Volume: 4 Issue: V Month of publication: May 2016
DOI:

www.ijraset.com
IC Value: 13.98

Volume 4 Issue V, May 2016
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Design and Implementation of Logic Gates and Adder Circuits on FPGA Using ANN

Neelu Farha[1]., Ann Louisa Paul J[2]., Naadiya Kousar L S[3]., Devika S[4]., Prof. Ruckmani Divakaran[5]

[1,2,3,4,5] *Department of Electronics and communication Engineering, Dr.T.Thimmaiah Institute of Technology (Approved by AICTE, New Delhi & Affiliated to VTU, Belgavi) Oorgaum, Kolar Dist.-563120,Karnataka*

Abstract- In this paper, Design and hardware implementation of multiple neurons on Field Programmable (FPGA)is done sequentially. First Multiple Neurons are implemented then logic Gates and Adder circuit are implemented using Feed Forward Neural Network. FPGA has been used to reduce the neuron hardware by designing activation function inside the neuron without using lookup tables. With low precision neural network designs, FPGA's have higher speed and smaller in size for real time applications compare to VLSI.

Keywords- Artificial Neural Network, Activation functions, Feed Forward Neural Network, verilog.

I. INTRODUCTION

As the title suggests our project deals with the hardware implementation of Artificial Neural Networks using Activation functions as application on Logic Gates ,Half Adder and Full Adder. In technological view humans are trying to emulate the behavior of Biological Neuron. Artificial Neuron can be modeled to develop future applications in computational Neuroscience, as well as Artificial intelligence. ANN simplify the behavior of human brain so their applications are used in different fields such as Medical applications, Security, Robotics, Electronics, Military etc. ANN's are implemented in software, but designers need not to know the inner network of neural network elements, they are only to concentrate on application of neural network. However, software based ANN's used in real time application is slower execution compared with hardware based ANN's FPGA based architectures offer high flexibility designs. Digital system architecture is to realize Feed Forward Multilayer neural network. The developed architecture is designed using very high speed integrated circuit hardware description language. There are 3 important layers in neuron model, they are input layer, hidden layer and output layer. ANN's are biologically inspired and require parallel computation, hence microprocessors and DSP's are not suitable for parallel designs [1].

II. NEURON ACTIVATION FUNCTION

One of the most important part of a neuron is its Activation function and the Activation function of a node defines the output of that node given in an input or set of inputs. The non-linearity of the Activation function makes it possible to approximate any function. The types of Activation functions used in this work are Symmetrical Hard limit Activation Function, Saturating Linear Activation Function and Sigmoid Activation Function [2].

A. *Symmetrical Hard limit Activation Function*

It is referred as '*Hardlims*'. Its is used to classify inputs into two distinct categories. Hard limiting means clipping, it is an limiting action in which there is a over permitted dynamic range, negligible variation in the expected characteristics of the output signal and a steady state signal at the maximum permitted level. Hard limit activation function can be calculated as follows.

$$a = \begin{cases} -1 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

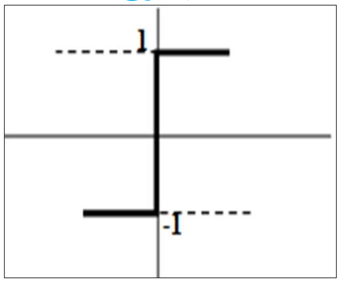

Fig 1. Hardlimit Function

B. Saturated Linear Activation Function

Satlin is a neural transfer function. Transfer function calculate a layers output from its net input. The output is as shown below

$$a = \begin{cases} -1 & n < -1 \\ 0 & -1 \leq n \leq 1 \\ 1 & n > 1 \end{cases}$$

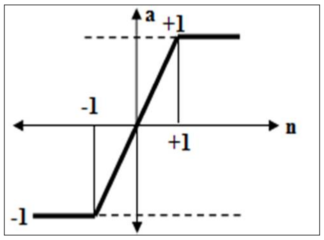

Fig 2. Satlin Function

C. Sigmoid Activation Function

In the hardware concept of Neural Network it is not easy to implement on FPGA, because it consists of infinite exponential series. Formula to calculate sigmoid function is as follows.

