. Now, the potential generator is able to recive the spike input and accordingly it generates the spike potential output. The inhibit block is shown in Fig. 8.

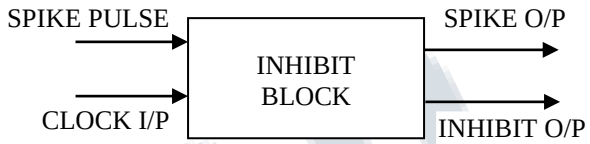


Fig 8. Inhibit block

K. The complete function of neuron model

The complete artificial neuron model is shown in Fig 9. The model comprises of three dendrite inputs with the respective weights *S1, S2 and S3*. The clock signal plays an important role in synchronizing the counter of the timer as well as the inhibit blocks. Further, the three alpha blocks and an addition and comparison blocks are used to generate the spike discharge output and action potential output, respectively. Finally, the inhibit block was used to maintain the refractive period of the neuron during the period of action potential response.

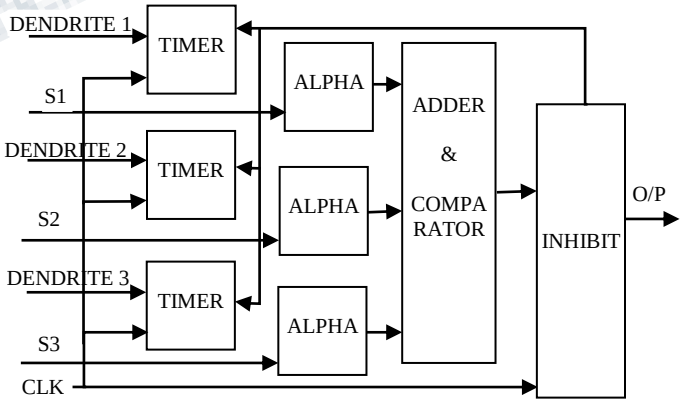


Fig 9. Artificial Neuron Model

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

IV. SIMULATION RESULTS AND DISCUSSION

The functional block diagram shown in Fig. 9 was developed and compiled using Xilinx ISE (Integrated Synthesis Environment) 14.7 webpack. This is a free version software tool of Xilinx for synthesis and analysis of HDL designs, which enables the developers to synthesis their designs, examine RTL digrams and perform timing analysis for FPGA based designs [10]. The simulation and the reponse of the individual blocks are described in the following sub-sections.

L. Timer block

The simulated timer block output is shwon in Fig. 10. Here, the timer counts like 0.0, 0.1, 0.2… and the counter was used for checking the new pulse arrived at the input terminal. The reset pin was used to indicate the process status of the present input and the neuron's waiting phase for the next input.

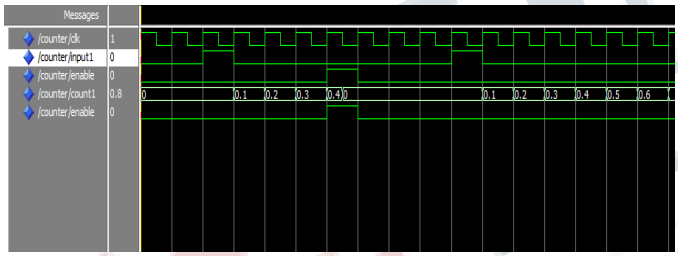


Fig. 10 Output for Timer block

From the Fig. 10, it was observed that till the count of 0.4, the the first input was processed. After the completion of this process, the counter was automatically reseted, which indicated that the current process was finished and the timer was waiting for the next input. After receiving the next input, the counter was started again and the process was continued up to the count of 0.7.

M. Alpha block

The input for this block is the output from the previous block i.e., from the timer block. Here, the input is processed by the function $f(x)=x*e^{-x}$, where x is the input. The input is multiplied with the corresponding weight and the result is shown in the Fig. 11.

