International Conference on Recent Development in Engineering Science, Humanities and Management
National Institute of Technical Teachers Training & Research, Chandigarh, India (ESHM-17)
29th January 2017, www.conferenceworld.in ISBN: 978-93-86171-25-2

FPGA IMPLEMENTATION OF DIGITAL CIRCUIT USING NEURAL NETWORK

Rita Mahajan[1], Sakshi Devi[2], Deepak Bagai[3]

[1]Assistant Professor, [2]ME student, [3]Professor,
Department of Electronics and Communication Engineering,
PEC University of Technology,Chandigarh, (India).

ABSTRACT

The hardware implementation of neural networks in FPGA along with the design of activation function in VHDL is done in this paper. The size of the neural network becomes limited as there is the requirement of more number of multipliers and adders. This paper deals with the technique to reduce the resource by incorporating layer multiplexing of the neural network. It is useful to implement only the largest layer in the neural network instead of implementing the complete network. Then, the largest layer can be used to make it work like the smaller layers. The control block consists of weights, bias, inputs and activation function for every layer of the network. This proves to be useful to decrease the resource requirement at the cost of speed.

Index Terms: Artificial Neural Networks (Anns), Field Programmable Gate Array (FPGA), Layer Multiplexing, Neural Networks.

I. INTRODUCTION

Artificial Neural Networks tackle problems dealing with pattern identification, image processing and medical diagnosis. The ANNs based on biological neurons show parallel behavior and their information processing systems are distributed. This system provides very good speed in real time applications only if the implementation of hardware is done using parallel hardware architecture. The implementation of ANNs is done in two ways: software and hardware implementations. Software implementation of ANNs is generally employed. This is advantageous, since a person is least concerned with the internal working of neural network and only deals with the application of the neural network. [1]

However, the software based Ann's gives slower execution in comparison to the hardware based ANNs. Analog and digital are two hard ware based ANNs. [2]

The analog implementations make use of CMOS's device based non-linear characteristics, but results in thermal drift, inappropriate computation results and deficit of re-programmability. So, digital implementation is preferred. Recent advancements in reprogrammable logic help in implementing huge ANNs on one FPGA chip.

ISBN: 978-93-86171-25-2

ANNs are biologically motivated and need parallel computation for their design. For parallel computations, microprocessors and digital signal processors are not good. Designing fully parallel components are supported by ASICs but it is a very time consuming process of developing such chips. Flexible designs, low cost and design cycle as well as parallel behavior is supported by FPGAs. [3]

II. ARTIFICIAL NEURON

The idea behind artificial neural networks is biological neural systems. In biological neurons, the signals are transmitted through synapses as shown in Fig. 1.The signal transmission is a very complex chemical process in which specific transmitter substances which consists of chemicals are released from the sending side of the synapse. Finally there is increase or decrease in the electrical potential in the body of the receiver cell. If a threshold value is reached by this electrical potential, the neuron fires. This characteristic is required to be reproduced by various algorithms in neural networks. [4]

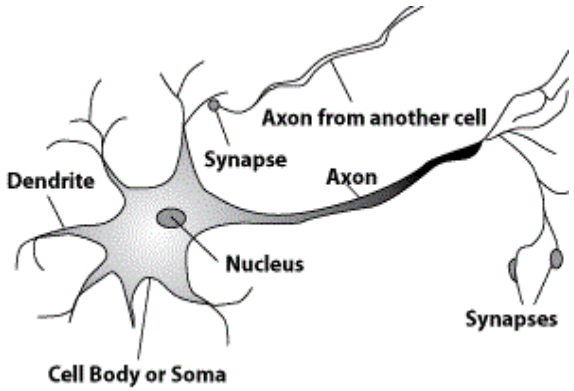

Figure1. A Biological Neuron

Activation or activity level is the function of the inputs received by the neural network and forms the internal state of each neuron. Typically, the neuron which sends the signal to the other neurons is its activation. At a time, only one signal is sent by a neuron, although that signal is further received by several other neurons. For example, suppose a neuron Y shown in Fig. 2. Y obtains input from neurons X_1, X_2 and X_3.These neurons have activations x_1, x_2 and x_3 respectively. w_1, w_2 and w_3 are the corresponding weights on the connections from X_1, X_2 and X_3 to neuron Y. [5]

