Artificial Neural Network with Hardware Chip Implementation

[1]Km Manisha, [2]Dr. A.K. Gautam

[1]M.Tech Scholar, [2]Principal
Department of Electronics & Communication Engineering
S. D College of Engineering and Technology, Muzaffarnagar, U.P. India

***Abstract*: An Artificial Neural Network (ANN) system is based on a group of associated units or nodes in a biological brain called neurons. The signal is transferred from one neuron to another neuron. The ANN model accepts the input signal which are multiplied with their weights. The multiplied output is weighted sum with the bias component and processed with the nonlinear signals. In the research work we have considered the 8 inputs ANN signal which are multiplied with their corresponding weights. The hardware chip is designed to support the system functionality in Xilinx ISE 14.2 software. The designed chip is simulated with Modelsim 10.0 software for test cases. The designed chip is also synthesized on SPARTAN-3E FPGA and hardware and timing parameters are also analyzed.**

***Keywords*: Artificial Neural Network (ANN), FPGA, Xilinx ISE software**

1. Introduction

An Artificial Neural Network (ANN) [1, 2] is a computing system paradigm based on biological neuron networks [3, 4], such as the brain, process information. The main component of this paradigm is the innovative structure of the information processing system. The system consists of the extremely interrelated processing elements called neurons work in unison to resolve some explicit problems. The ANN is also learning system, as the people learn by the examples. The ANN based system is configured for a specific application, such as data classification or pattern recognition [5] with the help of learning process. The learning behavior in biological systems comprises changes to the synaptic connections that exist between the neurons. An ANN architecture is based on a gathering of associated neurons which forms a biological brain. The interconnected system can transmit the signal from one artificial neuron [7, 8] to another like synapses called artificial neurons which loosely model the neurons in a biological brain. Each connection of the biological brain [6] like the synapse that can transfer a signal from one artificial neuron to another artificial neuron. The artificial neuron that accepts these signals should be capable to process the signals and additionally signals of artificial neurons associated to it.

1.2 ANN Architecture

The neural architecture consists of the input signals, weight coefficients [2, 9, 10] and activation function. The fig.1 presents the architecture of neural network in which 8 inputs are given as P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8. The corresponding weights of the inputs are given as W_1, W_2, W_3, W_4, W_5, W_6, W_7 and W_8. The inputs are multiplied with their weight confidents and weight sum [6, 11] is added with another bias input (b) [12, 13].In the same way the ANN architecture can be extended [14, 15] for 'N' inputs. For the activation function $y = f(x)$ the output is expressed as,

