Journal of Engineering Technology
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 100-105
ORIGINAL ARTICLE

Implementation of a digital neuron using system verilog

Azhar Syed and Vilas H Gaidhane

Department of Electrical and Electronics Engineering, BITS Pilani Dubai Campus, DIAC Dubai-345055, UAE

Abstract: Introduction Artificial neural network is an advanced field which has great applications in almost all real life applications. The neuron is a building block of neural network. Therefore the design of hardware of such neuron is a challenging task for the researchers. **Methodology** In this paper, a simple and efficient digital neuron architecture is presented. It is based on the System Verilog approach. The sigmoidal activation function is considered in the design. All the building blocks of the neuron implemented and simulated independently. **Results** Various examples are considered to check the accuracy of the presented hardware architecture. The output of the neuron is calculated using the theoretical calculations as well as system Verilog simulation. It is observed that the proposed approach gives the best results. **Conclusion** The presented neuron hardware is less complex as compared to the conventional neuron architecture. The various simulations show the accuracy of the presented approach. It can be used in the implementation and design of the VSLI circuits based neural network design.

Keywords: Digital Neuron, Sigmoidal function, Artificial Neural Network, System Verilig.

1. Introduction

Artificial Intelligence is one of the constantly evolving and an actively growing field. Artificial Neural Networks (ANN) play a vital role in solving complex engineering problems by improving their computing capability. It is a computational network that emulates the highly complex and parallel computing capability of a biological neural network. There are certain tasks that are effortless for a human brain. However, it becomes extremely challenging for researchers during the automation process. During the early stage of development of the neural network it was unthinkable for computer to the tasks such as object recognition, pattern identification, data mining and classification, etc., [1].

ANNs are the front-runners in the world of computations that are developed to mimic the intelligent behavior. A neural network is highly comprised of huge numbers of neurons working in union for problem solving. A network is a term that is defined as a group of interconnected things which contains nodes and edges. Each edge is connected to a node commonly called as neuron. The different computations are performed at every node and information is passed to the next neuron through these edges. Each edge has a weighted value associated with it which helps in the learning process of the neural networks. This learning process is similar to humans, a child learns to recognize a tree by looking at examples of trees. There are three approaches which can be used to implement an artificial neural network architecture, they are analog, digital and mixed-signal approach. In the analog approach the sigmoid activation function is obtained using BJT or BiCMOS technologies. However, it becomes difficult to implement the same using CMOS technology as it requires certain approximations. Moreover, the analog and mixed signal implementations also suffer from problems of drift, noise sensitivity and quantization effects mainly caused by MOS geometries. The digital implementation of the architecture is drift free and relatively noise immune. However, the digital implementation is limited by physical constraints which are dependent on the chosen technology. In

Journal of Engineering Technology
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 100-105
ORIGINAL ARTICLE

particular, for a digital implementation the major concerns are power utilization, and the complexity. The solution to these concerns have been reported in the literature [2, 3].

In this paper, a single digital neuron using System Verilog is implemented which can be used to construct a complex digital artificial neural network. The activation function is implemented by reducing the hardware complexity and hence reducing the area and overall power utilization.

2. System Verilog-Based Design

System Verilog was adopted as an IEEE standard in 2005. It incorporates object oriented programming techniques which are closely related to Java. It has introduced the enhanced variable types such as 'logic' and 'bit' which add new capabilities in the design. Moreover, it provides the superior features for system architecture, design and verification. System Verilog helps us to design the behavior more concisely than the conventional methods. In System Verilog, all ports and signals can be defined as logic data type and the language will correctly infer these declarations as nets or variables.

System Verilog introduces three new procedural blocks which are, always comb, always latch and always ff. These procedural blocks prevent modeling errors. These blocks help the software tools to verify the design intent. It has been recommended to use the newly introduced procedural blocks in all RTL code. The general purpose block is to be used in models that are not intended to be synthesized.