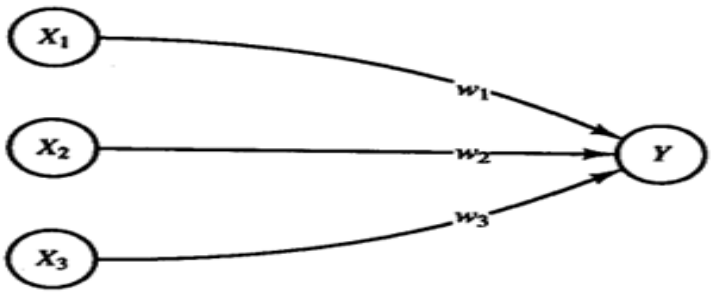

Figure2. A Simple Artificial Neuron

ISBN: 978-93-86171-25-2

To neuron Y, the net input $y_{\_in}$ is equal to the sum of the X_1, X_2 and X_3 i.e. the weighted signals from neurons which are as follows:

$$y_{\_in} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

$y = f(y_{\_in})$ is called as the activation of neuron Y which is the function of its net input e.g. the logistic sigmoid function, $f(x) = \frac{1}{(1+e^{x})}$, f(x) can be other function like identity function, binary step function, binary sigmoid

tc. [6]

ANNs mostly use the following neuron model by adding some variations.

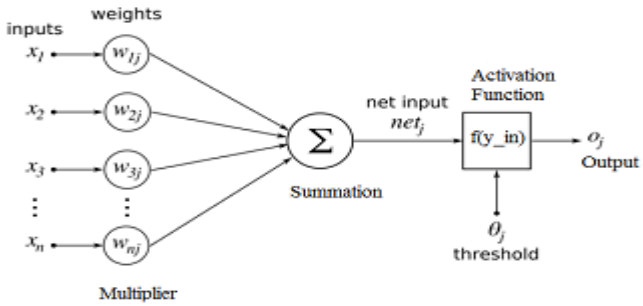

Figure3. Structural Diagram of a Neuron

The ANN which is shown above has n inputs, denoted as x_1, x_2, x_3, ... x_n. Each line which propagates the inputs to the neuron is given a weight value, denoted as w_{1j}, w_{2j}, w_{3j},..... w_{nj} respectively. The activation f(y_in) determines the neuron should be fired or not. It is evaluated in the following way:

$$f(y_{\_in}) = \sum_{k=1}^{n} w_{kj}\, x_k$$

The excitatory connection is denoted by positive weight whereas the prohibitory connection is denoted by negative weight. The neural network' output is given as

$$o_j = f(y_{\_in})$$

The neural computing system includes various activation functions and their interconnections.

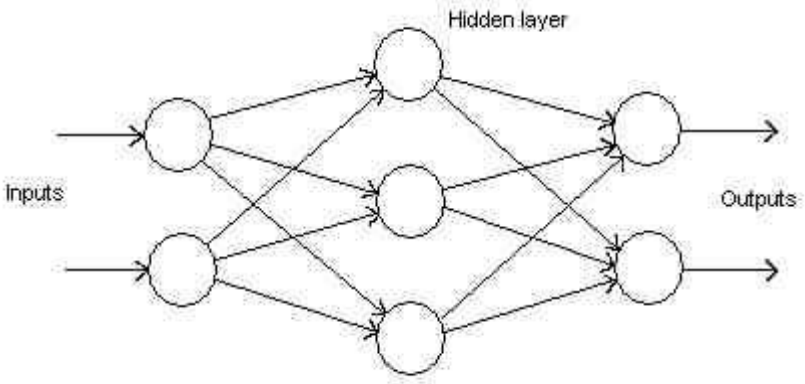

Figure4. Layered Feed forward Neural Network

ISBN: 978-93-86171-25-2

In layered neural networks, the arrangement of neurons is done in the form of layers. The previous layer feeds input to the present layer and then, it is forwarded to the next layer. This type of networks are known as feedforward networks. The input layer neurons transmit only the applied external input to their outputs and their activation function is an identity function. The last layer is called as the output layer and the layers lying between input and output layers are called as hidden layers. The neural networks which consist only of one input and one output layer are called single layer networks and the networks consisting of one or more than one hidden layers are called multilayer networks. [7]

III. THE PROPOSED DESIGN

It includes neuron architecture, activation function and neural design to solve the problem.

