iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VI **Month of publication:** June 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue V, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

FPGA Implementation of Xor Gate 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: ***In this paper, the XOR gate neural network is implemented on Field Programmable Gate Arrays (FPGA) . Firstly, one MAC (Multiply and accumulate) unit and a binary step activation function is developed in VHDL. These are then used in finite state machine used to implement the three layered XOR neural network and then the XOR gate is obtained. FPGA supports reconfigurable computing architectures and hence, are much more convenient for neural network hardware implementation. The neural networks being concurrent show very good speed for the computation of certain tasks. This is the reason why ANNs are preferred to be implemented in VLSI technology. A 3-layers 5-neurons model is used to design the XOR gate. However, the ANNs with large number of neurons is still difficult to be realized.***

Index Terms: ***Artificial Neural Networks (ANNs), Field Programmable Gate Array (FPGA), ADALINE and MADALINE, 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 ANNs gives slower execution in comparison to the hardware based ANNs. Analog and digital are two hardware based ANNs. [2]

ANNs consists of massive parallel network of neurons which are also known as parallel processing elements. [3]

The three basic features of ANNs are concurrency, modular behavior and dynamic approach. FPGA realization of ANNs having a large number of neurons is still a challenging task. [4]

II. ADALINE TRAINING RULE

ADALINE stands for Adaptive Linear Neuron. It makes use of bipolar activations (+1 and -1) for its input signals and its final output. The weights on the connections on input signals to the ADALINE are adjustable. This rule is used for single layered nets with many output units, ADALINE is its special case in which output uit is single. Identity function is the activation function during training . The mean squared error between activation and target value is reduced by this learning rule. This helps the net to continue learning on all training patterns even after the correct output is obtained for some patterns. [5]

III. MADALINE TRAINING RULE

The combination of Adalines is called as MADALINE and sometimes multilayered Adaline. If the adalines are joined in such a way that output of one adaline becomes input to the other adaline, then the network becomes multilayered. This leads to MADALINE. It includes Many Adaptive Linear Neurons which are arranged in multilayer net.It consists of two training algorithms: MRI and MRII. [6]

The output of the MADALINE is given by

$Y= f(y_{in})$

where $f(y_{in})$ stands for the activation function and is given as follows: [7]

Volume 5 Issue V, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

$$f(y_{\_in}) = \begin{cases} 1 & \text{if } y_{\_in} \geq 0 \\ -1 & \text{if } y_{\_in} < 0 \end{cases}$$

IV. DATA REPRESENTATION

Before starting the hardware implementation of an ANN, a number format must be considered for the inputs, weights and activation functions.[8]

To represent the inputs and weights, 7 bit format is used. +16 is the weight value for +1 and -16 is the weight value of -1.Similarily, the remaining weights are taken. Using this, the circuit is synthesized and simulation is carried out.

V. IMPLEMENTATION APPROACH

The XOR problem is implemented using a 3 layer, 5 neuron multilayer MADALINE model of Artificial Neural Network as shown in fig 1. Firstly, the network is initialized and random values are given to the weights (between -1 and +1). The weighted sum of the input values at the hidden neurons is calculated. Then, the obtained value is run through hard limiter function. The outputs of the hidden layer are taken as the inputs and the same procedure is followed for the output layer. The delta values for the output and the hidden layer is calculated. The weights are then updated using delta values. The learning parameter (ŋ) is taken as 0.5. This is done in MATLAB. These weights are then used in VHDL to implement the XOR gate. The values are approximated to the nearest integers and mid-integer values so that these can be easy to be implemented.[9]

Firstly, a MAC unit is designed in VHDL and a binary hardlimiter function is developed. [10]

These are then called repeatedly in the main program for the summation and accumulation purpose and to obtain the required result. It then includes various states which do each multiplication and summation operation be the MAC unit. Implementation in VHDL is beneficial for the XOR neural net since it exploits the concurrent behavior of neural net.

