New artificial neural network design for Chua chaotic system prediction using FPGA hardware co-simulation

Wisal Adnan Al-Musawi, Wasan A. Wali, Mohammed Abd Ali Al-Ibadi
Department of Computer Engineering, College of Engineering, University of Basrah, Basrah, Iraq

Article Info

Article history:

Received Mar 22, 2021
Revised May 25, 2021
Accepted Jul 19, 2021

Keywords:

Artificial neural network
Chua's circuit
Floating point format
Hardware co-simulation
encryption and decryption
Xilinx system generator

ABSTRACT

This study aims to design a new architecture of the artificial neural networks (ANNs) using the Xilinx system generator (XSG) and its hardware co-simulation equivalent model using field programmable gate array (FPGA) to predict the behavior of Chua's chaotic system and use it in hiding information. The work proposed consists of two main sections. In the first section, MATLAB R2016a was used to build a 3×4×3 feed forward neural network (FFNN). The training results demonstrate that FFNN training in the Bayesian regulation algorithm is sufficiently accurate to directly implement. The second section demonstrates the hardware implementation of the network with the XSG on the Xilinx artix7 xc7a100t-1csg324 chip. Finally, the message was first encrypted using a dynamic Chua system and then decrypted using ANN's chaotic dynamics. ANN models were developed to implement hardware in the FPGA system using the IEEE 754 Single precision floating-point format. The ANN design method illustrated can be extended to other chaotic systems in general.

Corresponding Author:

Wisal Adnan Al-Musawi
Department of Computer Engineering, College of Engineering, University of Basrah
Basrah, Iraq
Email: wisal.eng@gmail.com

1. INTRODUCTION

Real system modeling plays an important role in the study and helps to better understand the behavior. It is a leader in the use of systems in many disciplines and contributes to technological implementation. Most problems are based on modeling techniques such as predictions, classifying, pattern recognition, hybrid prevision, optimizing, and control [1]. Since chaotic systems are extremely complex, their efficiency in chaotic system modeling is not very successful and therefore remains a challenge. Chaotic systems are complex systems whose behavior is highly dependent on their initial conditions. The initial conditions need to be understood in all of their details to predict the long-term conduct of such systems [2].

One of the most efficient and general modeling methods is artificial neural networks (ANN); developed and inspired by actual biological neural brain structures as an artificial intelligence approach, non-parametric and non-linear, which can model complex systems [3]. The ability of ANNs to predict both linear and nonlinear maps is extremely effective for modeling systems, especially when the behavior is chaotic [4]. Many researchers modeling chaotic systems using ANN [5], [6], but in this research, the proposed design was implemented in field programmable gate array (FPGA) and used in the masking of a sine wave and can be used for image encryption in the future.

Numerous studies and research based on a prediction of chaotic systems using neural networks have been established such as: Alçın *et al.* [4] implemented the Pehlivan-Uyaroglu chaotic system (PUCS) in very-high-density lipoprotein (VHDL) IEEE-754 32 bit floating-point standard by using artificial neural networks (ANNs). Koyuncu *et al.* [7], in this research, a novel four-dimensional hyper-chaotic system was implemented as a multi-layer feedforward artificial neural network (FFANN) on an FPGA chip with a 32-bit IEEE-754-1985 floating-point number standard for use in real-time chaos-based applications. Zhang and Lei [8] examines the training efficiency of multilayer ANN architectures for chaotic systems implementation.

ANNs have the advantages of generalizing results obtained from known conditions into unknown situations, fast response times in the operational process, high degrees of systemic parallelism, reliability, and effectiveness. ANNs based on hardware have been a big part of many areas in recent years [9]. For example, random number generation [10], prediction [8], design oscillation, synchronization [11], image processing, medical technology, chemical industry [12], and secure communication [13]. Software-based ANNs are simple to implement, but large neural networks are run very slowly and cannot be adapted to many real-time applications [14]. An alternate option is to use software-based ANNs, such as the FPGA [15].

For many reasons, ANN implementations on FPGAs proved useful. FPGAs can offer higher speeds because they can be completely exploited by the parallel design of ANNs. Moreover, FPGAs are easily reconfigurable and FPGAs have a relatively small design time compared to application-specific integrated circuit (ASIC) design [16]. In comparison to implementations based on software, the well-engineered hardware implementation of the ANN makes substantial improvements in processing efficiency. The potential of ANNs to work in real-time applications gives them an advantage in many such applications over software implementations [17].

This paper’s motivation is to implement an ANN model by Xilinx system generator (XSG) capable of generating a chaotic system directly after training it instead of using the solution of differential equations and using it in secure communications to encryption information. The paper is organized as follows: Section 2, which provides a short introduction to the chaotic Chua system. Section 3 introduces ANN modeling with FPGA. Section 4 illustrates ANN architecture and training. Section 5 shows the XSG design of the ANN based Chua chaotic system. In section 6, Cryptography using ANN is investigated and a sine signal takes as a case study. FPGA hardware Co-simulation testing was conducted in section 7. Finally, some conclusions are stated in section 8.