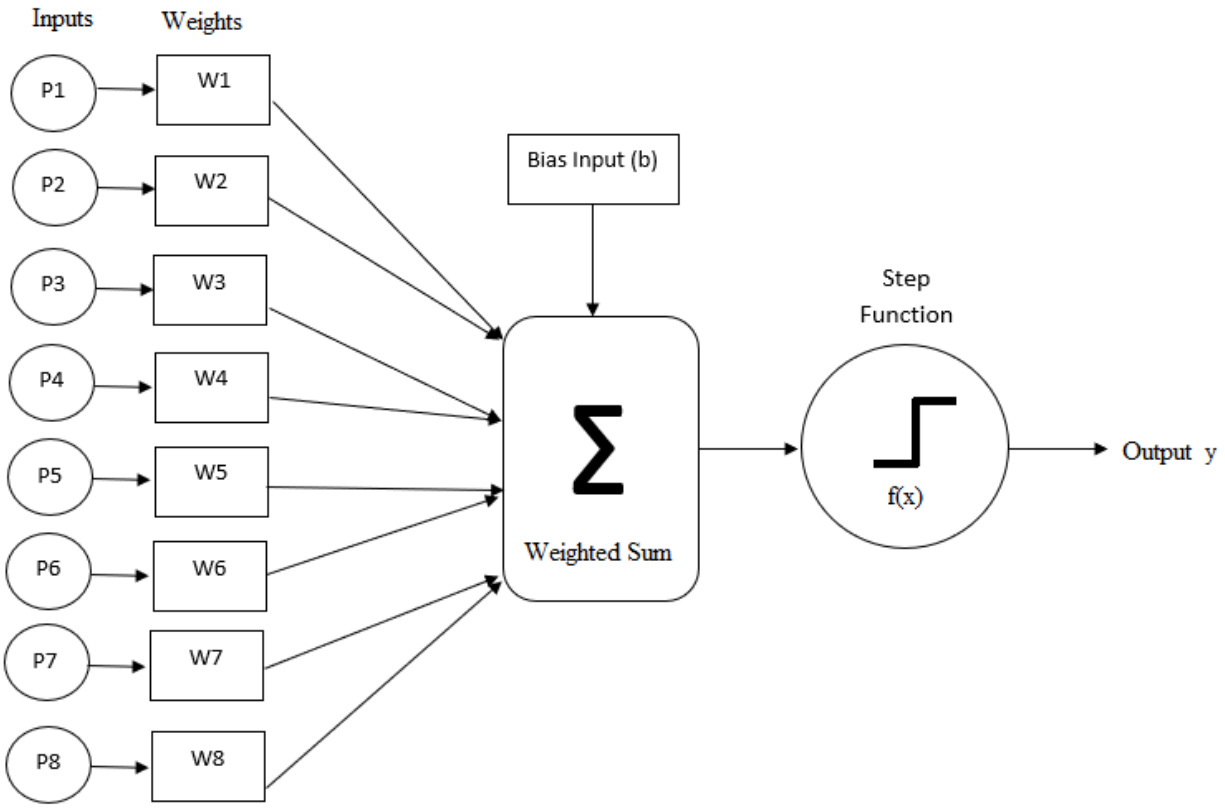

Fig. 1 Neural Network with 8 inputs

$$y = \sum_{i=1}^{n} P_i W_i + (bias) \qquad (1)$$

For 8 inputs

$$y = \sum_{i=1}^{8} P_i W_i + (b) \qquad (2)$$

$$y = P_1W_1 + P_2W_2 + P_3W_3 + P_4W_4 + P_5W_5 + P_6W_6 + P_7W_7 + P_8W_8 \qquad (3)$$

3. Results & Discussions

The Xilinx software simulation results for 8 input ANN Architecture is shown in Fig 2 that presents the RTL level view of the developed chip. The RTL provides the details of all the pins used for the design of the chip.

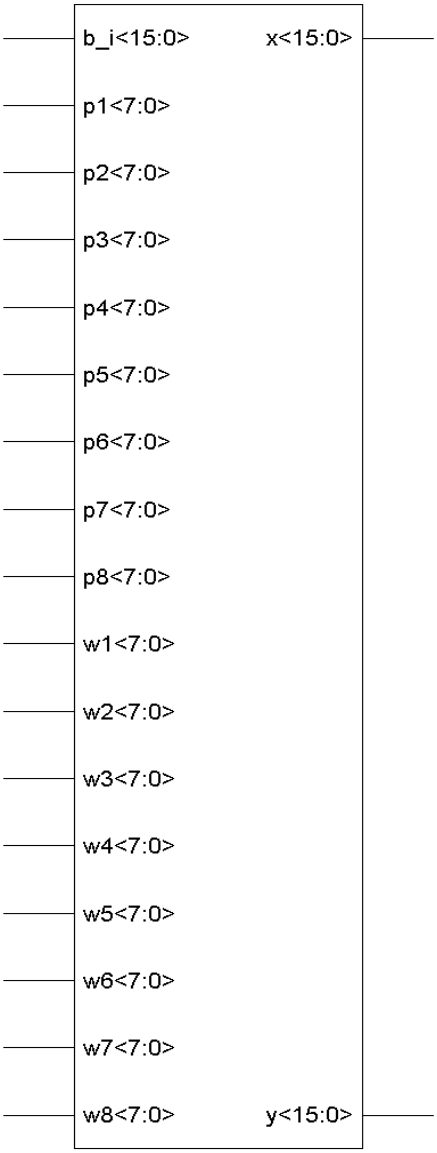

Fig. 2 RTL View of 8 point ANN Architceure Chip

Test-1 : P1<7:0> = "00000011" in binary = 3 in decimal, P2<7:0> = "00000100" in binary = 4 in decimal, P3<7:0> = "00000101" in binary = 5 in decimal, P4<7:0> = "00000110" = 6 in decimal, P5<7:0> = 00000101" in binary = 5 in decimal, P6<7:0> ="00000100" in binary = 4 in decimal, P7<7:0> = "00000011" in binary = 3 in decimal, P8<7:0> = "00000010" in binary = 2 in decimal, W1<7:0> = "00000001" in binary = 1 in decimal, W2<7:0> = "00000010" in binary = 2 in decimal W3<7:0> ="00000011" in binary = 2 in decimal, W4<7:0> ="00000010" in binary = 2 in decimal, W5<7:0> = "00000010" = 2 in decimal, W6<7:0> = "00000001" = 1 in decimal, W7<7:0> = "00000001" = 1 in decimal, W8<7:0> = "00000010" = 2 in decimal, b_i<15:0> = "0000000110111100" = 220 in decimal, Then output will be Y <15:0> = "0000000100010111" = 279 in decimal.