3.1 Neuron Architecture

In ANN, the information processing element is known as neuron. The processing of data is done in three steps: give input and weight values, their summation and then sigmoid function filteration. There is a need of a large number of multiplier units to calculate the weighted outputs. Serial accumulation calculates the summation. To solve this problem, Multiplier/Accumulator architecture is selected. It receives the serial input, evaluate the product of inputs with their corresponding weights and collects their sum in a register. The processes are synchronized with the help of clock signal. The number of clock cycles required to perform the neuron's job, is same as that of the number of connections from the previous layer. All the neurons are provided with the bias values with the help of accumulator. The accumulator is given the load signal, so that the bias values are loaded to all the neurons at start-up.

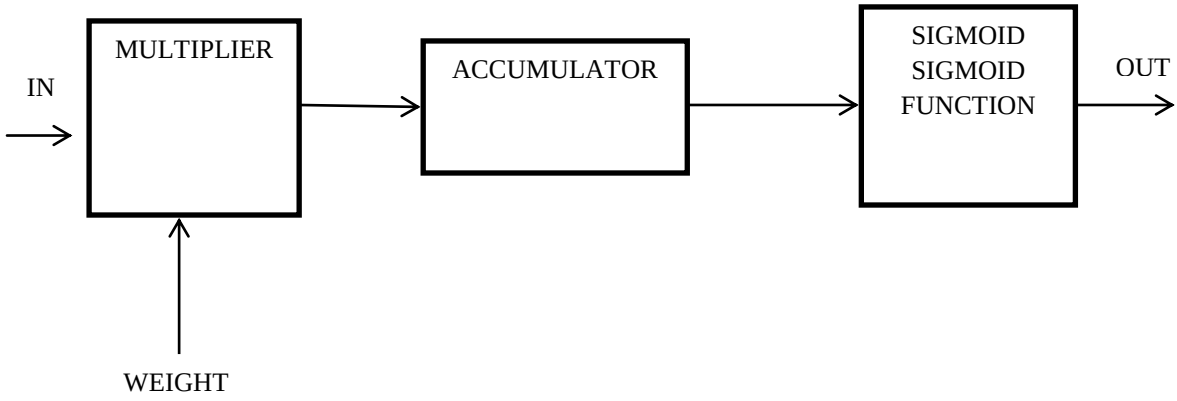

Figure5. Neuron Architecture

3.2 Activation Function

Activation is a necessary part of a neuron. It is difficult to implement activation function on hardware. Care should be taken while implementing large ANNs on digital hardware. The most commonly used activation function is the sigmoid function which is given as follows [8]:

$$y = \frac{1}{1+e^{-x}}$$

ISBN: 978-93-86171-25-2

Direct implementation of activation function on FPGA is not suitable. Mostly, alternatives of sigmoid function which are easily simplified are used. It is an expensive process too. To approximate, sigmoid functions, there are two practical approaches. One is piece-wise linear approximation and the other is lookup tables. [9]