2. CHUA CHAOTIC SYSTEM

Chua chaotic system has an easy structure and creates chaotic dynamics with sufficient parameters, showing chaos and several known phenomena of bifurcation. Therefore, several researchers have been interested in this method. In 1983, Leon Chua developed a nonlinear circuit that is capable of demonstrating a rich collection of dynamical phenomena, ranging from fixed points to cycle points, standard bifurcations (period-doubling), other standard routes to chaos, and chaos itself. The relevance of the Chua circuit has recently made possible the birth of a big family of multi-scroll oscillators and techniques to control the chaos [18]. Chua’s circuit has found many applications in physics, communication, and control, mechanics, as well as, chemistry, economics, and medicine [19]. Chua’s circuit has also been used as a chaotic noise generator. Because of this property, it has found many applications in cryptography and steganography [20]. The following nonlinear equations describe this chaotic system:

$$x = \alpha(y - x - f(x)$$
$$y = x - y + z$$
$$z = -\beta y \tag{1}$$

where f(x) described as follow:

$$f(x) = m_1 x + 1/2(m_0 - m_1)(|x + b_1| - |x - b_1|) \tag{2}$$

The f(x) function describes electrical nonlinear resistor response. α and β parameters are calculated using the unique values for the circuit components. In this equation, m_0, m_1 is a slope that must have negative values and bi-break point values. The f(x) function controls the number of created scrolls. Figure 1 shows the XSG configuration of the Chua equation system (1) and f(x) in (2). Figure 2 show the double scroll describing the relationship between x and y and the 3d phase portrait respectively. Figure 3 shows a chaotic time series x, y, and z. The system parameters are set as follows: α=10, β=14.87, m0=-1.27, m1=-0.68 and b1=1. The initial conditions are randomly selected as follows: x0=-1, y0=0 and z0=0.6. and a 0.01(dt) clock step size. The system has long-term chaotic behaviors and uneventfulness with these parameters and is sensitive to the initial parameters.