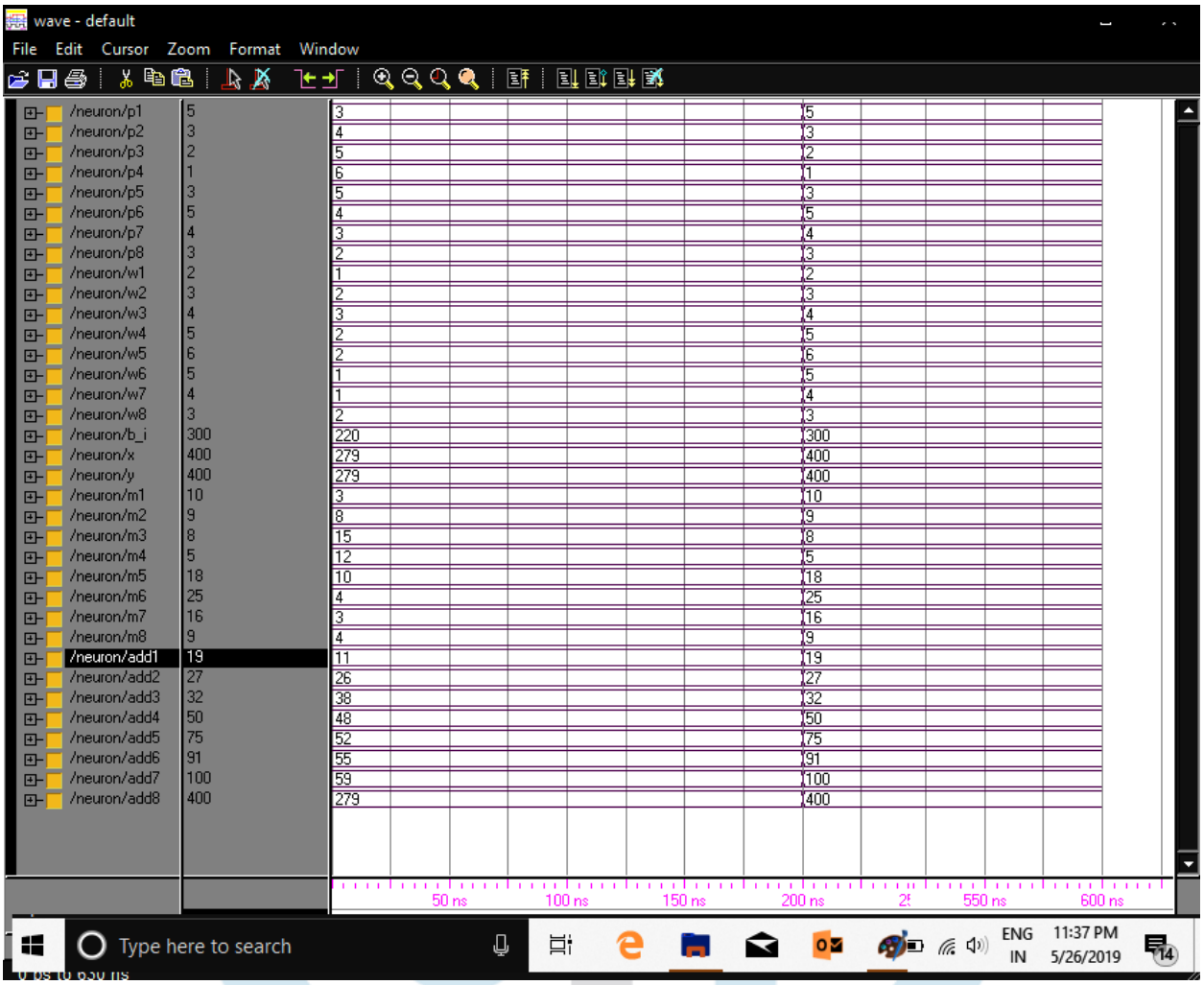

Fig. 3 Modelsim simulation of 8 input ANN with test-1 and test-2in integer

Test-2 : P1<7:0> = "00000101" in binary = 5 in decimal, P2<7:0>"00000011" in binary = 3 in decimal, P3<7:0> = "00000010" in binary = 2 in decimal, P4<7:0> = "00000001" = 1 in decimal, P5<7:0> = "00000011" in binary = 3 in decimal, P6<7:0> = 00000101" in binary = 5 in decimal, P7<7:0> = "00000100" in binary = 4 in decimal, , P8<7:0> = "00000011" in binary = 3 in decimal , W1<7:0> = "00000010" in binary = 2 in decimal, W2<7:0> = "00000011" in binary = 3 in decimal W3<7:0> = "00000100" in binary = 4 in decimal, W4<7:0> ="00000101" in binary = 5 in decimal, W5<7:0> = "00000110" = 6 in decimal, W6<7:0> = "00000101" in binary = 5 in decimal, W7<7:0> = "00000100" in binary = 4 in decimal, W8<7:0> = "00000011" = 3 in decimal, b_i<15:0> = "0000000100101100" = 300 in decimal, Then output will be Y <15:0> = "0000001100100000" =400 in decimal.