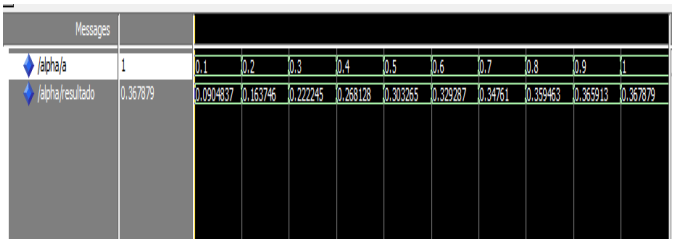


Fig 11. Output for alpha block

N. Addition/Comparision block

Fig. 12 shows the output potential of alpha block at every instant of time during the addition of all the input spike potentials. This block also performed the comparison operation of the cumulative potential with the threshold. The threshold level was set by the user, which decided the action potential of the individual neuron through comparison. Here, the threshold level was fixed as 1.2 mV and when the addition of all the inputs exceeded the 1.2 mV, the alpha block generated an action potential, which showed that the neuron was in the exited state.

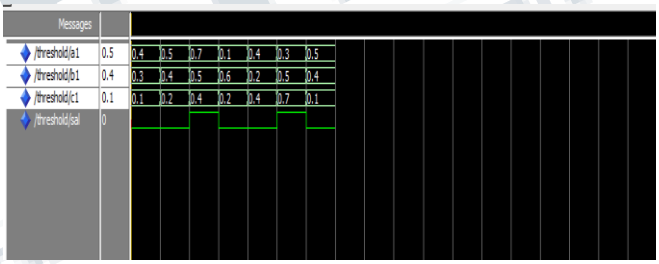
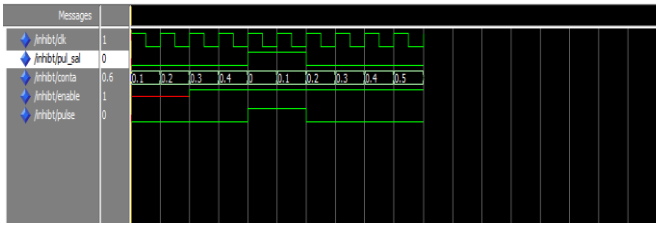


Fig 12.Output for Addition/Comparision block

O. Inhibit block

When the inhibit block received the input signal, it reset the counter, activated the enable signal and started the count again. In Fig. 13, as the input was received, the count was reset and started counting again from the begining. After the completion of the this process, the enable signal was deactivated.



ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

Fig 13. Output for inhibit

H. ***The complete artificial neuron model***

The complete artificial neuron model was developed and implemented in the Anvyl board and the simulation output of the neuron model is given in the Fig.14. Here, two different inputs were given to the neuron from the two dendrites. But, the input with higher value was processed and its weight was added, whereas the lower input was rejected. For digital neuron modeling the weight was given as '1'. Therefore, the inhibit value was the addition of input (in1) with the higher value, which rejected the other neuron's input and its weight. For example, If the weight of the input *in1* is 0011 and the weight of the input *in2* is 0001 (i.e., the input *in1* is greater than *in2)*, then the input *in1* was processed and the weight 0011 was added, which was given as an inhibit result. This condition was stored in the neuron and an output was sent out. This is represented by a high signal, i.e. '1' in the output wave form. The neuron fires only when the addition of inputs exceeds the threshold level. If the inputs of *in1* = 0000, *in2* = 0000 and *in1* = 0001, *in2* = 0001, then these input combinations didn't cross the threshold level and the inhibit value of these two inputs was found to be zero. For all the other inputs the respective outputs are shown in the Fig. 14.

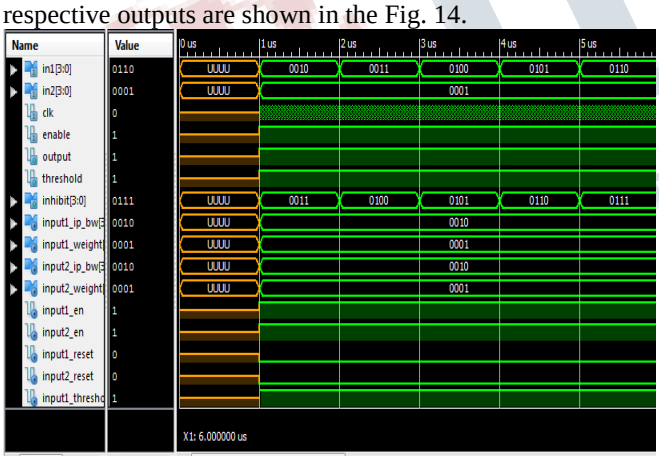


Fig 14. Output of Complete Neuron model

V. HARDWARE IMPLEMENTATION

The proposed neuronal model was implemented in VHDL (Very High Speed Integrated Circuit Hardware Description Language) language. The model was developed in the VHDL language using top-down approach, i.e., divide the complex design into simple designs or modules and each module is redefined with greater details or divided in more sub-systems with the intention to implement in FPGA hardware.