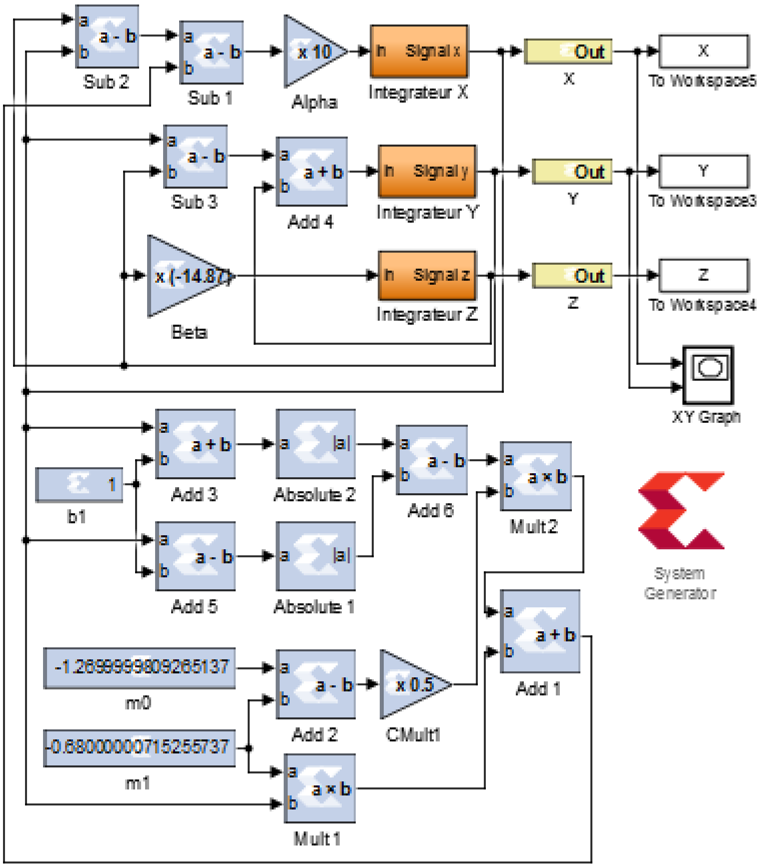


Figure 1. XSG design for 2-scroll Chua's circuit

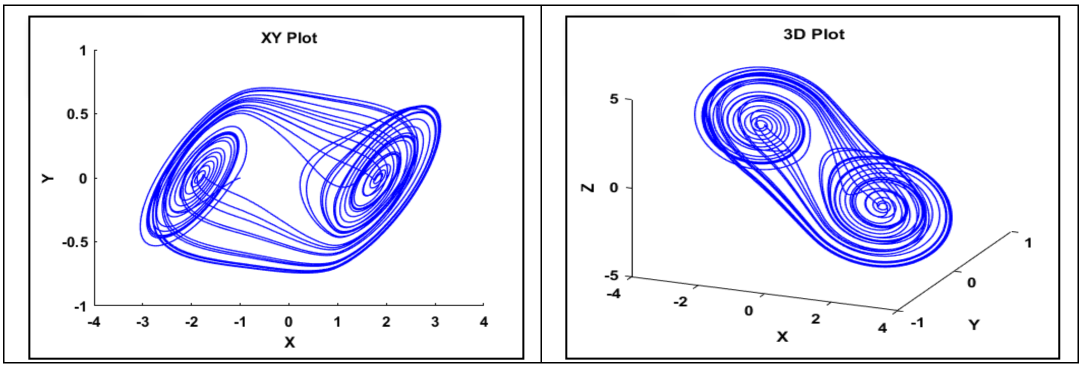


Figure 2. XY and XYZ phase portrait plot

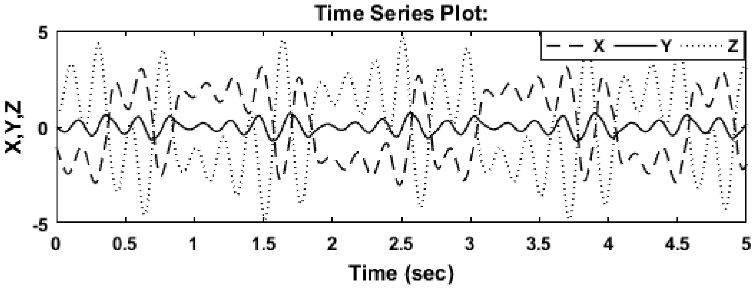


Figure 3. The X, Y, and Z time series

3. ANN MODELING USING FIELD PROGRAMMABLE GATE ARRAYS (FPGA)

ANN modeling that mimics human brain abilities, including learning and generalization, is known to be successful in adapting its behavior in various fields of research. An feed forward neural network (FFNN) consists of many interconnected layer-type neurons. It requires an entrance layer that receives the network input data. The output layer is another layer that exports its outputs externally. There are hidden neurons that form the hidden layer between the two layers. Artificial neurons are the essential information processing units of NN. The block diagram in Figure 4 shows the neuron model that forms the basis for the ANN design. The synaptic weights reflect the effect of the related input signals when comparing the (AN) with the biological one. Figure 4 includes a weighted input adder and an option to increase or decrease the net input of the activation function [4].

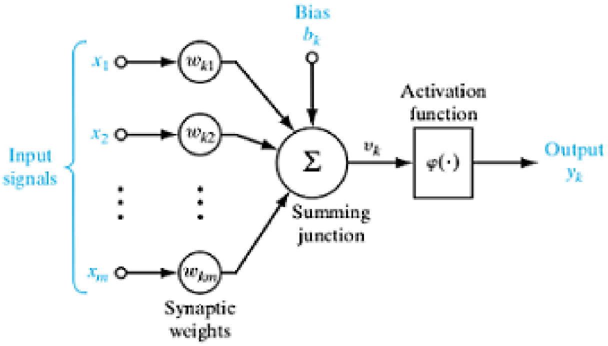


Figure 4. One single neuron

The output y is determined in (3) and f is the neuron activation function.

$$yj = f(\textstyle\sum_i WiXi + b) \tag{3}$$

The input layer neurons only serve as a buffer for transmitting the input signals to hidden layer neurons [9]. The activation function is one of the neuron’s most significant components. The activation function φ(x) limits the neuron's output from the induced local field (x). Sigmoid is a nonlinear continuous function that maps an infinite input domain (-∞, ∞) to a finite output domain (0, 1) for unipolar and (-1, 1) for bipolar [21]. This sigmoid function allows the generalization of any function in ANN models.

$$\varphi(x) = \frac{1}{1+exp\,(-\beta x)} \tag{4}$$

In this equation (β) is the sigmoid function slope parameter. The activation function of the tan sigmoid is popular because it is easily differentiated and important in training algorithms for backpropagation [22]. Since this function requires exponential conditions, it cannot be applied directly [23]. FPGAs provide inexpensive, simple, scalable, and versatile solutions that can also be used for the realization of a single chip digital system [24]. Conventional overall processors are sequential and cannot support the parallel architecture of the neural network. An FPGA implementation not only offers the option of parallelism, but also decreases the design cost, while flexibility, making it especially applicable for applications in ANNs [4]. That is why FPGA has been employed in the design of an ANN-based Chua chaotic system. The designed architecture has been described using XSG with 32-bit floating-point arithmetic since floating-point arithmetic has high precision with a lower number of bits [25]. An optimized ANN architecture with fewer hidden neurons will dramatically reduce the usage of FPGA’s hardware resources and increase running speed. The representation of the data was important because it is important for the efficiency and activation function of the network architecture. To this purpose, before FFNN-based system design using the XSG is initiated, a number format and accuracy for input, weight, and activation functions must be taken into consideration. With the increase in number format accuracy, FPGA resources are also significantly increasing. However, accuracy in the training process has a powerful effect. It is important to define the precise number for the training stage of ANN as high as possible [4]. The IEEE-754 standard for single-precision floating-point is presented in Figure 5. It consists of 1 sign bit, 8 exponent bits, and 23 bits of mantissa.