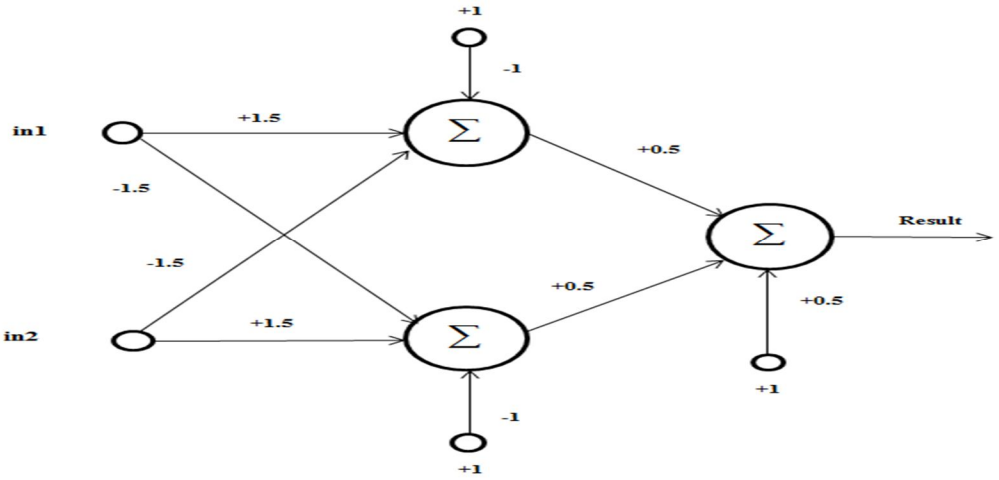


Figure 1. MADALINE model for XOR gate

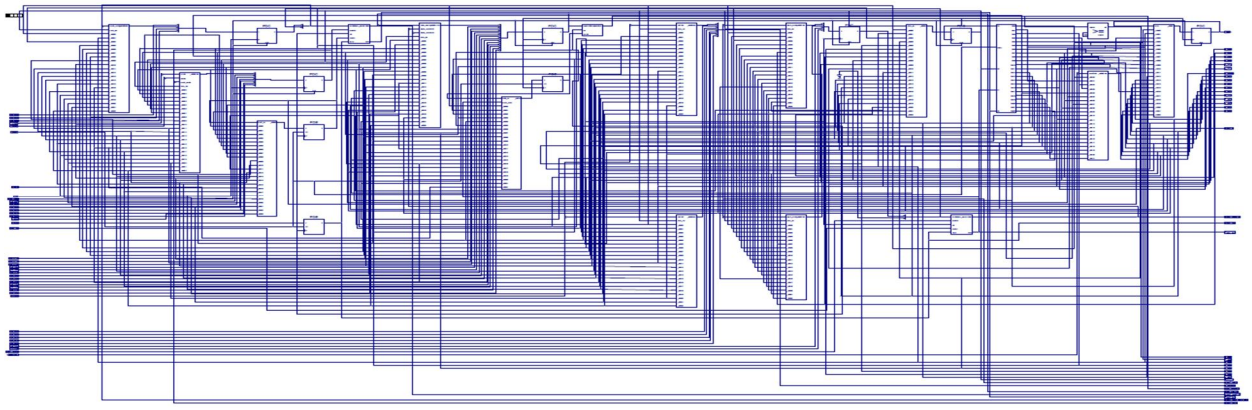


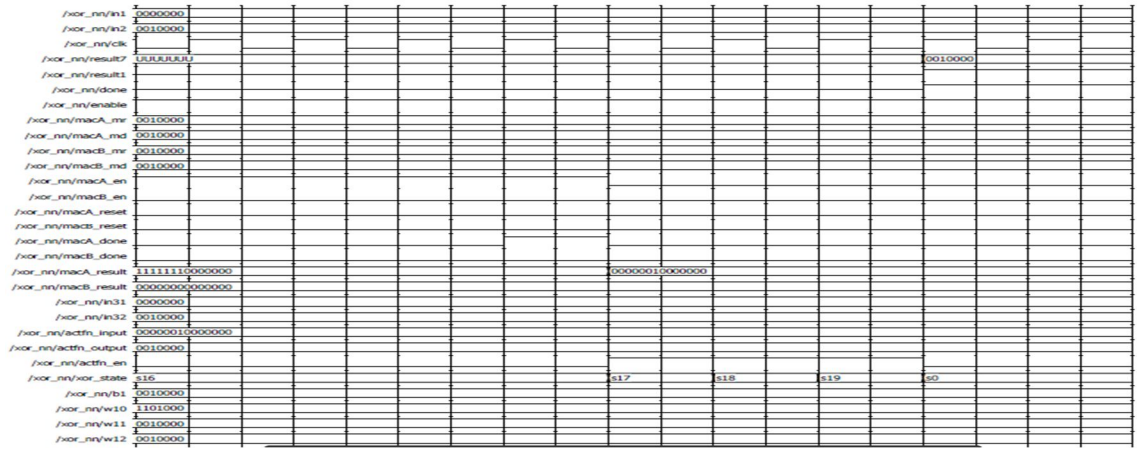
Figure 2. RTL Schematic of XOR gate