3. Activation Function

In practical scenarios multilayer neuron networks are used in which each neuron of a particular layer gets the same type of activation function. The various types of activation functions used in a neural network are identity function, binary sigmoid function, step function and bipolar sigmoid function. The activation function acts as an identity function for the input unit. The sigmoid function is most commonly used as it is smooth, continuous, derivative of the signal is always positive and monotonically increasing in nature. It has a bounded range, but never reaches max or min. In this paper binary sigmoid function is used as the activation function. The basic architecture of a neuron is shown in Fig. 1. It sums up the products of the input signals and its associated weight and generate an output or an activation function. The sigmoid function is represented as

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

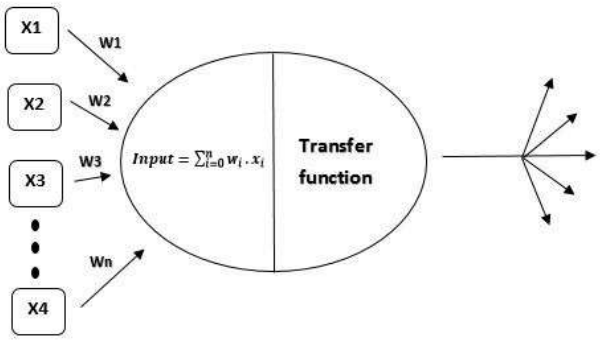

Figure. 1. Basic neuron structure.

Journal of Engineering Technology
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 100-105
ORIGINAL ARTICLE

The sigmoid function can be implemented using different techniques such as piecewise linear approximation, Taylor series and lookup tables. In this paper, we have used Co-Ordinate Rotation, Digital Computer (CORDIC) algorithm to implement the sigmoid function. Taylor series expansion takes up large silicon areas as it requires implementation of multipliers to expand up to five terms. The look-up-table technique requires the y-value associated with the corresponding x-value to be stored in memory. This technique requires a large silicon area when we need a high precision. It is practically not feasible to include single look-up table per processing element in a neural network. The piecewise linear approximation is a direct calculation approach in which the output is approximated by a combination of straight lines. The CORDIC scheme used in this paper is a modification of the CORDIC algorithm proposed by Volder. The scheme uses an iterative method which utilizes low precision arithmetic components and then corrects the approximation error periodically in order to get high precision computation at near low precision speed [6-8]. The data provided to obtain a high precision sigmoid output needs to be transformed. The System Verilog data type is real and 64 bit in size. In order to achieve the high precision up to 64 bits we make use of the 64-bits precision table [7].

4. Neuron Architecture

In recent years, the various neuron architecture is presented in the literature. Fig. 2 shows one of the conventional neuron architecture [3]. Such architectures are comprised of buffer accumulator, multiplexers, multiplier, register and a finite state machine (FSM). The workings of all the blocks are self explanatory. Moreover, the FSM plays an important role in generating the signal pulses at different time intervals. It also assures coordination between the different operations realized by neuron and therefore the more time is required to obtain the result. Hence the system become slow and complicated. To overcome this issue a new architecture is required.

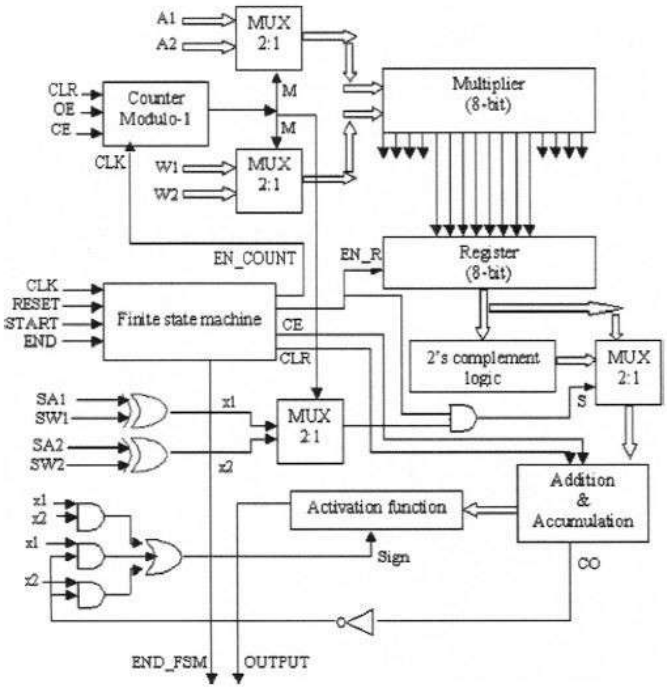

Figure. 2. Basic Neuron Architecture [3].

Journal of Engineering Technology
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 100-105
ORIGINAL ARTICLE

In this paper, a modified architecture is proposed which reduces the overall hardware used in conventional architecture. It is possible due to the use of System Verilog HDL for architecture design. The proposed architecture comprises of two multiplexers instead of three multiplexer as shown in Fig. 3.