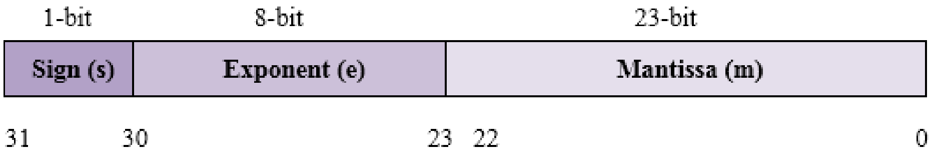


Figure 5. Floating-point number IEEE-754 32-bit representation

4. ANN ARCHITECTURE AND TRAINING-BASED CHUA'S CIRCUIT

In this project, we have a 3-neuron in the input and output layer and 4-neuron in the hidden layer. The number of hidden neurons must be calculated by training results to achieve the minimum neuron number and optimal ANN topology. The mean square error (MSE) between ANN’s outputs and targets determines the training efficiency. A smaller MSE indicates better performance in training. The efficiency of the FPGA hardware implementation is based on ANN topology. Figure 6 shows the 3x4x3 architecture of a two-layer feed-forward network with tangent sigmoid (TanSig) activation function has been used for the hidden layer due to success in modeling nonlinear dynamic systems. The linear (PureLin) function has also been used for the activation function of the output layer. An ANN can be trained for complex tasks like pattern recognition and classification by changing weight. The ANN training technique involves a dataset of different input and output pairs [26]. These pairs of data are sequentially used. After a single set of inputs and errors are calculated, the ANN output values are measured and used to change the weight and bias values for the next input/output pair. The process continues until the outputs of the ANN match the target in a predefined error range. Training data is generated from the Chua system equations, which are implemented directly by a single group of ordinary differential equations based on their definitions (ODEs). Ode5 (Dormand-Prince) is used for solving ODEs, which is a 5th-degree differential equation solver. 10,000 samples were produced divided into three sub-sets: 70% of the samples were used for training, 15% were for validation, and 15% for testing. For assessment, the best MSEs of the validation data set are used. The ANN training is performed using Bayesian regularization training algorithms, which usually take longer, but can generalize well for difficult, limited, or noisy data sets. An ANN feed forward can be trained to produce chaotic systems' expected outputs.

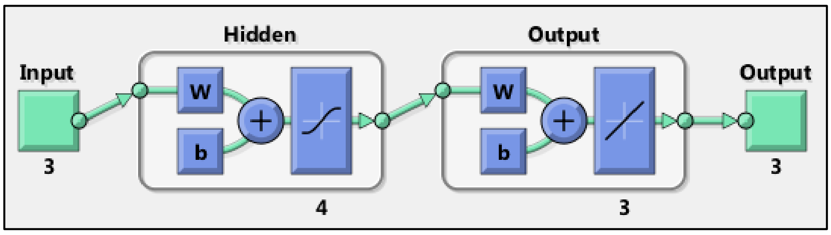


Figure 6. ANN Architecture

5. IMPLEMENTATION OF ANN USING XILINX SYSTEM GENERATOR

This section discusses how the architecture proposed was implemented with the XSG block set in the FPGA. It shows also how the developed XSG model on the actual FPGA from the MATLAB/Simulink (hardware co-simulation) was tested and validated. The ANN-based Chua chaotic system using the Xilinx system generator model is shown in Figure 7, which contains Chua’s circuit implemented as in (1). One hidden layer with four neurons. Each neuron has a hyperbolic tangent sigmoid activation function as shown in Figure 8. One output layer with three neurons each neuron implemented as shown in Figure 9. The speed of the design based on the ANN is lower than the equation-based design of the Simulink. The implementation of the FPGA model on ANN is considerably higher than the model based on equations. This is reasonable, as the architecture of ANN has been designed to simulate a generic chaotic system with unknown representational equations. Figure 10(a) and Figure 10(b) display the phase portrait generated from the Chua circuit that was implemented directly and that from ANN design respectively. The synchronization between them is shown in Figure 11(a). From the results, we can see the difference between the outputs of the Chua circuit created by the equation directly and those generated by the ANN. As an example, the error between the Xc signal of the Chua circuit and the Xn signal of the ANN is (-0.0011 to 0.0009) as shown in Figure 11(b).

