ISSN 2229-5518

Digital Hardware Implementation of Artificial Neural Network for Signal Processing

A.Thilagavathy , K.Vijaya Kanth

Abstract— These Artificial Neural Networks support their processing capabilities in a parallel architecture. It is widely used in pattern recognition, system identification and control problems. Multilayer Perceptron is an artificial neural network with one or more hidden layers. This paper presents the digital implementation of multi layer perceptron neuron network using FPGA (Field Programmable Gate Array) for image recognition. This network was implemented by using three types of non linear activation function: hardlims, satlins and tansig. A neural network was implemented by using VHDL hardware description Language codes and XC3S250E-PQ 208 Xilinx FPGA device. The results obtained with Xilinx Foundation 9.2i software are presented. The results are analyzed by using device utilization and time delay.

Index Terms— Back propagation, *FPGA, Multi Layer Perceptron, Neuron PLAN approximation, Sigmoid Activation,VHDL*

1 INTRODUCTION

The human brain is probably the most complex and intelligent system in the world. It consists of the processing element which is called as neurons. Each neuron has performs a set of simple operations but they exhibit complex global behavior in the network. Artificial neural network (ANN) is used for engineering purposes, to replicate the brain's activities. Artificial neural networks (ANNs) have been used successfully in solving pattern classification and recognition problems, function approximation and predictions. Their dispensation capabilities are based on their architecture which is highly, parallel and interconnected. For a specific application in all biological systems through a learning process an ANN was configured; the synaptic connections are adjusted between the neurons is required by the learning process.

An artificial neural network is a massively parallel distributed processor made up of simple processing units (neurons), which has the ability to learn functional dependencies from data. It resembles the brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process.
- Strengths of Inter neuron connection are called synaptic weights, which is used to store the acquired knowledge.

This procedure is used to perform the learning process so it is called as a learning algorithm; the function is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective.

A simple processing unit such as neuron, which collects some weighted data, sums them with a bias and calculates an output to be passed on. Fig 1 shows the simple neuron model. The inputs to the neuron are P1, P2, P3 and the w1, w2, w3 are the corresponding weight values. The weight values and their corresponding inputs are multiplied and summed together, which is the input to the activation function block. The function that the neurons are used to calculate the output is called the activation function.

An artificial neural network consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes are called the multilayer perceptron. Each node in one layer connects with a certain weight W_{ij} to every node in the following layer.

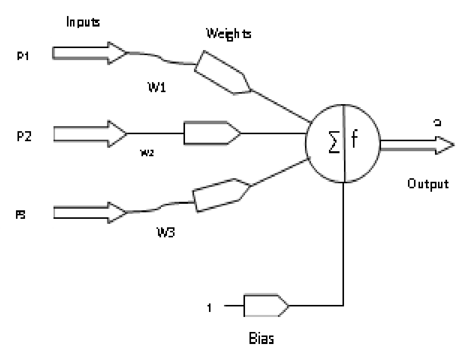

Fig.1 A simple artificial neuron model

Hardware implementation of ANN can be implemented on Application Specific Integrated Circuits (ASICs) or Field-Programmable Gate Arrays (FPGAs). ASIC design has several drawbacks such as the ability to run only a specific algorithm and limitations on the size of a network. Hence FPGA are used to overcome the drawback, which has flexibility with an appreciable performance. It maintains the high processing density, which is needed to utilize the parallel computation in an ANN. Every digital module is instantiated on the FPGA by concurrently and hence the parallel operation is performed. Thus the speed of the network is not dependent on the complexity. To design a multilayer perceptron model is the main purpose of this work. The model consists of two stages. The first stage is the multiplication of parallel inputs and weight values. And the second stage is the nonlinear activation function for the output signal and weight update. Three types of nonlinear activation functions were considered: symmetrical hard limiter, symmetric saturating linear and hyperbolic tangent sigmoid.

2 Multilayer Perceptron Network

The terms "Neural Network" (NN) and "Artificial Neural Network" (ANN) is used without qualification, is usually referred as Multilayer Perceptron Network. There are many types of neural networks including Probabilistic Neural Networks, General Regression Neural Networks, Radial Basis Function Networks, Cascade Correlation, Functional Link Networks, Kohonen networks, Gram-Charlier networks, Learning Vector Quantization, Hebb networks, Adaline networks, Heteroassociative networks, Recurrent Networks and Hybrid Networks.

Here we used the most widely used type of neural networks: Multilayer Perceptron Networks. A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. A multiple layers of nodes in a directed graph; with each layer fully connected to the next one is model by an MLP. Apart from the input nodes, each node is a neuron has a processing element such as neuron is connected to the nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training the network.

Fig 2 shows a three layer perceptron network. This network has an input layer (on the left) with p inputs, one hidden layer (in the middle) with L neurons and an output layer (on the right) with m outputs.