FPGAs have limited usable area and design tool chains, which create difficulty in implementation of large neural networks. Therefore, designs should be optimized in size to implement in an FPGA with limited number of resources [11].

An FPGA design neuron model includes the pre- and post-synaptic connections and an integrator which computes the spikes. Further, it models the activity level of the neuron based on the firing rate that depends on the integrated value. In the basic model, the integrated value was updated for each incoming spike by an increment or a decrement operation on the membrane potential (MP), which was modeled by a counter and depends on the polarity of the received spike. If the spike is positive, MP is found to be incremented, whereas it is decremented when the spike is negative. Also, the present MP is found to be translated into output stream of spikes continuously.

In this work, the Anvyl FPGA development platform was employed, which is a complete, ready-to-use digital circuit development platform based on a speed grade-3 Xilinx SPARTAN-6 XC6SLX45 CSG484 package FPGA kit as shown in Fig. 15. The Anvyl is compatible with all Xilinx CAD tools, including ChipScope, EDK, and the free ISE WebPack™, so designs can be easily carried out economically.

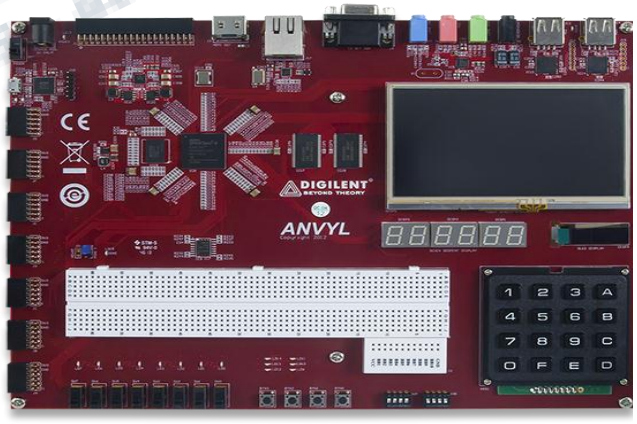


Fig 15. Anvyl FPGA BOARD

In the Anvyl board, the slide switches sw0 to sw3 are used as input 1 (*in1*). Slide switches sw4 to sw7 are used as input 2 (*in2*). LEDs (LD0-LD3) are used for displaying the inhibit output. LED (LD14) is used to output the threshold signal. LED (LD13) is used to display the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

output of the neuron. LEDs (LD9-LD12) are used to show the counter output. LEDs (LD7, LD8) are used to exhibit the spike outputs.

The input combinations experimented in Section H are tested in the Anvyl board and the results are shown in Figs. 16 and 17. Table 1 shows the various input combinations and their correspond output and other signal conditions observed in the simulation as well as hardware designs. The experiment was repeated for other combinations of the inputs (i.e. in1<in2) and the results exhibited similars trends in both the simulation and the hardware design.

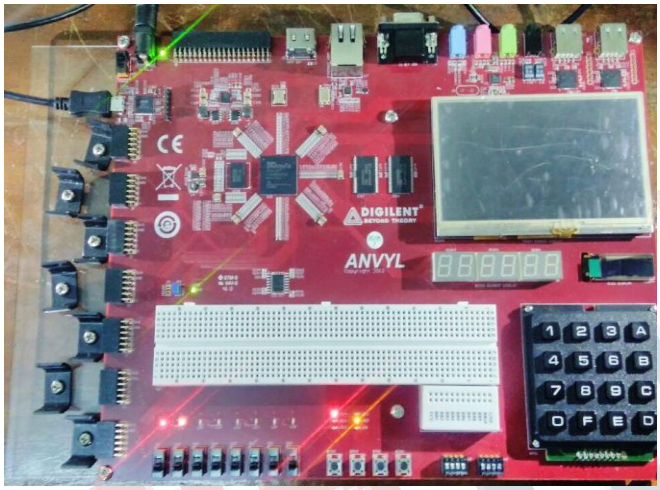


Fig 16. Output for the condition in1=0000, in2=0000, Inhibit = 0000, Threshold = 0 and Output = 1.