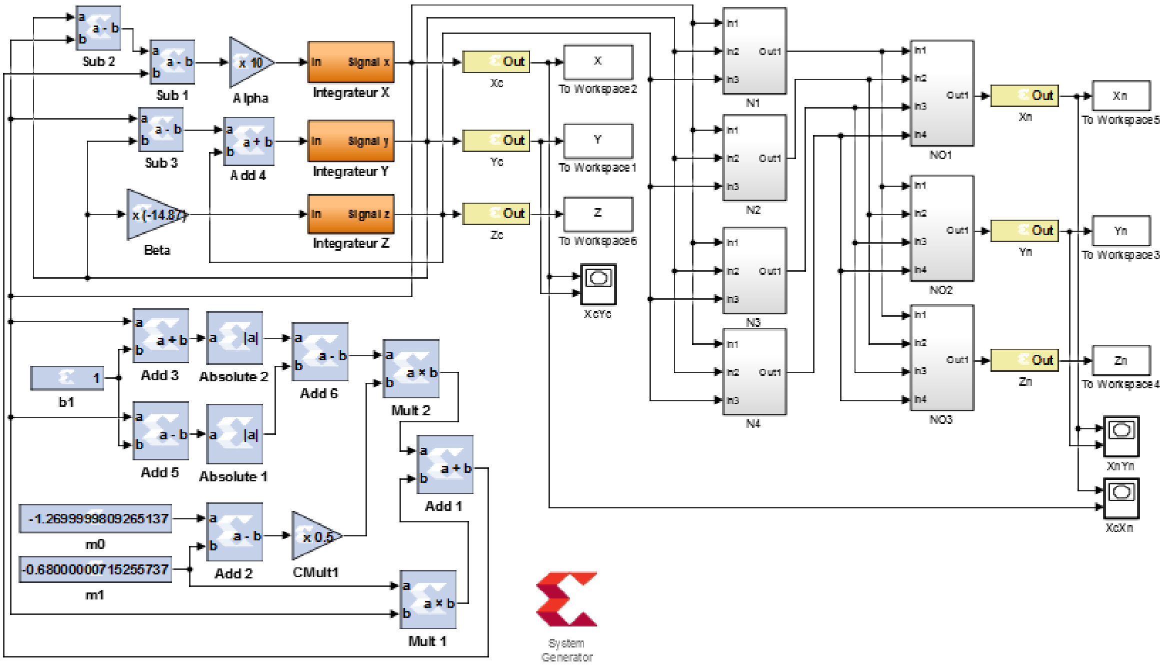


Figure 7. ANN based Chua chaotic system design using XSG

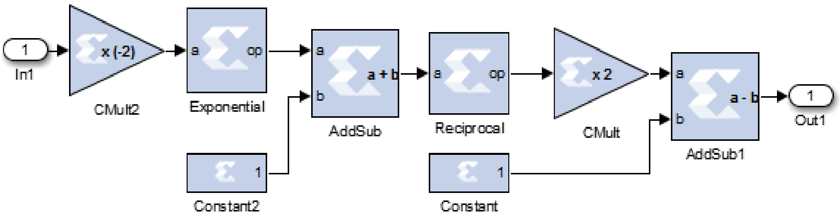


Figure 8. Hidden neurons design with tan-sigmoid activation function

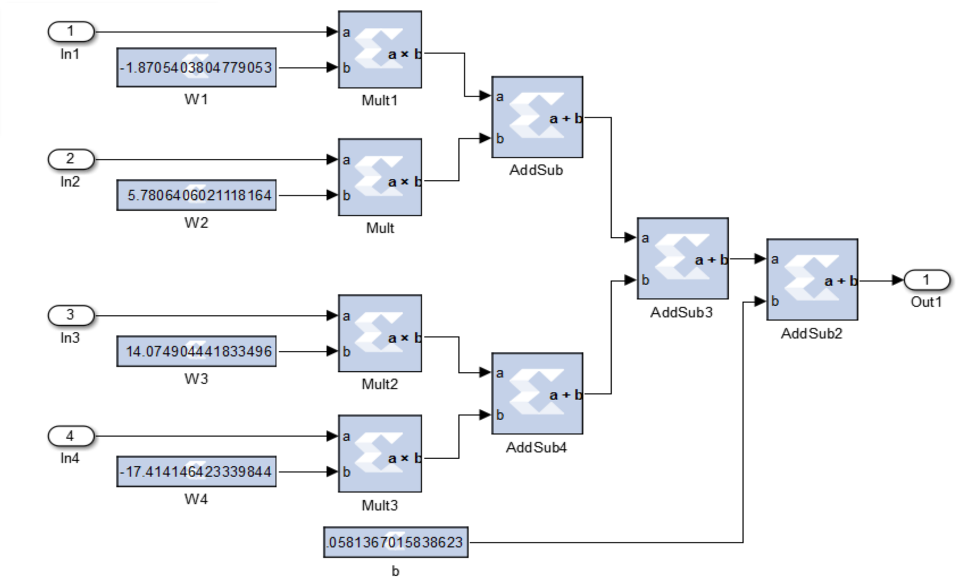
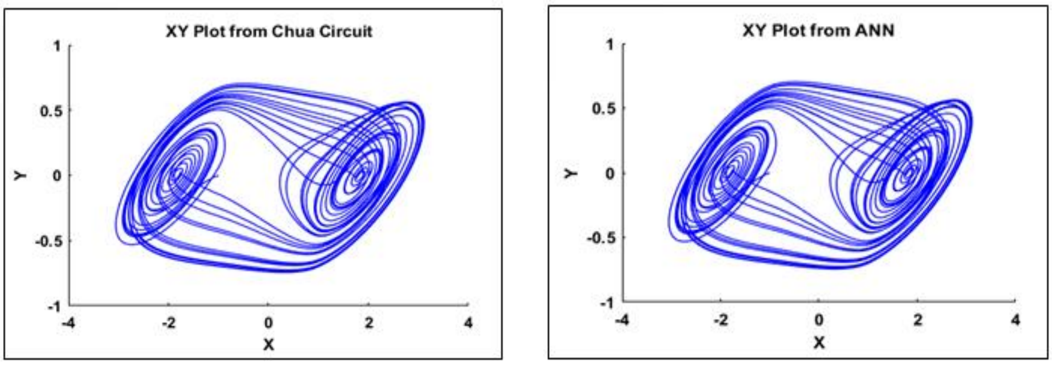


Figure 9. Output neurons design



(a) (b)

Figure 10. Phase portrait from (a) numerical design and (b) from ANN design

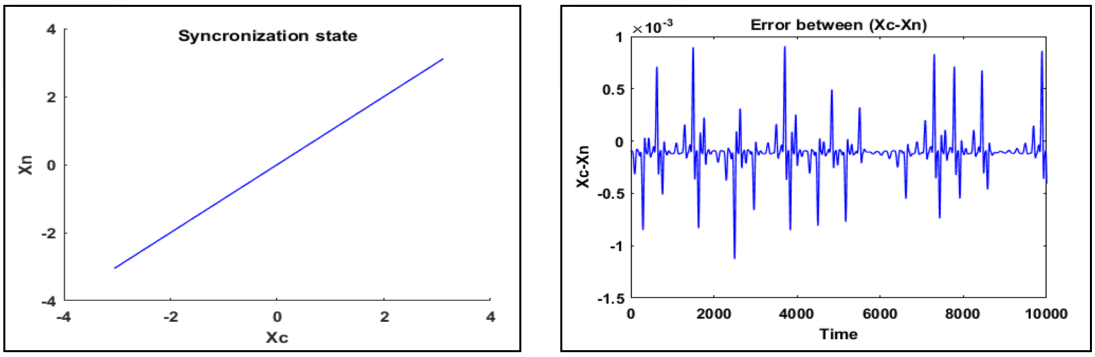


Figure 11. Synchronization state (a) phase portrait describes the X_c versus X_n and (b) error between Xc and Xn signals