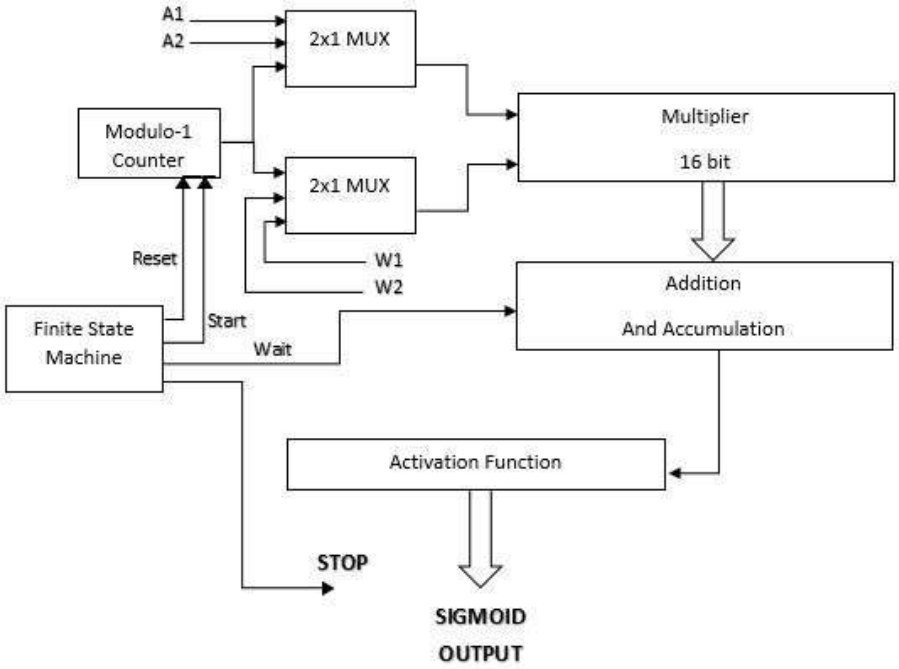

Figure. 3. An improved proposed Architecture of the neuron.

The other blocks include a finite state machine (FSM), multiplier, counter, accumulator and activation function. Signals A1, A2 represents the 8-bits input, W1 and W2 represent the 8-bits synaptic weights of the neuron. The FSM dictates the entire working of the neuron. The CLK signal of the FSM is the system clock. The High RESET signal will reset the entire neuron to the starting condition. The START signal initiates the neuron functioning after the RESET signal is low. It is very important that signals and their corresponding weights are processed in sync. This synchronization between the signals and their weights is obtained with the help of modulo-1 counter and two 2X1 multiplexers as we are using only two inputs, similarly for three inputs we will have to use modulo-2 counter. The multiplier used in this architecture is a 16-bit multiplier. The 16-bit multiplier output is given to the accumulator block. The accumulator block takes the multiplied result of the two signals with their corresponding weights and adds them. The output of the accumulator block is given to the activation function block to obtain the equivalent sigmoidal output. On obtaining the sigmoidal output the finite state machine gives STOP signal indicating the completion of the process for one set of inputs and their corresponding weights [3, 9,10]. The neuron architecture presented in [3] uses of three AND gates and one OR gate logic to remember the sign of the number. The architecture proposed in this paper does not include additional circuitry to record the sign of the accumulated number received from the accumulation block. This additional hardware is avoided due to the use of System Verilog HDL for designing the neuron architecture. The System Verilog HDL provides us with a data type **real**, which is used to work with signed numbers [4].

Journal of Engineering Technology
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 100-105
ORIGINAL ARTICLE

5. Simulation and Results

Mentor Graphics Modelsim 10.4c is used as a System Verilog simulation tool to test proposed neuron architecture. All blocks shown in Fig. 3 simulated and tested individually. The following illustrate the operation and result of the presented neuron architecture. The neuron is tested for many sample values.