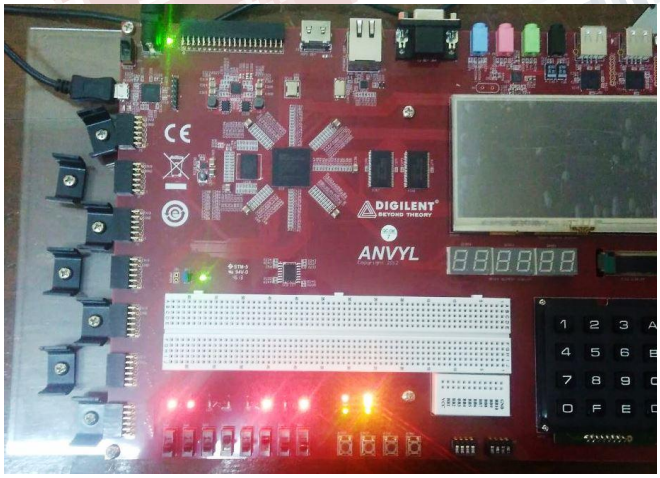


Fig 17. Output for the condition in1=0100, in2=0001, Inhibit = 0101, Threshold = 1 and Output = 1

TABLE I. Signal status for various input combination with the condition in1 > in2 and in2 is constant.

S.No	in1	in2	INHIBIT	OUTPUT	THRESHOLD
1	0000	0000	0000	1	0
2	0001	0001	0000	1	0
3	0010	0001	0011	1	1
4	0011	0001	0100	1	1
5	0100	0001	0101	1	1
6	0101	0001	0110	1	1
7	0110	0001	0111	1	1
8	0111	0001	1000	1	1

VI. CONCLUSION

In this work, basic functionality of the neuron was simulated and implemented in the ANVYL SPARTAN-6 XC6SLX45 CSG484 package FPGA board. The various input and output responses of the three dendtrites model was simulated in the VHDL environment, but a two imput model was implemented in the FPGA board due to the limitation in the input/output devices. But, the implemented model reposnses exhibited the expected or true behaviour responses in line with the simulation results. Thus, this study proved the use of FPGA and VHDL tools for the study of neuron behaviour in the laboratory environment, which paves many ways to study the artificial neuron concept for many real-time applications such as therapy, robotic control, etc.

VII. FUTURE WORK

The present work tested the neuron activity under the influence of some digital input. In the future efforts, it will be extended to study the behaviour of neuron in real-time. Further, there is a possibility to interconnect more artificial neurons to create a complete neuronal network and study their behaviour for many therapeutic or robotic applications.

REFERENCES

[1] Richard Morris and Marianne Fillenz, “Neuroscience: the Science of the Brain”, The British Neuroscience Association, 2003, pp. 1-4, ISBN: 0-9545204--0-8.

[2] WulframGerstner , Werner M. Kistler , Richard Naud, Liam Paninski, “Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition”, Cambridge University Press, July

2014, ISBN-13: 978-1107635197, ISBN-10: 1107635195.

[3] M.L. Hinesa and N.T. Carnevale,"Discrete event simulation in the Neuron", Neurocomputing, Elsevier B.V., pp. 1117-1122, 2004.

[4] J.A. Bailey et.al., "Behavioral simulation and synthesis of biological neuron systems using synthesizable VHDL", Neurocomputing, Elsevier B.V., pp. 2392-2406, 2011.

[5] R.R. Borges et.al., "Effects of the spiketiming-dependent plasticity on the synchronisation in a random Hodgkin-Huxley neuronal network", Communications in Nonlinear Science and Numerical Simulation, Elsevier B.V., pp. 12-22, 2015.

[6] Fernando Pérez-Peña et.al., "Inter-spikes-intervals exponential and gamma distributions study of neuron firing rate for SVITE motor control model on FPGA", Neurocomputing, Elsevier B.V., pp. 496-504, 2014.

[7] Javier Gomez Casado, "Implementation of an artificial neuron in VHDL", 2008.

[8] Peter Tino etal., "Artificial Neural Network Models", Springer handbook of computational intelligence, 1997, Volume 8, Issue 3, pp. 456-457, ISBN: 978-3-662-43504-5.

[9] John D. Zakis and Brian J. Lithgow, "Neuron Modeling using VHDL", 1999.

[10] Xilinx ISE: https: // en. wikipedia . org/ wiki/ Xilinx _ ISE

[11] Anvyl™ FPGA Board Reference Manual: https://reference.digilentinc.com/_media/anvyl:anvyl_rm.pdf