6. CRYPTOGRAPHY USING ANN BASED CHUA’S CHAOTIC SYSTEM

Many studies have been conducted in order to develop and use powerful crypto-systems in communications [10]. Some of these researchers are called “classic crypto-systems” are now updated and still being implemented. As an alternative to classical crypto-systems, chaotic crypto-systems can be implemented [27]. Chaotic systems are known as non-periodic systems and respond topologically to initial conditions, system parameters, and transitivity. These are excellent properties for cryptanalysts [28]. Cryptology is a field that covers numerous studies on the overcoming of security and communication mechanisms identifiers. Encryption can be defined as the conversion of the initial message, called plain text, by an algorithm with secret keys into an inscrutable form, named cipher text. The method of decryption is the opposite type of encryption. Under XSG, as shown in Figure 12, secure communication through a chaotic system supported by ANN synchronization was simulated. A sine wave with an amplitude of 2 V is the transmitted signal. The sine wave signal is added to the chaotic signal Xc(t) generated to encrypt signal S(t) as shown in Figure 13. A chaotic signal Xn(t) is extracted from S(t) and Ic(t) is received as shown in Figure 14. The summary of system utilization for the proposed cryptography using ANN-based chaotic generator is shown in Table 1.

Table 1. Device utilization summary for cryptography using ANN

Resource type	Available	Utilization
LUT	63400	21172
Slice Registers (FF)	126800	96
Bonded IOB(IO)	210	193
BUFGCTRL(BUFG)	32	1
DSP	240	162
Minimum period Ts (ns)	130	
Worst negative slack (WNS)	1.087	
Maximum Frequency (MHz)	7.76	
Power(W)	0.153	

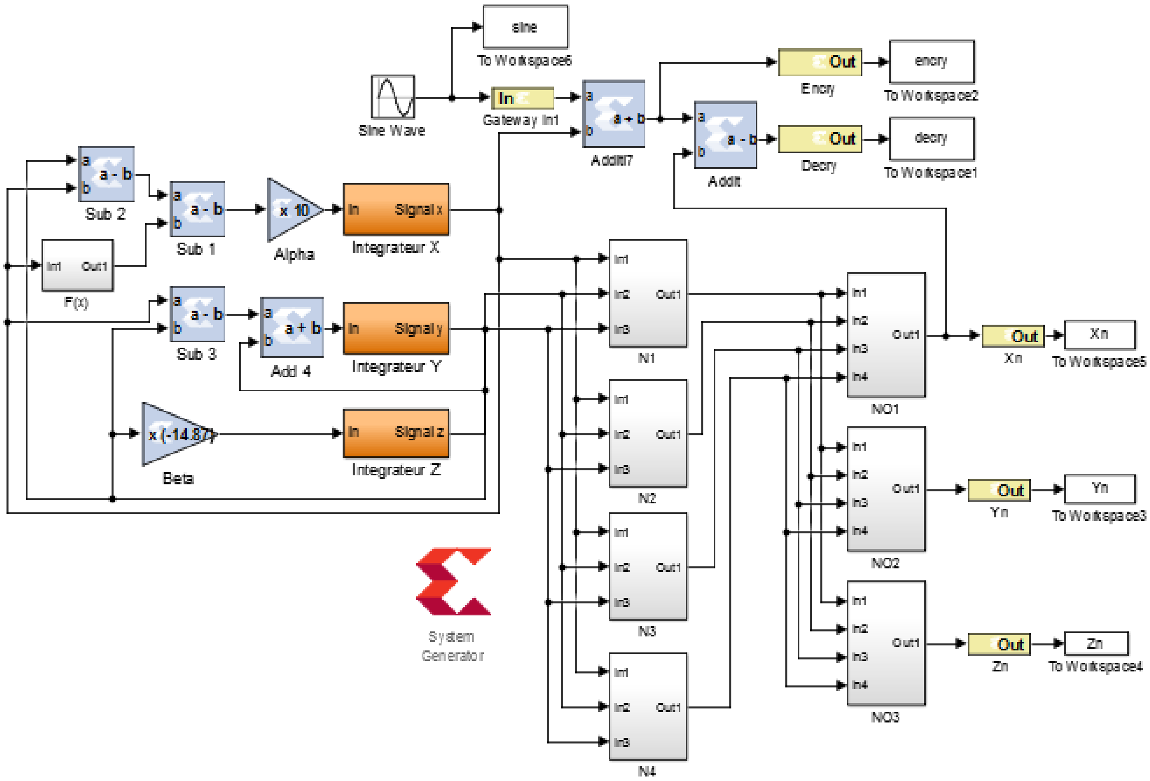


Figure 12. XSG design for cryptography used ANN synchronization

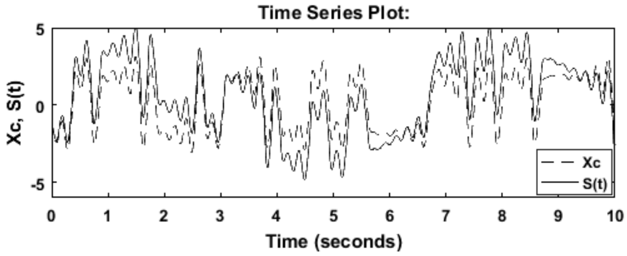


Figure 13. The chaotic signal Xc(t) and encrypted signal S(t)=Xc(t) +I(t)