Volume 5 Issue V, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

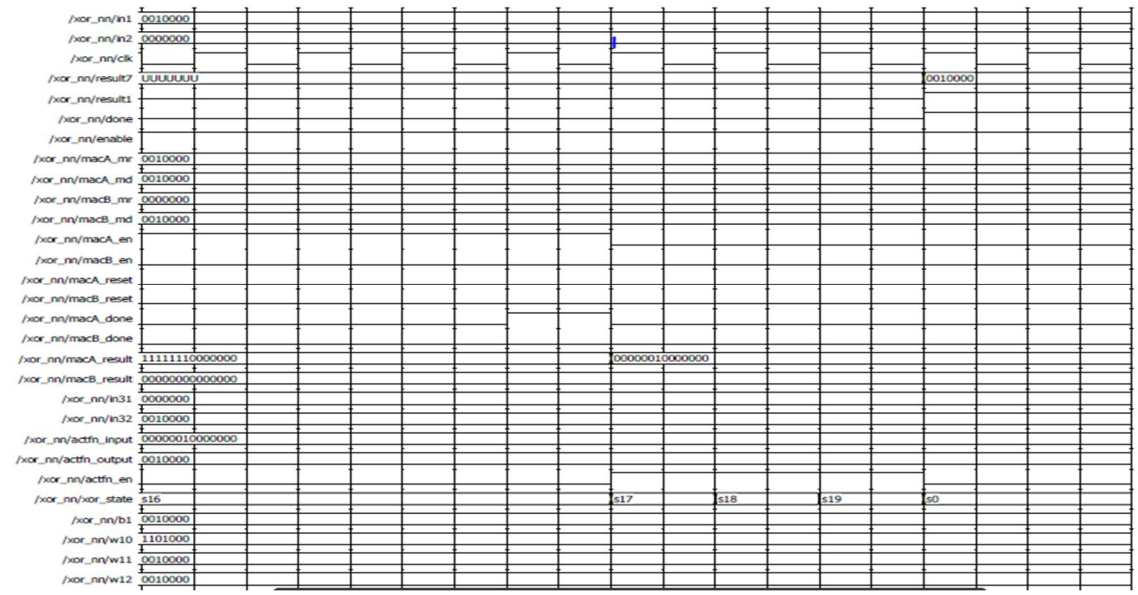
VI. RESULTS AND DISCUSSIONS

The inputs are given the values 00, 01, 10 and 11 and the final results are obtained which are as follows:

A. *Simulation results for input 00*

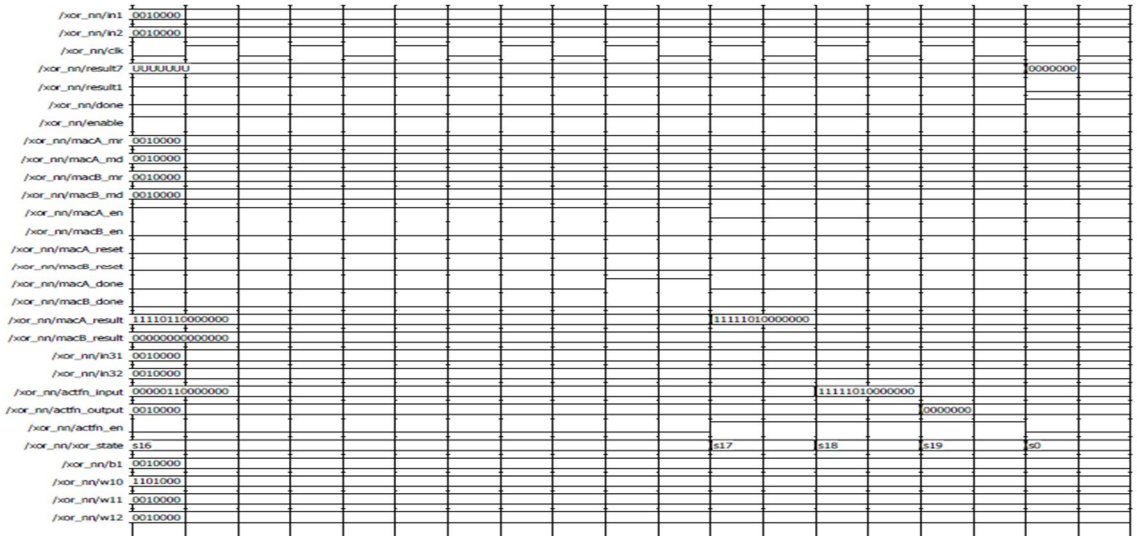


B. *Simulation results for input 01*



Volume 5 Issue V, June 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

C. Simulation results for input 10



D. Simulation results for input 11

LOGIC UTILIZATION	USED	AVAILABLE	UTILIZATION (%)
Number of slices	222	192	115
Number of slice flipflops	219	384	57
Number of 4 input LUTs	412	384	107
Number of bonded IOBs	24	90	26
Number of GCLKs	1	4	25

