International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072

Design and Synthesis of Artificial Neural Network using HDL

Oshin R. Ojha[1], Hardik H. Makwana[2]

[1]P.G. student, Hasmukh Goswami College of Engineering, Ahmedabad, Gujarat
[2]Assistant professor, Hasmukh Goswami College of Engineering, Ahmedabad, Gujarat

---

Abstract - *In this paper, a Single Layer Perceptron for AND function has been designed and synthesized. The code is designed in MATLAB 13 and synthesized in Xilinx .It has been trained using Perceptron Convergence Algorithm. The results are compared with NNTOOL (Neural Network Design and Development Toolbox)available in MATLAB 13.The results shows that code designed in MATLAB gives exact output without any errors and NNTOOL gives output with errors present.*

Key Words: **Artificial Neural Network, Single Layer Perceptron, Perceptron Convergence Algorithm**

1. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain process information. The advances in the VLSI research and the performances attainable by Integrated circuits (IC) have risen to the great interest in the VLSI implementation of ANN. when going for implementation of single quantity, ASIC implementation becomes too much costly than implementation on FPGA. Advances in FPGA technology in last decade, make implementation of ANN on FPGA more interesting and easier than ASIC.

1.1 Overview of Artificial Neural Network

The Artificial Neural Network (ANN) is inspired by the 'Biological Neuron'. In Biological Neural Network, a cell or Neuron receives responses from other neurons via dendrites. When cell body contains enough action potential from dendrites, it fires and sends activation signal via axon to the dendrites of other neuron.

Same way, Artificial Neural Network has been conceptualized. It gets electrical signal from input and their summation is passed through a threshold function. If the result of summation exceeds the threshold than the neuron 'fires' that is the output of threshold function will be positive else output of threshold function will be negative. In case when actual output is different from desired output then weights and bias are set. So when same pattern will come it will produce correct output. It is shown in fig 1.

1.2 Design of Single Layer Perceptron

Definition: "A perceptron is a type of Neural Network which computes a weighted sum of its inputs and puts this sum through a special function, called the activation, to produce the output". The activation function can be linear or nonlinear [1].

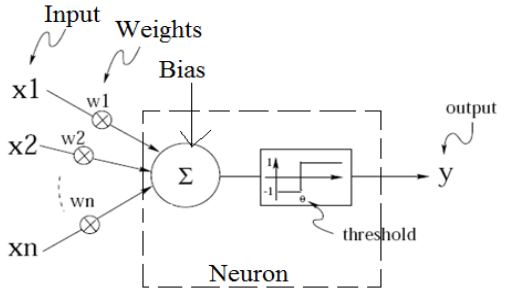

Fig -1: Block diagram of Artificial Neural Network [1]

Single Layer Perceptron can classify input vector into linearly separable classes. The perceptron convergence algorithm is used for training of Single Layer Perceptron.

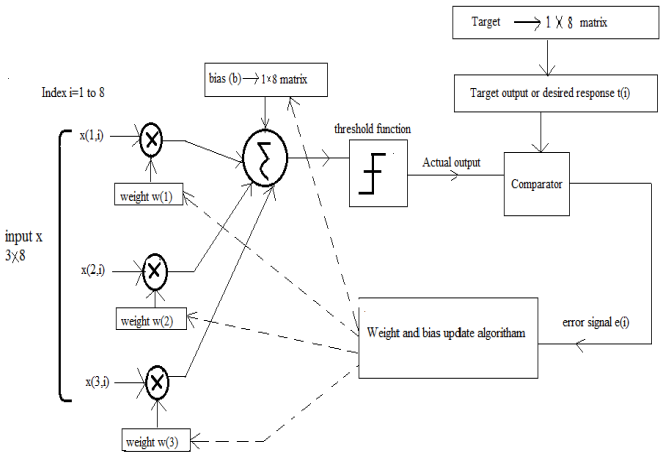

Fig -2: Block diagram of Single Layer Perceptron.

A figure shows the block diagram of Single Layer Perceptron which is designed for AND functions. Input of perceptron used for AND function is 3x8 matrix. Which is X=[1 1 1 1 -1 -1 -1 -1; 1 1 -1 -1 1 1 -1 -1;1 -1 1 -1 1 -1 1 -1]; Target matrix for AND function is T= [1 -1 -1 -1 -1 -1 -1 -1];
The updating of weights and bias process follows throughout

n algorithm which is known as perceptron convergence algorithm. Variables and Parameters are as below:

X(n)= (m+1)-by-1, input vector

= [+1, x1(n),x2(n),....xm(n)]^T

b(n)=bias

y(n)=actual response

d(n)=desired response

η =learning parameter, a positive constant less than unity.

Here, Perceptron gets input from input matrix and maps actual output (which is initially taken as zero) with target output. The goal is to minimize the generated error signal which is nothing but the difference between actual output and target output. In this paper, the error signal becomes zero within only four iterations for learning rate= 1 and threshold = 0. For varying learning rate and threshold the algorithm takes different number of epochs to converge.

2. Synthesis result

Here, Single Layer Perceptron for AND function is designed in MATLAB. It is synthesized in Xilinx and obtained VHDL code.

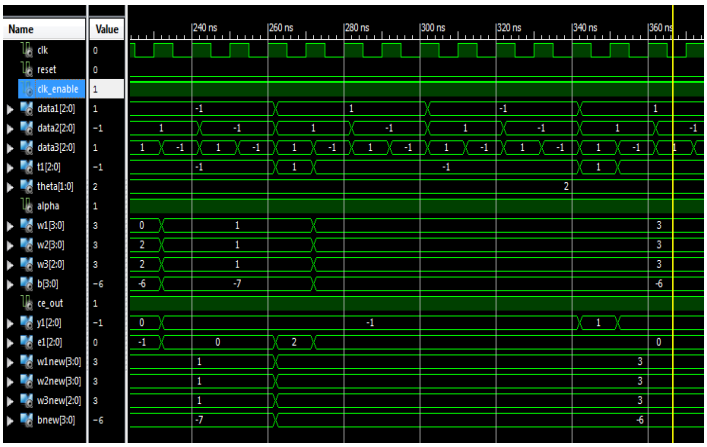

Fig- 3: Simulation result of Single Layer Perceptron for AND function

Here, 'data 1', 'data 2' and 'data 3' are the inputs. 't1'is target output of perceptron. 'y1' is the actual output, which is initially zero and after one epoch it becomes equals to target output. Here 'e1' is the error signal. 'Alpha' is learning rate which is set to 1 and 'theta' is threshold which is set to zero. 'w1', 'w2' and 'w3' as shown in fig 3.'b' is the bias (fixed output). 'w1new', 'w2new' and 'w3new', 'bnew' are updated values of weights and bias respectively. When weights are not updated error is present but when weights are updated error becomes zero.

Final weights are [w1, w2, w3] = [3 3 3]

Bias = [-6]

The RTL schematic of obtained result is shown in fig 4. It shows the pins and its functions.

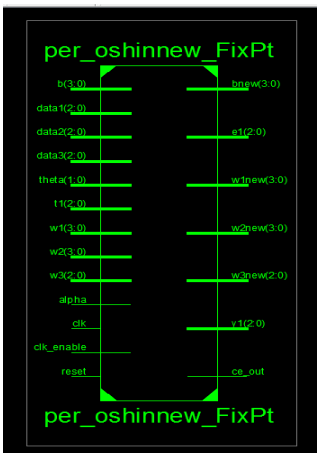

Fig -4: RTL schematics of Single Layer Perceptron for AND function (Top view)

3. Comparison with NNTOOL

MATLAB has Neural Network Design and Development Toolbox (NNTOOL) which is used extensively for designing of Neural Network. Using that block diagram of Single Layer Perceptron is obtained.

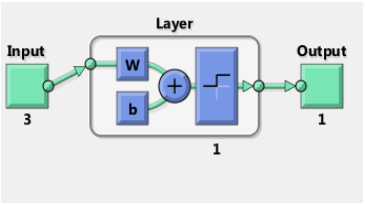

Fig -5: MATLAB generated block diagram of Single Layer Perceptron from NNTOOL

Single Neuron is used in this block diagram. The disadvantage of NNTOOL output is that more number of weights and bias are used. And even after 1000 epochs error is not zero.

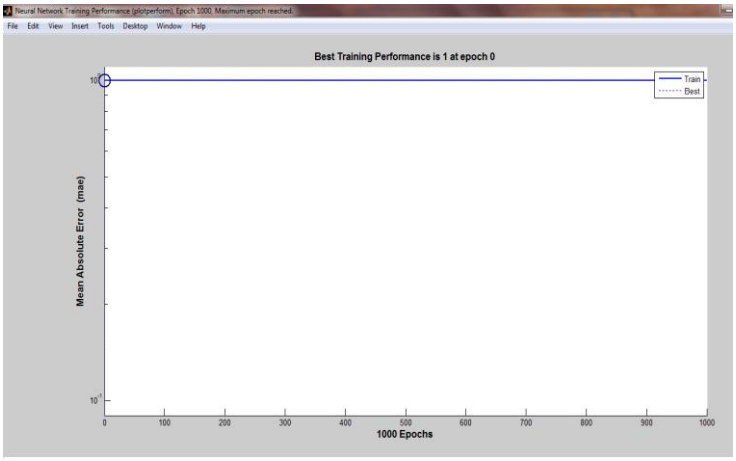

Fig- 6: Error graph generated from NNTOOL

In NNTOOL number of weights and bias generated are as shown.

Weight matrix after 1000 epochs= [6198 6198 6198];

Bias after 1000 epochs = [-18604];

The Mean Absolute Error is 1.

4. CONCLUSIONS

In this paper, Single Layer Perceptron is designed in MATLAB[13] using 3 input bits one output bit and single neuron. Here first, Single Layer Perceptron is designed in MATLAB then after MATLAB code is converted into VHDL by using HDL coder available in MATLAB[13]. We have synthesized in Xilinx and got the RTL schematics and corresponding clock waveforms.
We have overcome the disadvantage of NNTOOL (Neural Network Design and Development Toolbox) used for Neural Network in which more number of weights and bias are used and error is present even after 1000 epochs. Whereas in our algorithm same process is done by using only 3 weights and 1 bias within only 4 iteration and error has become zero.

REFERENCES

[1] Simon Haykin "Neural Networks A comprehensive Foundation" second edition, Prentice Hall, 2007, pp. 1-277.

[2] Abrol, Sahil, and Rita Mahajan. "Implementation of Single Artificial Neuron Using Various Activation Functions and XOR Gate on FPGA Chip." In Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on, pp. 118-123. IEEE, 2015.

[3] Shinde, Jitesh R., and S. Salankar. "Multi- objective optimization for VLSI implementation of Artificial Neural Network." In Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on, pp. 1694-1700. IEEE, 2015.

[4] Rani, S. P., and P. Kanagasabapathy. "Multilayer Perceptron Neural Network Architecture using VHDL with Combinational Logic Sigmoid Function." In Signal Processing, Communications and Networking, 2007. ICSCN'07. International Conference on, pp. 404-409. IEEE, 2007.

[5] Hu, Zhongzhong, Helei Wu, and Shuyun Zhu. "An Engineering Solution to Taylor Series Expansion Coefficients Based on BP Neural Network." In Internet Computing & Information Services (ICICIS), 2011 International Conference on, pp. 491-494. IEEE, 2011.

[6] Taright, Yamina, and Michel Hubin. "FPGA implementation of a multilayer perceptron neural network using VHDL." In Signal Processing Proceedings, 1998. ICSP'98. 1998 Fourth International Conference on, vol. 2, pp. 1311-1314. IEEE, 1998.

[7] Hardik H.Makwana, Dharmesh I. Shah, and Privesh P. Gandhi. "FPGA Implementation of Artificial Neural Network." International Journal of Emerging Technology and Advanced Engineering 3, no. 1 (2013).