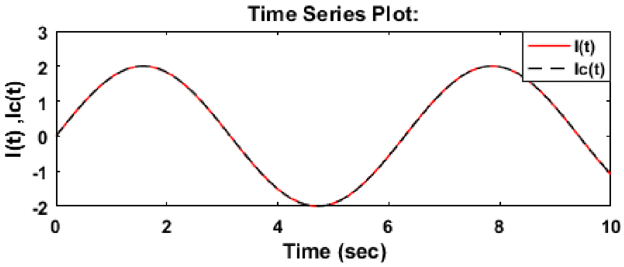


Figure 14. The sine signal I(t) and recovered signal (decrypt) Ic(t)=S(t)-Xn(t)

7. FPGA HARDWARE CO-SIMULATION

The proposed model is formulated using the FPGA board Artix7 xc7a100t-1csg324. The XSG design can be simulated to validate the proposed hardware implementation approach in the environment matrix laboratory MATLAB/Simulink as shown in Figure 15. It is possible to use the hardware co-simulation on an existing FPGA from MATLAB/Simulink. The system generator creates a bit stream file and its associated

Simulink block automatically. This is then uploaded and performed using a joint test action group connection (JTAG) onto the FPGA board. The system generator reads the signals from the MATLAB workspace and sends them to the board via a JTAG connection and the results generated are returned via JTAG to the personal computer (PC) and are displayed by MATLAB.

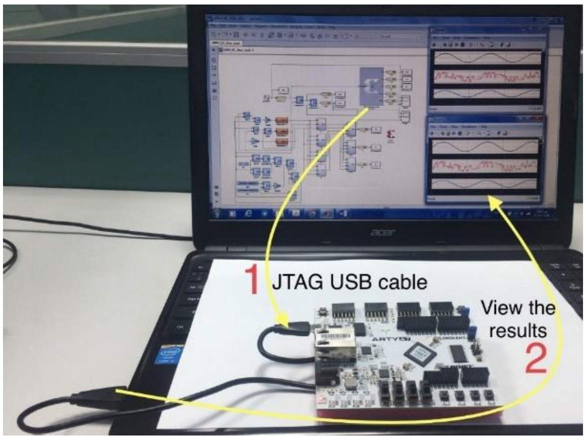


Figure 15. Hardware Co-simulation block diagram

8. CONCLUSION

In this study, the prediction of the Chua chaotic system time series has been implemented on an FPGA using a multi-layer feed-forward structure. The FFNN (3×4×3) architecture was chosen because of its better results with respect to the number of hidden neurons. At the end of the training, the performance function reaches 4.0445E-13MSE. After successfully training the ANN-based Chua chaotic system, the system on FPGA has been designed in XSG with the 32-bit IEEE-754-1985 floating-point number standard by taking the network structure, bias, and weight values as reference. The implementation of the proposed system on FPGA has been tested by synthesizing it with the xilinx vivado program. Resource utilization has been measured and the proposed system has a maximum frequency of approximately 7.76 MHz. In conclusion, the real-time evaluation of the system proposed was co-simulated using the FPGA Xilinx Artix7 xc7a100t-1csg324kit. The proposed system is used in secure communication, and the sine signal was successfully encrypted and decrypted. The proposed method would in the future be used to encrypt images and voices. This study demonstrated that the ANN-based Chua chaotic system built on FPGA can be used successfully in chaotic engineering applications such as synchronization, secure communication, and random number generator.

REFERENCES

[1] M. Molaie, R. Falahian, S. Gharibzadeh, S. Jafari, and J. C. Sprott, “Artificial neural networks: powerful tools for modeling chaotic behavior in the nervous system,” *Frontiers in Computational Neuroscience*, vol. 8, Apr. 2014, doi: 10.3389/fncom.2014.00040.

[2] R. Archana, A. Unnikrishnan, R. Gopikakumari, and M. . Rajesh, “An intelligent computational algorithm based on neural networks for the identification of chaotic systems,” in *2011 IEEE Recent Advances in Intelligent Computational Systems*, Sep. 2011, pp. 605–609, doi: 10.1109/RAICS.2011.6069382.

[3] D. Brezak, T. Bacek, D. Majetic, J. Kasac, and B. Novakovic, “A comparison of feed-forward and recurrent neural networks in time series forecasting,” in *2012 IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, Mar. 2012, pp. 1–6, doi: 10.1109/CIFEr.2012.6327793.

[4] M. Alçın, İ. Pehlivan, and İ. Koyuncu, “Hardware design and implementation of a novel ANN-based chaotic generator in FPGA,” *Optik*, vol. 127, no. 13, pp. 5500–5505, Jul. 2016, doi: 10.1016/j.ijleo.2016.03.042.

[5] H. Lin, C. Wang, W. Yao, and Y. Tan, “Chaotic dynamics in a neural network with different types of external stimuli,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 90, Nov. 2020, Art. no. 105390, doi: 10.1016/j.cnsns.2020.105390.

[6] H. Lin, C. Wang, Q. Hong, and Y. Sun, “A multi-stable memristor and its application in a neural network,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3472–3476, Dec. 2020, doi: 10.1109/TCSII.2020.3000492.

[7] M. Alcin, P. Erdogmus, and I. Koyuncu, “Artificial neural network-based 4-d hyper-chaotic system on field programmable gate array,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 8, no. 2, pp. 102–108, Jun. 2020, doi: 10.18201/ijisae.2020261591.

[8] L. Zhang, “Multilayer artificial neural network design and architecture optimization for the pattern recognition and prediction of EEG signals based on hénon map chaotic system,” in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2017, pp. 345–350, doi: 10.1109/CSCI.2017.58.