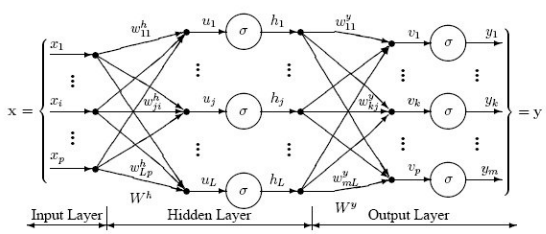

Fig.2 Three layer perceptron network

For classification problems, only one winning node of the output layer is active for each input pattern. Each layer has provided a connection with its adjacent layers. There is no connection between non-adjacent layers and there are no recurrent connections. Each of these connections is defined by an associated weight. The weighted sum of its inputs is calculated neuron and applies an activation function that produces the high or low neuron output. By using this type of propagation from the output of each layer, the MLP generates the specified output vector from each input pattern. By using the supervised learning algorithm such as back propagation, the synaptic weights are adjusted.

Different types of activation functions have been proposed to transform the activity level (weighted sum of the node inputs) into an output signal

3 Activation Function

The activation function may be linear or nonlinear function of N. A particular nonlinear activation function of neuron is chosen to satisfy specification of the training algorithm that the neural network is attempted to run. In this work, three types of the most commonly used nonlinear activation functions are implemented on FPGA. They are hard limit activation function, saturating linear activation function, Hyperbolic Tangent Sigmoid activation function.

3.1 Hard limit activation function

In the hard limit activation function, if the function argument is less than 0 then the output of the neuron is 0, if the function is greater than or equal to 0 then the output of the neuron is 1. This function is used to create neurons that classify inputs into two distinct categories.

$$Y = f(u)\begin{cases} 0 \; if \; u < 0 \\ 1 \; if \; u \geq 0 \end{cases} \qquad (1)$$

If the hard limiter activation function is used with neuron then it is referred as the McCulloch-Pitts model.

3.2 Saturating linear activation function

This type of nonlinear activation function is also referred to as piecewise linear function. It has either a binary or bipolar range for the saturation limits of the output. The mathematical model for a symmetric saturation function is described as follows:

$$Y = f(u) = \begin{cases} -1 & for \quad u < -1 \\ u & for -1 \leq u < 1 \\ 1 & for \quad u \geq 1 \end{cases} \qquad (2)$$

3.3 Hyperbolic tangent sigmoid activation function

This function takes the input any value between plus and minus infinity and the output value into the range - 1 to 1, according to the expression

$$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}} \qquad (3)$$

The tansig activation function is commonly used in multilayer neural networks that are trained by the back propagation algorithm since this function is differentiable. The tansig function is not easily implemented in digital hardware because it is consists of an infinite exponential series.

A simple second order nonlinear function presented by Kwan can be used as an approximation to a sigmoid function. This nonlinear function can be implemented directly using digital techniques.

ISSN 2229-5518

The following equation is a second order nonlinear function which has a tansig transition between the upper and lower saturation regions:

$$f(n) = \begin{cases} n(B - g.n) & for \quad 0 \le n \le L \\ n(B + g,n) & for \quad -L \le n < L \end{cases} \qquad (4)$$

Where B and g represent the slope and the gain of the nonlinear function *f* (*n*) between the saturation regions -L and L.

4 Results

The digital hardware were modeled using VHDL and simulated using Model Sim 5.7 and Xilinx 9.2 ISE was used as synthesis tool for implementing the designs in Spartan 3E FPGA

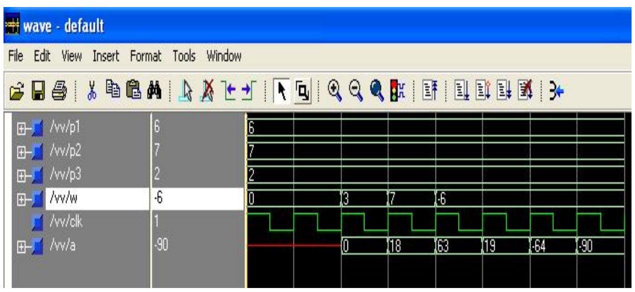

Fig.3 a) simulation wave form of linear neuron without activation function

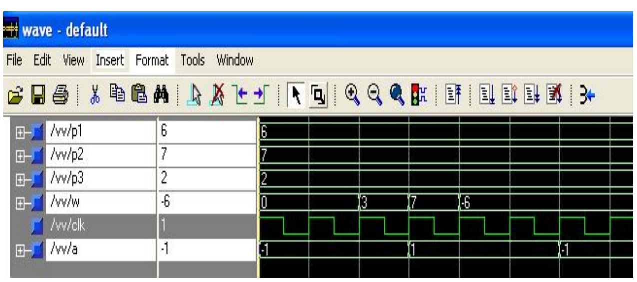

Fig.3 b) simulation wave form of linear neuron with hard limit activation function