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

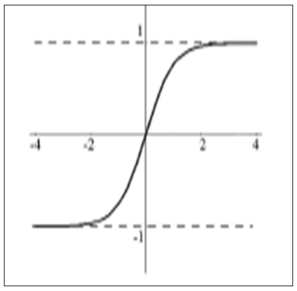

Fig 3: Sigmoid Function

III. COMPARISION OF BIOLOGICAL NEURON WITH ARTIFICIAL NEURON

A Human Brain consists of large number of neurons .A typical neuron collects a signal from hair like structures called Dendrites. Neurons sends out a spike of electrical signal through a long, this stand known as Axon [3]. The comparision between biological neuron and Artificial neuron is shown in fig. 4, where the Neuron is compared to processing element, Dendrites are compared as combining functions, Cell body is compared as transfer function, Synapse are compared as weights and Axon is compared to element output.

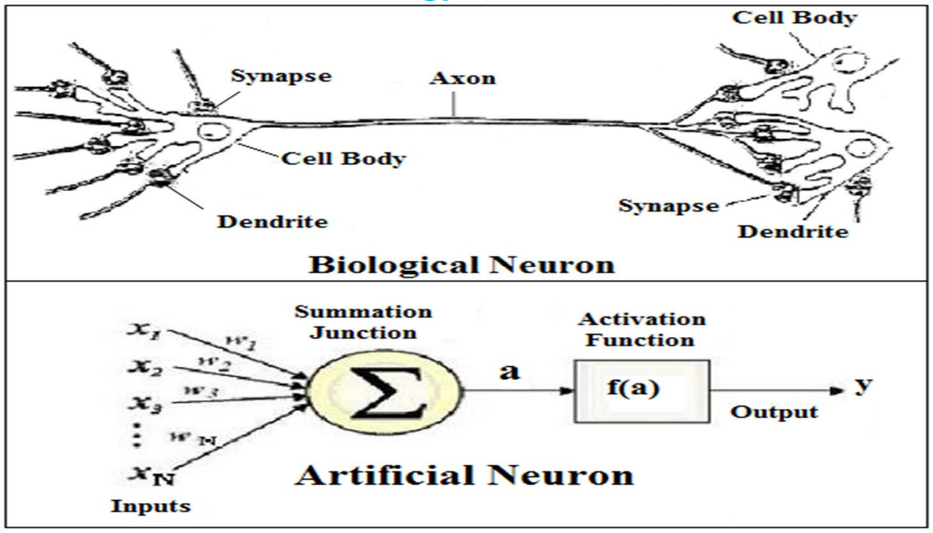

Fig 4. Comparison of biological Neuron and Artificial Neuron

IV. ARTIFICIAL NEURAL NETWORK WITH FEED FORWARD SYSTEM

Artificial Neural Network is an abstract description of human brain. It is a highly interconnected mathematical model that can be programmed in software and moduled in hardware. It provides very high degree of parallel computation.

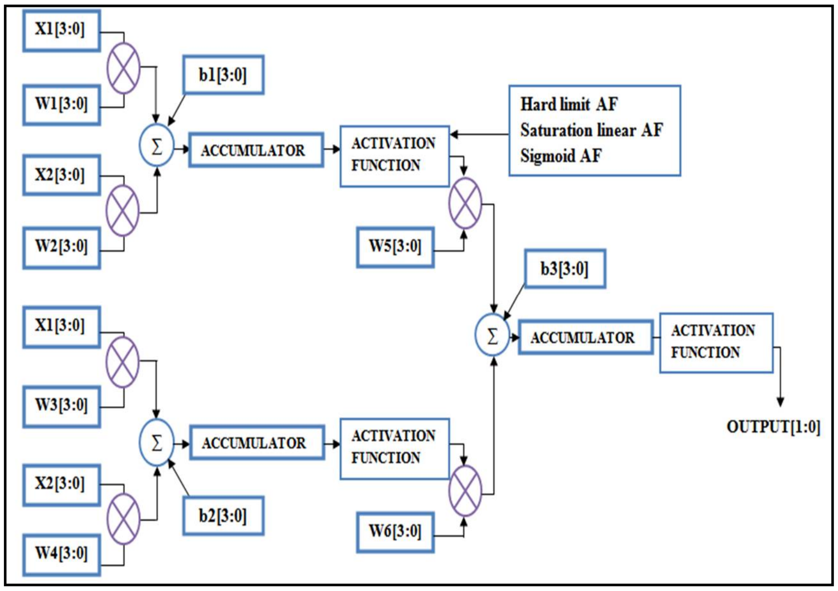

Fig 5. Functional Model of an Artificial Neuron

A. *Working of Neuron*

The working model of neuron is shown in Fig.5. It consists of n bits of input and weights which are multiplied with shift and add multiplier. There are two sub inputs for each neuron and output result is given to activation function [4]. These activation functions are used to fire the neuron when their action potential crosses its threshold. Result of corresponding activation block will generate 1 bit output signal to fire neuron.

$$u=\sum_{i=1}^{N} x_i w_i + b_i$$

B. *Feed Forward Neural Network*

The Feed Forward networks are generally arranged in distinct layers that contains only forward path that is shown in Fig.6. In this type of network each layer receives inputs from the previous layer and outputting to the next layer which indicates there is no feedback. It means that signals from 1 layer are not transmitted to a previous layer.

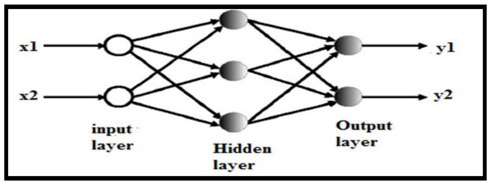

Fig 6. Structure of Feed Forward Neural Network

V. DESIGN OF LOGIC GATES USING ARTIFICIAL NEURAL NETWORK

A. *AND Gate Using Hard Limit Activation Function*

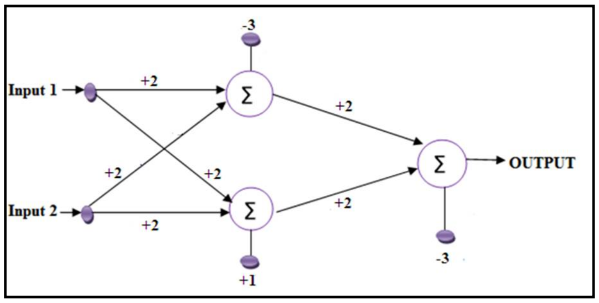

Fig 7. Neuron Model for AND Gate Hard Limit Activation Function

In case of Hard limit activation function A High output (1) results only if obtained output from neural network is greater than 0. The low output(-1) results only if obtained output from neural network is less than 0.

B. *OR Gate Using Saturating Linear Activation Function*

The OR gate is a digital logic gate that implements logical disjunction. A HIGH output (1) results if one or both the inputs to the gate are high. If neither input is high, a LOW output 0 results. In Neural network, if output is greater than 0 it is considered as +1,if output is less than or greater than 0 it is considered to be -1.This can be decided by the Saturating Linear activation function.