[9] I. Dalkiran and K. Danişman, "Artificial neural network based chaotic generator for cryptology," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 18, no. 2, pp. 225–240, 2010, doi: 10.3906/elk-0907-140.

[10] M. Alcin, I. Koyuncu, M. Tuna, M. Varan, and I. Pehlivan, "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*, vol. 47, no. 3, pp. 365–378, 2019, doi: 10.1002/cta.2581.

[11] H. K. Lam and L. D. Seneviratne, "Synchronization of chaotic systems using neural-network-based controller," *International Journal of Bifurcation and Chaos*, vol. 17, no. 6, pp. 2117–2125, 2007, doi: 10.1142/S0218127407018269.

[12] C. Yilmaz, I. Koyuncu, M. Alcin, and M. Tuna, "Artificial neural networks based thermodynamic and economic analysis of a hydrogen production system assisted by geothermal energy on field programmable gate array," *International Journal of Hydrogen Energy*, vol. 44, no. 33, pp. 17443–17459, Jul. 2019, doi: 10.1016/j.ijhydene.2019.05.049.

[13] M. Şahin, Y. Oğuz, and F. Büyüktümtürk, "ANN-based estimation of time-dependent energy loss in lighting systems," *Energy and Buildings*, vol. 116, pp. 455–467, Mar. 2016, doi: 10.1016/j.enbuild.2016.01.027.

[14] R. Abdu-Aljabar, "Design and implementation of neural network in FPGA," *Journal of Engineering and Development*, vol. 16, no. 3, pp. 73–90, 2012.

[15] S. Vaidyanathan *et al.*, "A novel ANN-based four-dimensional two-disk hyperchaotic dynamical system, bifurcation analysis, circuit realisation and FPGA-based TRNG implementation," *International Journal of Computer Applications in Technology*, vol. 62, no. 1, pp. 20–35, 2020, doi: 10.1504/IJCAT.2020.103921.

[16] P. Dondon, J. Carvalho, R. Gardere, P. Lahalle, G. Tsenov, and V. Mladenov, "Implementation of a feed-forward artificial neural network in VHDL on FPGA," in *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*, 2014, pp. 37–40.

[17] J. Schmitz and L. Zhang, "FPGA hardware implementation and optimization for neural network based chaotic system design," in *Proceedings of the 9th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies*, Jun. 2018, pp. 1–6, doi: 10.1145/3241793.3241812.

[18] A. Byagowi and W. Kinsner, "Implementation of a Chua circuit to demonstrate bifurcations and strange attractors in a class," *Proceedings of the Canadian Engineering Education Association (CEEA)*, 2012, doi: 10.24908/pceea.v0i0.4706.

[19] A. S. Andreatos and A. P. Leros, "Secure image encryption based on a Chua chaotic noise generator," *Journal of Engineering Science and Technology Review*, vol. 6, no. 4, pp. 90–103, 2013, doi: 10.25103/jestr.064.11.

[20] L. Merah, A. Ali-Pacha, N. H. Said, and M. Mamat, "A pseudo random number generator based on the chaotic system of chua's circuit, and its real time FPGA implementation," *Applied Mathematical Sciences*, vol. 7, no. 53–56, pp. 2719–2734, 2013, doi: 10.12988/ams.2013.13242.

[21] L. Zhang, "Artificial neural network model-based design and fixed-point FPGA implementation of hénon map chaotic system for brain research," in *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Aug. 2017, pp. 1–4, doi: 10.1109/INTERCON.2017.8079643.

[22] M. Panicker, "Efficient FPGA implementation of sigmoid and bipolar sigmoid activation functions for multilayer perceptrons," *IOSR Journal of Engineering*, vol. 02, no. 06, pp. 1352–1356, 2012, doi: 10.9790/3021-026113521356.

[23] M. A. Cavuslu, C. Karakuzu, and F. Karakaya, "Neural identification of dynamic systems on FPGA with improved PSO learning," *Applied Soft Computing Journal*, vol. 12, no. 9, pp. 2707–2718, 2012, doi: 10.1016/j.asoc.2012.03.022.

[24] E. Z. Mohammed and H. K. Ali, "Hardware implementation of artificial neural network using field programmable gate array," *International Journal of Computer Theory and Engineering*, pp. 780–783, 2013, doi: 10.7763/IJCTE.2013.V5.795.

[25] I. Sahin and I. Koyuncu, "Design and implementation of neural networks neurons with radbas, logsig, and tansig activation functions on FPGA," *Elektronika ir Elektrotechnika*, vol. 120, no. 4, pp. 51–54, 2012, doi: 10.5755/j01.eee.120.4.1452.

[26] L. Zhang, "Artificial neural networks model design of Lorenz chaotic system for EEG pattern recognition and prediction," in *2017 IEEE Life Sciences Conference (LSC)*, Dec. 2017, pp. 39–42, doi: 10.1109/LSC.2017.8268138.

[27] İ. Koyuncu, M. Tuna, İ. Pehlivan, C. B. Fidan, and M. Alçın, "Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator," *Analog Integrated Circuits and Signal Processing*, vol. 102, no. 2, pp. 445–456, 2020, doi: 10.1007/s10470-019-01568-x.

[28] M. Tuna, "A novel secure chaos-based pseudo random number generator based on ANN-based chaotic and ring oscillator: design and its FPGA implementation," *Analog Integrated Circuits and Signal Processing*, vol. 105, no. 2, pp. 167–181, Nov. 2020, doi: 10.1007/s10470-020-01703-z.