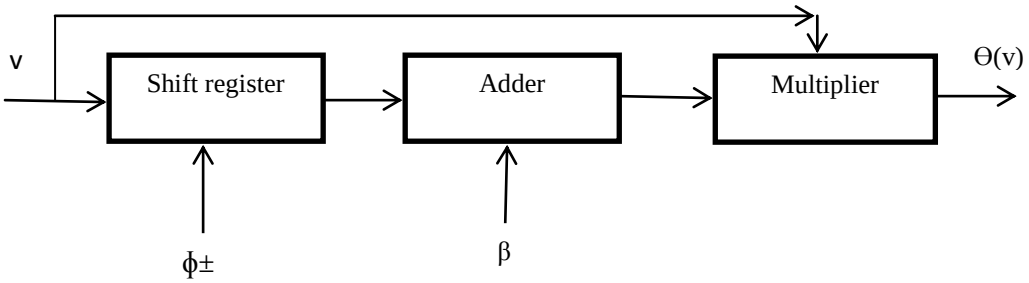

Figure6. Structural Diagram of Sigmoid Function

3.3 Layer Architecture

During the first implementation of the ANN layer, input is received from a common input line, evaluate the product of input and its weights from their weight ROM and find the summation of the product. The present layer receives and processes the information from the previous layer 3 neurons in 3 clock cycles. Every neuron has its final value determined after the three clock cycles. Then, the present layer pass these values to its output one after the other for the next layer. It is shown in Fig. 7.

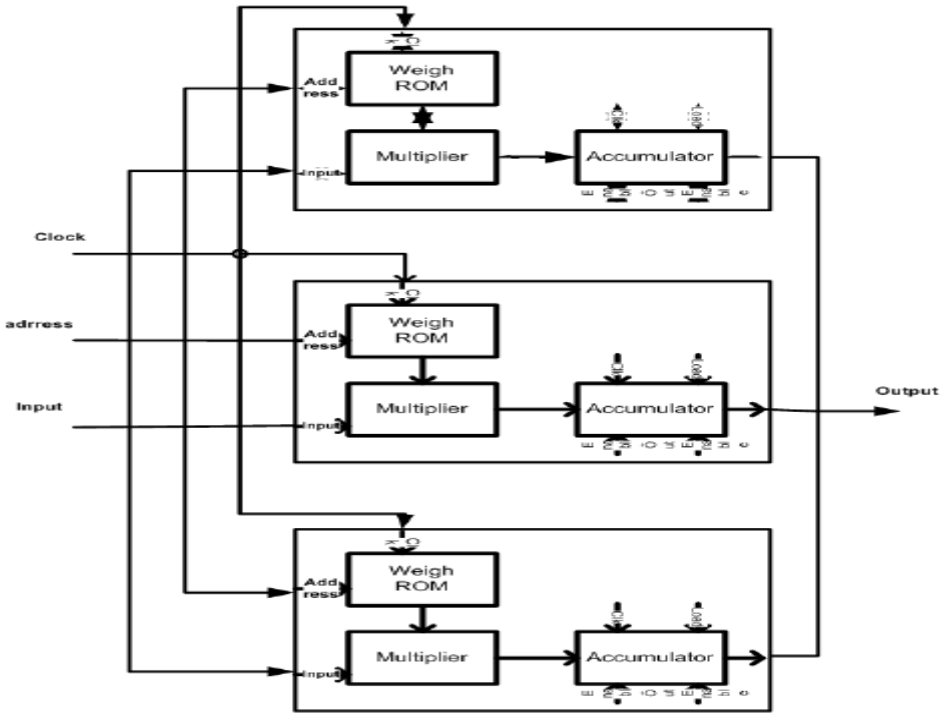

Figure7. Block Diagram of a Layer Having Three Neurons

At one time, only one neuron transfers its output to the output layer. The implementation of one activation function for each layer is very easy. Pipelining also appears since a new pattern can enter the network while another is propagating through the layers.

ISBN: 978-93-86171-25-2

In a fully parallel neural network, the number of multipliers for one neuron must be same as the number of connections to this neuron which gives a fast but inflexible network. Since all the product terms should be added, the number of full adders is one less than the number of connections to the previous layer minus one. Another disadvantage of a fully parallel network is efficient utilization of gate resources. Figure 8 depicts a neural network having single layer with three input units and one output unit.

IV. CONCLUSION

This study has illustrated the neural network implementations by FPGAs. The number of neurons as well as layers can be increased or decreased since the proposed network is modular in nature. The key idea behind this study arises from the fact that an FPGA coprocessor having less logic density can be preferred in making artificial neural network to solve different application problems. Future work involves the implementations of huge multilayer ANNs. The main points are the size and number of multipliers and the number of interlayer connections.