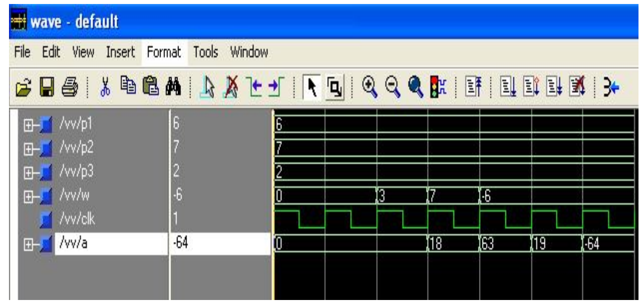

Fig.3 c) Simulation wave form of linear neuron with saturating linear activation function

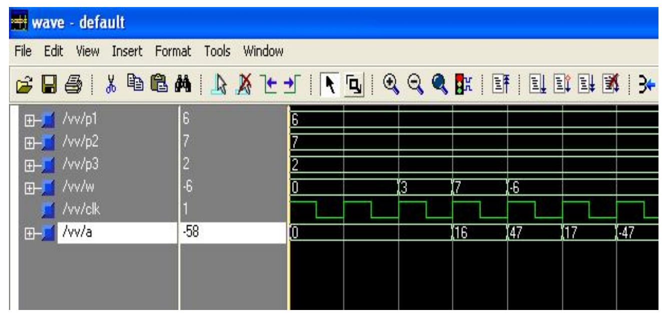

Fig.3 d) Simulation wave form of linear neuron with hyperbolic tangent sigmoid activation function

Figure 3 shows the simulation waveform of neural network with all the three activation functions. Table 1&2 gives the performance and resource use summary for 3 layer perceptron network with 3 different activation functions.

Table 1: Timing summary for different activation functions.

Neuron type	Hard limit	Saturating linear	Hyperbolic tangent
Max path delay	15.820 ns	15.231ns	29.598ns

Table 2: Device utilization summary for different activation functions.

Neuron type	Hard limit	Saturating linear	Hyperbolic tangent
Number of Slices	10	11	33
Number of Slice FF	8	8	8
Number of 4 input LUTs	16	21	63
Number of bonded IOBs	25	25	25
Number of Multipliers	3	3	4
Number of GCLKs	1	1	1

5 CONCLUSION

The results of this work successfully demonstrate the hardware implementation of Multilayer perceptron networks with three different activation functions for image recognition. The design was synthesized using Xilinx 9.2i ISE tool and implemented in Spartan 3e FPGA. By using this, the comparisons to be made between the hardware realizations of this neuron, which are regarded as basic building block of artificial neural networks. The operation frequency in all cases is very good and it gives a clear idea of the advantages of using FPGAs, since multiple modules can be working in parallel with a minimum reduction in performance due to the increased number of interconnections. Finally, it can be said that FPGAs technology and their low cost, and reprogrammability make this approach a very powerful option for implementing ANNS. The implemented design can be used in adaptive filters, voice recognition, and image recognition.

REFERENCES

[1] Rafid Ahmed Khalil Sa'ad Ahmed AI-Kazzaz "Digital Hardware implementation of Artificial Neurons models using FPGA" , Springer U.S.,2012.

[2] Neeraj Chasta , Sarita Chouhan and Yogesh Kumar, " Analog VLSI implementation of neural network architecture for signal processing" VLSICS., Vol.3.no.12,2012.

[3] Manish Panicker, C.Babu, "Efficient FPGA Implementation of Sigmoid and Bipolar Sigmoid Activation Functions" *IOSR Journal of Engineering* 2012.

[4] R.Omondi, C. Rajapakse, "*FPGA Implementation of Neural Networks", Springer* U.S., 2006.

[5] O. Maischberger ,V. Salapura , " A Fast FPGA Implementation of a General Purpose Neuron", Technical University , Institute of in formatik , Austria , 2006.

[6] Hiroomi Hikawa , "A Digital Hardware Pulse-Model Neuron With Piecewise Linear Activation Function," *IEEE Trans.Neural Networks, vol.14,no.5*,pp.1028-1037,Sept.2003

[7] M. Banuelos Sauced, et. al., "Implementation of neuron model using FPGAs". *Journal of Applied Research and Technology ISSN: 1665- 6423, vol. 3 (10), 2003, pp. 248-255.*

[8] Parasovic A., latinovic I . , "A Neural Network FPGA Implementation" . IEEE. 5th seminar on Neural Network Application in Electrical Engineering , September 2000 , PP117 – 120