Example 1 : $a_1 = 0.6875,\ \ a_2 = 1, w_1 = 1, w_2 = 1$

$$x = a_1 \times w_1 + a_2 \times w_2$$
$$x = 0.6875 \times 1 + 1 \times 1 = 1.6875$$
$$y = \frac{1}{1 + e^{-x}}$$
$$y = 0.843895$$

Example 2 : $a_1 = 0.3125,\ \ a_2 = 1, w_1 = 1, w_2 = 1$

$$x = a_1 \times w_1 + a_2 \times w_2$$
$$x = 0.3125 \times 1 + 1 \times 1 = 1.3125$$
$$y = \frac{1}{1 + e^{-x}}$$
$$y = 0.787931$$

The above examples are simulated using the System Verilog and simulated results are shown in Fig. 4. The theoretical value of the neuron output in example 1 and example 2 is $y = 0.843895$ and $y = 0.787931$, respectively. It is observed from Fig. 4 that the theoretical and simulated values of neural output are exactly same.

In literature [3], the similar example has been considered in the analysis of the neuron architechture and obtained the theoretical results. However, the simulated results are deviated from the theoretical results (y = 0.83593). Hence the proposed architecture is more efficient than the reported method in the literature [3].

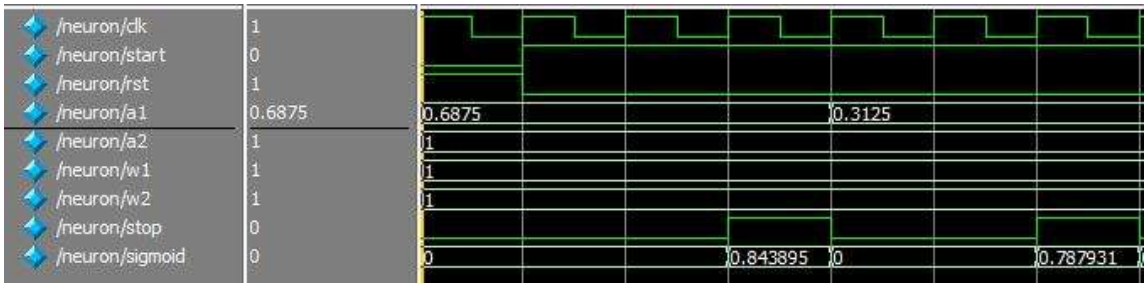

Figure. 4. Simulated output of sigmoid activation function.

6. Conclusion

In this paper, a digital neuron hardware architecture is presented. The detailed functional description is based on the System Verilog. The presented neuron hardware is less complex as compared to the conventional neuron architecture. The various simulations show the effectuality and the accuracy of the presented approach. It can be used in the implementation and design of the VSLI circuits based neural network design.

References

[1] Shahin. MA, Mark. BJ, Holger. RM, "Recent advances and future challenges for artificial neural systems in geotechnical engineering applications". Advances in Artificial Neural Systems, 2009; l(1): 1-9.

[2] Volanis. G, Antonopoulos. A, Hatzopoulos. AA, Makris. Y, "Toward Silicon-Based Cognitive Neuromorphic ICs—A Survey". IEEE Design and Test, 2016; 33(3): 91-102.

[3] Mishra. A, Raj. K, "Implementation of a digital neuron with nonlinear activation function using piecewise linear approximation technique". In Proc. IEEE International Conference on Microelectronics, 2007, 69-72.

[4] Sutherland. S, Mills. D, "Synthesizing system verilog busting the myth that system verilog is only for verification". SNUG Silicon Valley, 2013.

[5] Sibi. P, Jones, SA, Siddarth. P, "Analysis of different activation functions using back propagation neural networks". Journal of Theoretical and Applied Information Technology, 2013; 47(3): 1264-1268.

[6] Gaidhane. VH, Hote. YV, Singh. V, "Emotion recognition using eigenvalues and Levenberg-Marquardt algorithm based-classifier". Sadhana, 2016; 41(4): 415-423.

[7] Kantabutra. V, "On hardware for computing exponential and trigono-metric functions". IEEE Trans. on Computers, 1996; 45(3): 328-339.

[8] Leboeuf. K, Muscedere. R, Ahmadi. M, "Performance analysis of table-based approximations of the hyperbolic tangent acti- vation function". in Proc. IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), 1-4, 2011.

[9] Myers. DJ, Hutchinson. RA, "Efficient implementation of piecewise linear activation function for digital VLSI neural networks". *Electronics Letters*, 1989; 25(1):1662.

[10] Haghiri. S, Ahmadi. A, Saif. M, "Complete neuron-astrocyte interaction model: digital multiplierless design and networking mechanism". IEEE Trans. on Biomedical Circuits and Systems, 2017; 11(1): 117-127.