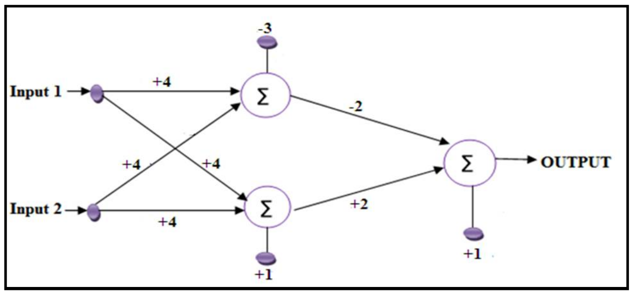

Fig 8. Neuron Model for OR Gate using Saturating Linear Activation Function

C. *XOR Gate Using Sigmoid Activation Function*

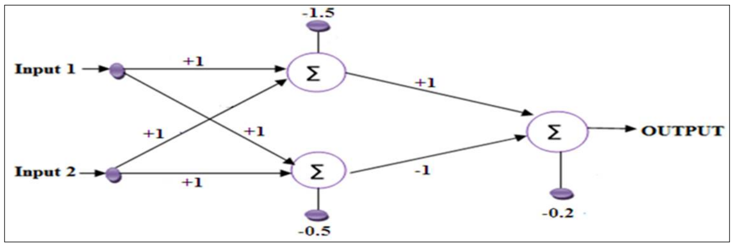

Fig 9. Neuron Model for XOR Gate using Sigmoid Activation Function

VI. DESIGN OF HALF ADDER AND FULL ADDER USING ANN

A. Half Adder Using Neural Network

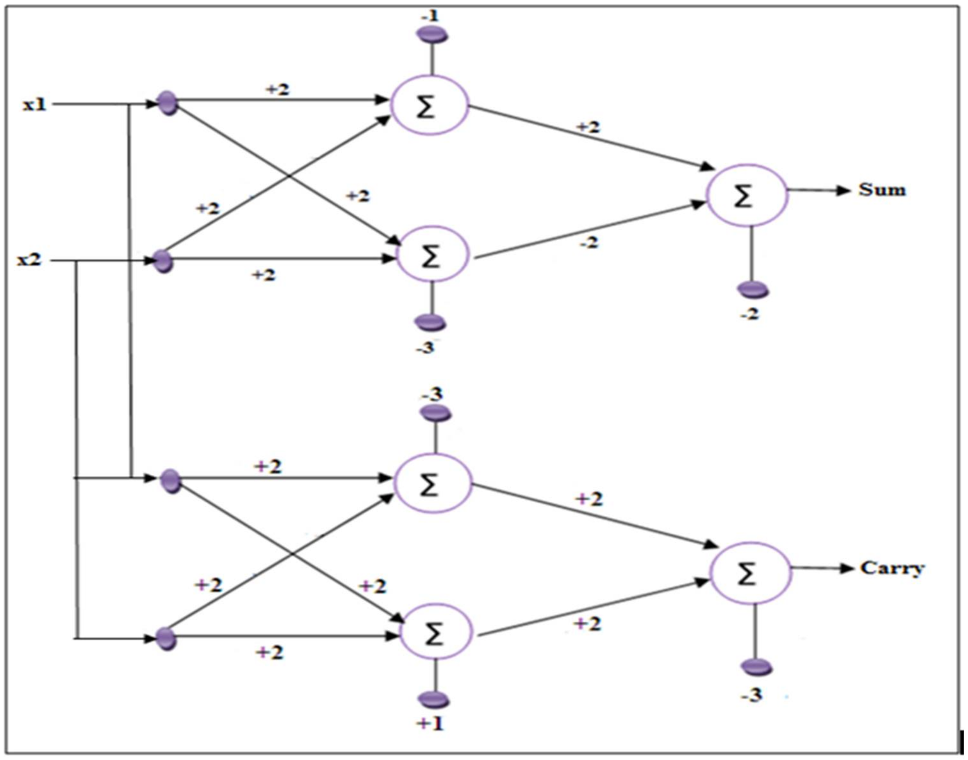

Fig 10. Neuron Model for Half Adder

B. Full Adder Using Neural Network

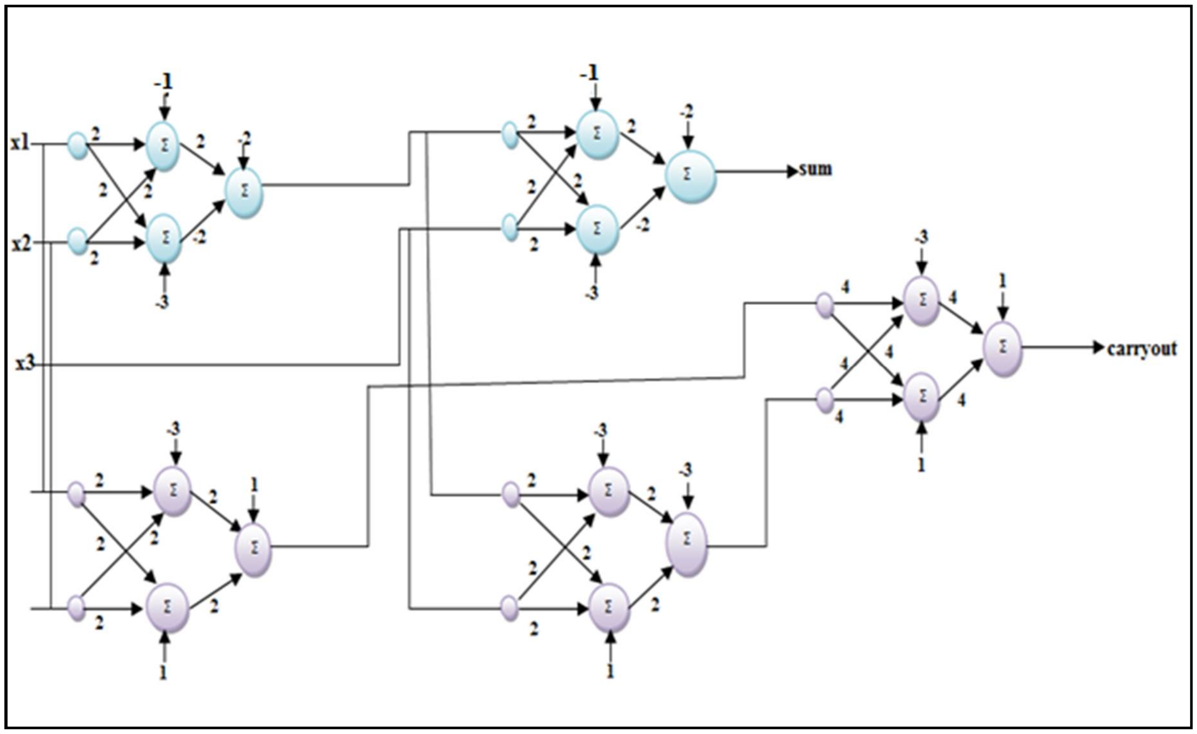

Fig 11. Neuron Model for Full Adder

Volume 4 Issue V, May 2016
ISSN: 2321-9653

VII. RESULTS

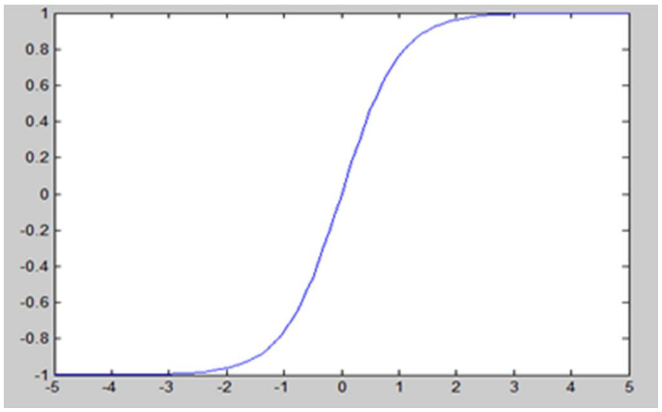

Fig12. Sigmoid MATLAB Curve

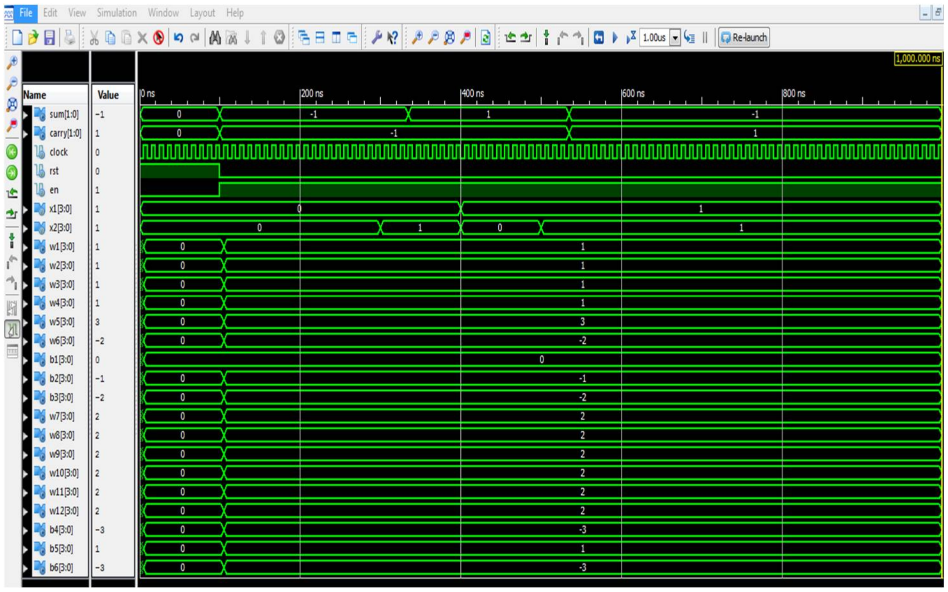

Fig13.Simulation for Half Adder using Activation function

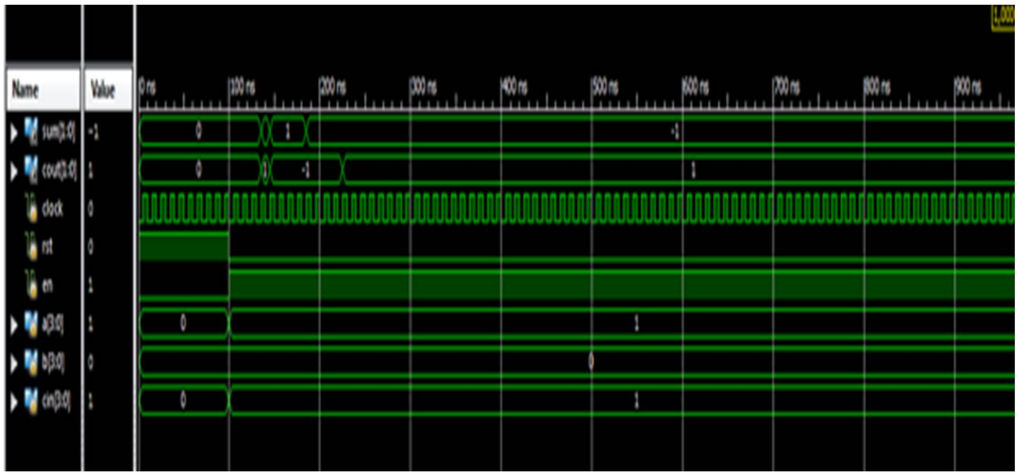

Fig14.Simulation for Full Adder using Activation function

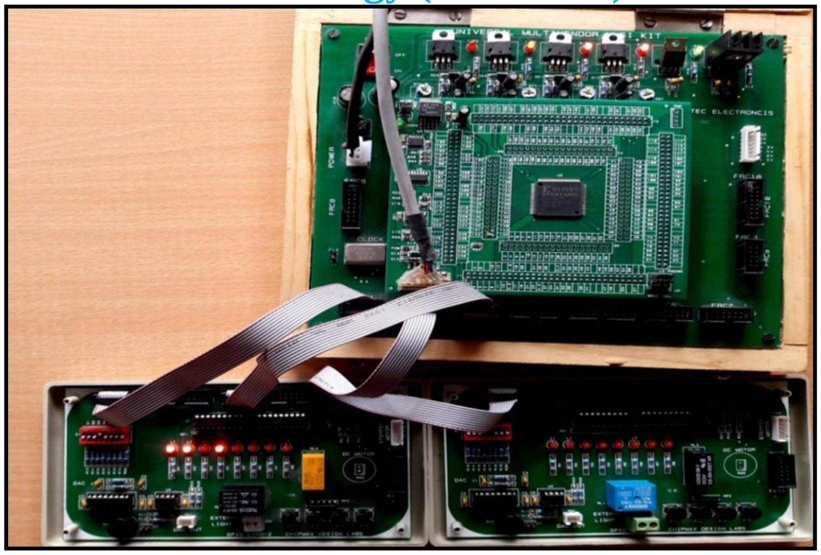
Fig 15.ANN Hardware implementation on Spartan-3 on FPGA

VIII. CONCLUSION

The Logic Gates, Half adder and Full adder using activation function is successfully implemented in Artificial Neural Network. The code is written in Verilog and successfully simulated and synthesized using Xilinx ISE9.1 using Spartan-3 FPGA. ANNs are resulting high degree of parallel computation. FPGA not only offers flexible designs and saving in cost .Hardware implementation of Logic Gates, Half Adder and Full Adder using Activation function is done on Spartan-3 FPGA and the device used is XC3S400 with package TQ144.Hence we can conclude that Artificial Neural Network is a best field to implement Combinational circuits on FPGA.

REFERENCES

[1] Engel,Fernando.M G.Moraes Rolf F. Molz, Paulo ," Codesign to Fully Parallel Neural Network for a Classification Problem",University Montpellier II, France, 2000.

[2] Esraa Zeki Mohammed and Haitham Kareem Ali, “Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array” International Journal of Computer Theory and Engineering, Vol. 5, No. 5, October 2013.
Haitham Kareem Ali and Esraa Zeki Mohammed, “Design Artificial Neural Network Using FPGA” IJCSNS VOL.10 No.8, August 2010.

[3] Hardik H. Makwana, Dharmesh J. Shah, Priyesh P. Gandhi,” FPGA Implementation of Artificial Neural Network” International Journal of Emerging Technology an Advanced Engineering Volume 3, Issue 1, January 2013

[4] Muthuramalingam. A ,S. Himavathi, and E.Srinivasan, “Neural network implementation using fpga : Issues and Application,” The International Journal of Information Technology , vol. 4, no.2, pp.86-92, 2008.