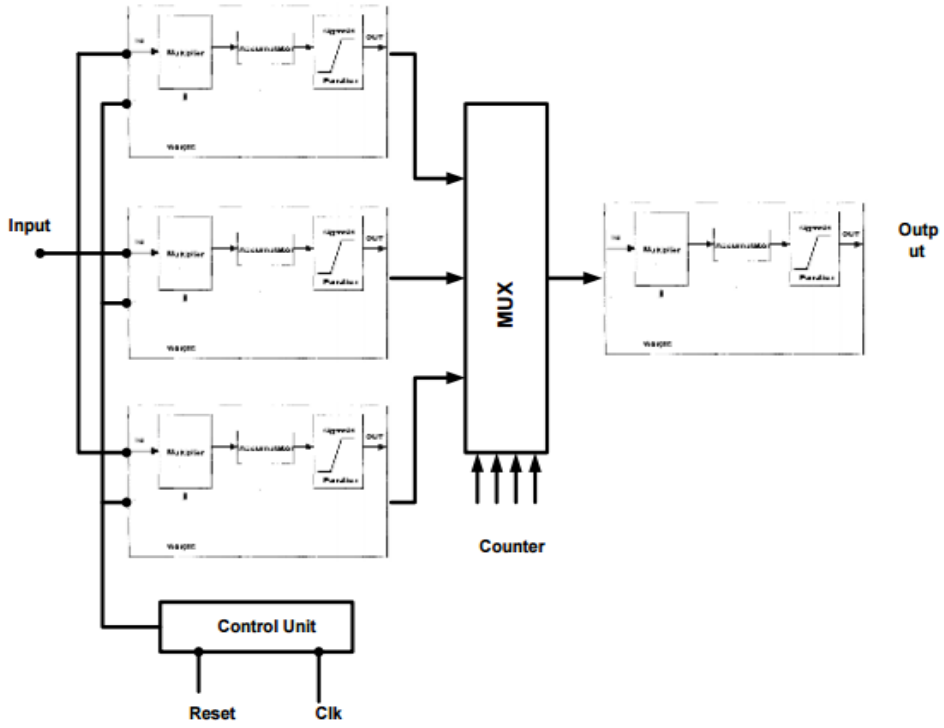

Figure8. Artificial Neural Network Design

REFERENCES

[1] Sahin S, Becerikli Y, Yazici S,Neural network implementation in hardware using FPGAs, International Conference on Neural Information Processing, 2006,pp. 1105-1112.

[2] Mohammed E. Z. and Ali H. K.,2013, Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array, International Journal of Computer Theory and Engineering, 5(5), 780.

ISBN: 978-93-86171-25-2

[3] Sahin, Suhap, Yasar Becerikli, and Suleyman Yazici. Neural network implementation in hardware using FPGAs, International Conference on Neural Information Processing,2006, pp. 1105-1112. Springer Berlin Heidelberg, 2006.

[4] S. Haykin, neural networks-a comprehensive foundations Second Edition, ISBN: 0132733501, 1998.

[5] Fausett L., 1994, fundamentals of neural networks: architectures, algorithms, and applications Prentice-Hall, Inc.

[6] Muthuramalingam, A., S. Himavathi, and E. Srinivasan,Neural network implementation using FPGA: issues and application,International journal of information technology,Vol.4, no. 2,2008,86-92.

[7] Asha R., Bowrna P., Mhaboobkhan F.,April-May,2013,Implementation of Feed Forward Neural Network Using Layer Multiplexing For Effective Resource Utilization in FPGA, International Journal Of Research in Engineering and Advanced Technology,Vol:1, pp.1-3.

[8] Rana D., Aljabar A., September, 2012, Design and Implementation of neural network in FPGA, Journal of Engineering and Development, Vol.16, No.3,pp.73-90.

[9] Dondon P., Carvalho J., Gardere R., Lahalle P., Tsenov G. and Mladenov V., November,2014, Implementation of a feed-forward Artificial Neural Network in VHDL on FPGA, Neural Network Applications in Electrical Engineering (NEUREL), pp. 37-40.