4. Device Hardware Detail

The percentage of hardware that is used by the device is given by device utilization report for the implementation of chip. Device hardware includes No. of slices. of input LUTs, No. of bounded IOBs and No. of gated clocks (GCLKs) used in design implementation. Timing details provides the knowledge of maximum frequency and combinational delay. To complete the design, the total memory utilization required is also given.The Xilinx software gives the details of the hardware utilization for 8 input ANN. Table 1 presents the utilization report of the hardware parameters.

Table 1 Hardware summary for 8 input ANN for SPARTAN 3E FPGA

Parameter	**ANN(8 input)**
Number of Slices	355
Number of Slice flipflops	418
Number of LUTs	648
Number of IoBs	178
GCLKs	1
Memory Usage	116688 kB

Table 2 Timing Parameters for 8 input ANN for SPARTAN 3E FPGA

Parameter	**ANN(8 input)**
Frequency (MHz)	235 MHz
Min Period (ns)	1.921 ns
Min time before clock signal (ns)	1.982 ns
Max Time after clock signal(ns)	3.517 ns
Combinational Delay (ns)	4.518 ns

5. Conclusions

The hardware chip of 8 input ANN architecture is designed in Xilinx ISE 14.2 software successfully and simulated in Modelsim 10.0 software. The designed chip is synthesized on SPARTAN-3E FPGA. The designed chip support the frequency of 235 MHz. The timing detail parameters are minimumperiod (ns), minimum time before clock signal (ns), maximumtime after clock signal(ns) and combinational delay (ns) which are estimated as 1.921 ns, 1.982 ns, 3.517 ns and 4.518 ns respectively.

References

[1] Alcin, M., Koyuncu, I., Tuna, M., Varan, M., &Pehlivan, I. (2019). A novel high speed Artificial Neural Network–based chaotic True Random Number Generator on Field Programmable Gate Array. *International Journal of Circuit Theory and Applications*, *47*(3), 365-378.

[2] Bose, B. K. (2007). Neural network applications in power electronics and motor drives—An introduction and perspective. *IEEE Transactions on Industrial Electronics*, *54*(1), 14-33.

[3] Carvalho, M. B., Amaral, A. M., da Silva Ramos, L. E., da Silva Martins, C. A. P., &Ekel, P. (2005, December). Artificial neural network engine: Parallel and parameterized architecture implemented in FPGA. In *International Conference on Pattern Recognition and Machine Intelligence*(pp. 294-299). Springer, Berlin, Heidelberg.

[4] Himavathi, S., Anitha, D., & Muthuramalingam, A. (2007). Feedforward neural network implementation in FPGA using layer multiplexing for effective resource utilization. *IEEE Transactions on Neural Networks*, *18*(3), 880-888.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[6] Joseph, C., & Gupta, A. (2010, February). A novel hardware efficient Digital Neural Network architecture implemented in 130nm technology. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on* (Vol. 3, pp. 82-87). IEEE.

[7] Kim, C. T., & Lee, J. J. (2008). Training two-layered feedforward networks with variable projection method. *IEEE Transactions on Neural Networks*, *19*(2), 371-375.

[8] Köppen-Seliger, B., & Frank, P. M. (1996). Neural networks in model-based fault diagnosis. *IFAC Proceedings Volumes*, *29*(1), 6389-6394.

[9] Li, Z., Li, H., Jiang, X., Chen, B., Zhang, Y., & Du, G. (2019, April). Efficient FPGA Implementation of Softmax Function for DNN Applications. In *2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)* (pp. 212-216). IEEE.

[10] Li, H., Zhang, D., & Foo, S. Y. (2006). A stochastic digital implementation of a neural network controller for small wind turbine systems. *IEEE transactions on power electronics*, *21*(5), 1502-1507.

[11] Misra, J., &Saha, I. (2010). Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomputing*, *74*(1-3), 239-255.

[12] Maeda, Y., &Wakamura, M. (2005). Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation. *IEEE Transactions on Neural Networks*, *16*(6), 1664-1672.

[13] Reyneri, L. M. (2003). Implementation issues of neuro-fuzzy hardware: going toward HW/SW codesign. *IEEE Transactions on Neural Networks*, *14*(1), 176-194.

[14] Wilamowski, B. M. (2009). Neural network architectures and learning algorithms. *IEEE Industrial Electronics Magazine*, *3*(4).

[15] Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015, February). Optimizing fpga-based accelerator design for deep convolutional neural networks. In *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (pp. 161-170). ACM.