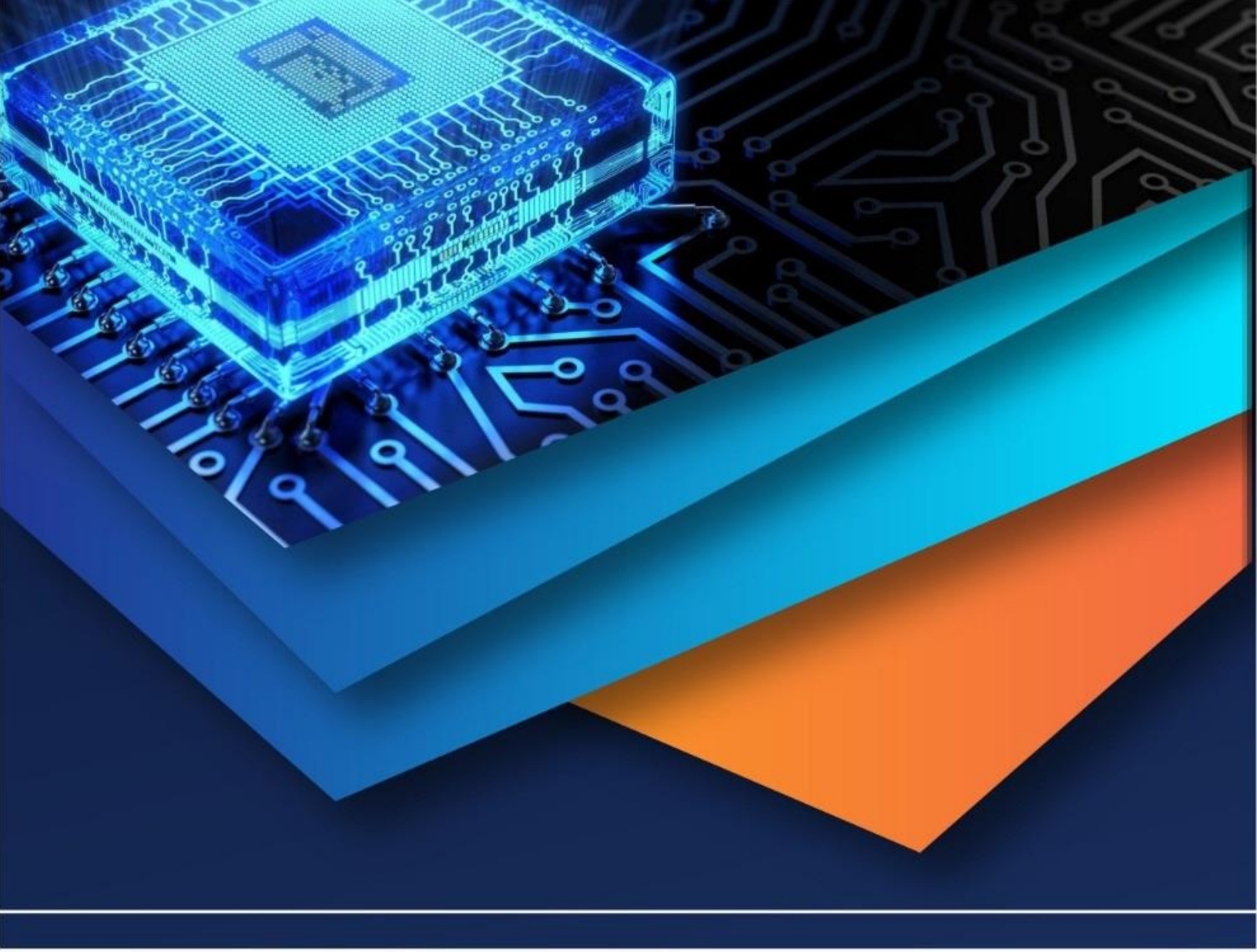
Table No. 1 Design Summary

VII. CONCLUSION

Multilayered ADALINE for XOR gate is implemented on Xilinx 2s15cs144-6. The design summary shows the results for the number of slices, flipflops, 4 input LUTs, bonded IOBs and GCLKs used by the XOR gate neural net. The final results for each of the combination of inputs are obtained and analyzed. Since neural networks are inherently parallel structures and hence are always faster than the serial ones.. It is found that FPGA implementation of neural networks is best option among the remaining. FPGAs are versatile devices for implementing different applications. FPGA-based reprogrammable computing architectures are suitable for hardware implementation of neural networks.

REFERENCES

[1] Sahin S, Becerikli Y, Yazici S, Neural network implementation in hardware using FPGAs, 2006, International Conference on Neural Information Processing, 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.
[3] Makwana H.H., Shah D.J. and Gandhi P.P., 2013, FPGA Implementation of Artificial Neural Network. International Journal of Emerging Technology and Advanced Engineering, 3(1), pp.672-679.
[4] Muthuramalingam, A., Himavathi, S. and Srinivasan, E., 2008. Neural network implementation using FPGA: issues and application. International journal of information technology, 4(2), pp.86-92.
[5] Sivanandam, S.N. and Deepa, S.N., 2006, Introduction to neural networks using Matlab 6.0.,Tata McGraw-Hill Education.
[6] S. Haykin, neural networks-a comprehensive foundations Second Edition, ISBN: 0132733501, 1998
[7] Fausett L., 1994, fundamentals of neural networks: architectures, algorithms, and applications Prentice-Hall, Inc.
[8] Savran, A. and Ünsal, S.December, 2003, Hardware implementation of a feed forward neural network using FPGAs, The third International Conference on Electrical and Electronics Engineering (ELECO 2003), pp. 3-7.
[9] Gadea, R., Cerdá, J., Ballester, F. and Mocholí, A., September, 2000, Artificial neural network implementation on a single FPGA of a pipelined on-line backpropagation. In Proceedings of the 13th international symposium on System synthesis IEEE Computer Society, pp. 225-230.
[10] Abrol S. and Mahajan R., 2015, May. 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 IEEE, pp. 118-123.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY