International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Artificial Neural Network-Based 4-D Hyper-Chaotic System on Field Programmable

Ismail Koyuncu*[1], Murat Alcin[2], Pakize Erdogmus[3] , Murat Tuna[4]

Submitted: 23/02/2020 Accepted: 08/06/2020

Abstract: In this presented study, a 4-D hyper-chaotic system newly proposed to the literature, has been implemented as Multi-Layer Feed-Forward Artificial Neural Network-based on FPGA chip with 32-bit IEEE-754-1985 floating-point number standard to be utilized in real time chaos-based applications. In the first step of the study, 4-D hyper-chaotic system has been numerically modeled on FPGA using Dormand-Prince numeric algorithm. In the second step, the data set (4X10,000) obtained from Matlab-based numeric model has been divided into two parts as training data set (4X8,000) and test data set (4X2,000) to create ANN-based 4-D hyper-chaotic system. A Multi-Layer Feed-Forward ANN structure with 4 inputs and 4 outputs has been constructed for ANN-based 4-D hyper-chaotic system. This structure has only one hidden layer and there are 8 neurons having Tangent Sigmoid activation function used as the activation function in each neuron. 2.58E-07 Mean Square Error (MSE) value has been obtained from the training of ANN-based 4-D hyper-chaotic system. In the third step, after the successful training of ANN-based 4-D hyper-chaotic system, the design of ANN-based 4-D hyper-chaotic system has been carried out on FPGA by taking the bias and weight values of the ANN structure as reference. In this step, at first, Matlab-based Feed-Forward Multi-Layer 4X8X4 network structure has been coded in Very High Speed Integrated Circuit Hardware Description Language (VHDL) to be implemented on FPGA chips. Then, the bias and weight values of the ANN structure has been converted from decimal number system to floating-point number standard and these converted values have been embedded into the network structure. In the last step, the ANN-based 4-D hyper-chaotic system designed on FPGA has been synthesized and tested using Xilinx ISE Design Suite. The chip statistics have been given after the Place&Route process carried out for the Virtex XC6VHX255T-3FF1155 FPGA chip. The maximum operating frequency of ANN-based 4-D hyper-chaotic system on FPGA has been obtained as 240.861 MHZ.

Keywords: Chaos, 4-D Chaotic system, ANN, Field Programmable Gate Array, VHDL.

1. Introduction

Nowadays, the rapid improvements in technology make our life easier and more comfortable. Whereas, this change can bring possible treatments to us, especially in private data that must be transferred and stored securely. To perform this, the importance devoted to cryptology need to be enhanced. Chaos and chaotic systems have been widely used today in many applications, including cryptology [1]–[4]. It is possible to state that when the chaotic systems and their applications in cryptology increase, cryptology will get even stronger.

Chaotic systems can be grouped as continuous-time and discrete-time [5], [6]. The real time chaotic engineering applications can be performed by modelling the continuous-time chaotic systems using numeric algorithms and different methods [7], [8]. Biomedical, cryptology, signal and image processing, artificial neural networks, random number generators, industrial control and power electronics can be given as the examples of these engineering applications [9]–[15]. The chaotic systems have unique properties including showing noise-like behaviours, having sensitive dependency on initial conditions and demonstrating non-periodic feature [16]–[18]. In addition to this, chaotic systems are also deterministic systems which have in harmony in themselves. To benefit from these properties in chaotic system-based engineering applications mentioned above, a chaotic system producing chaotic signal is needed.

Field Programmable Gate Array (FPGA) is an integrated circuit (I.C.) produced to be configured by the designer. Because of having properties including re-programmability, parallel processing, reliability and processing speed, FPGA-based studies have become an easy and attractive choice for hardware implementation [19]–[21]. In state of the art cryptographic applications, chaotic system-based cryptography is one of the most used security techniques [22], [23]. For this reason, there are several implementations proposed to the literature. Alçın et al (2016) performed the implementation of Pehlivan–Uyaroglu Chaotic System (PUCS) using Artificial Neural Networks (ANNs) in VHDL IEEE-754 32 bit floating-point standard on Virtex-6 FPGA chip [24]. Rajagopal et al. (2017) investigated the implementation of two new fractional-order 4D chaotic system on Xilinx Kintex 7 chip with Matlab-Xilinx System Generator Toolbox [25]. Koyuncu et al. (2013) proposed the implementation of Burke-Shaw Chaotic System (BSCS) utilizing RK5-Butcher (RK5B) numerical algorithm using IEEE-754 floating-point standard on Virtex-6 FPGA chip [26]. Akgül et al. (2016) carried

[1] *Electrical Electronics Eng. Afyon Kocatepe Uni. Afyon–03200, TURKEY ORCID ID : 0000-0003-4725-4879*
[2] *Electrical Electronics Eng., Afyon Kocatepe Uni., Afyon–8023, TURKEY ORCID ID : 0000-0002-2874-7048*
[3] *Computer Engineering, Düzce University, Düzce – 81620, TURKEY ORCID ID : 0000-0003-2172-5767*
[4] *Tech. Sciences Voc. School, Kırklareli Uni., Kırklareli – 39100, TURKEY ORCID ID : 0000-0003-3511-1336*
* *Corresponding Author Email: ismailkoyuncu@aku.edu.tr*

out the implementation of a 3D chaotic system with RK4 numerical algorithm on Xilinx FPGA chip [27]. Rajagopal et al. (2017) developed the implementation of Chameleon system on Virtex 6 FPGA chip with RK5B numerical algorithm using VHDL utilizing 32-bit IEEE-754-1985 floating-point number format [28]. Tuna et al. (2016) proposed the implementation of a novel chaotic system with Heun algorithm on FPGA with VHDL using 32 bit IEEE-754-1985 floating-point number format [29].

One of the most basic structures used in chaos-based studies and chaotic engineering applications is a chaotic oscillator that produces chaotic signal. In this study, the 4-D hyper chaotic system that newly presented in the literature has been successfully implemented to run on FPGA chips in ANN-based 4-D hyper-chaotic oscillator design in VHDL in accordance with the IEEE-754-1985 floating point number standard. This study has shown that ANN-based 4-D Hyper chaotic oscillator designed to work on FPGA can be successfully used in chaotic engineering applications such as synchronization, secure communication and random number generator.

The rest of the paper has been organized as follows: In Section 2, general information about Dormand-Prince numerical algorithm has been presented. Artificial Neural Network-Based 4-D Hyper-Chaotic System has been devoted to Section 3. In Section 4, the implementation of Artificial Neural Network-Based 4-D Hyper-Chaotic System on FPGA has been given. Finally, some concluding remarks have been presented in the last section.

2. Dormand-Prince Algorithm

In literature, many numeric algorithms have been developed for the discrete-time solutions of continuous time differential equations [29]–[33]. Euler, Heun, 4th order of Runge-Kutta, Runge-Kutta-Butcher and Dormand-Prince can be given as examples for these numeric algorithms. Euler algorithm is one of the basic algorithms used for the solutions of differential equations. This algorithm is 1st order algorithm utilized generally for the simplicity of solution and executing the processes quickly. Heun algorithm is more advanced algorithm and it generates more converging results for the solutions of differential equations than Euler algorithm does. 4th order Runge-Kutta algorithm has lower error rate in the solutions of differential equations and therefore it is one of the most chosen algorithms in literature. As can be understood from the name, Runge-Kutta-Butcher algorithm is the 5th order algorithm producing more sensitive results than 4th order Runge-Kutta algorithm does. Whereas, since Runge-Kutta-Butcher algorithm has more coefficients, the numbers of processes increase and time interval of the solution will increase. Dormand-Prince algorithm is a 7th degree differential equation solver that generates more sensitive results than the numeric algorithms mentioned above. Since Dormand-Prince algorithm is a 7th degree differential equation solver, more processes are needed for obtaining the result of the operation and thus, the result of the procedure has been obtained later than the alternatives.

Chaotic systems are the systems that have sensitive dependence on system parameters and initial conditions. In these systems, minor changes in initial conditions and system parameters change considerably the dynamic behaviour of the system. In this respect, the generation of chaotic signals with preserving the characteristic of chaotic systems is significantly important for chaos-based engineering applications. For this reason, Dormand-Prince algorithm is the most suitable one that can model numerically the characteristic of chaotic system with minimum error. Dormand-Prince algorithm is given in Eq. (1).

$$
\begin{aligned}
&y_{i+1} = y_i + h(\frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6)\\
&k_1 = F(x_i, y_i)\\
&k_2 = F(x_i + \frac{h}{5}, y_i + \frac{h}{5}k_1)\\
&k_3 = F(x_i + \frac{3}{10}h, \left(y_i + \frac{3}{40}k_1 + \frac{9}{40}k_2\right) * h)\\
&k_4 = F(x_i + \frac{4}{5}h, \left(y_i + \frac{44}{45}k_1 - \frac{56}{15}k_2 + \frac{32}{9}k_3\right) * h)\\
&k_5 = F(x_i + \frac{8}{9} * h, \left(y_i + \frac{19372}{6561}k_1 - \frac{25360}{2187}k_2 + \frac{64448}{6561}k_3 - \frac{212}{729}k_4\right) * h)\\
&k_6 = F(x_i + 1 * h, \left(y_i + \frac{9017}{3168}k_1 - \frac{355}{33}k_2 + \frac{46732}{5247}k_3 + \frac{49}{176}k_4 - \frac{5103}{18656}k_5\right) * h)\\
&k_7 = F(x_i + 1 * h, \left(y_i + \frac{35}{384}k_1 + 0 * k_2 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6\right) *
\end{aligned}
\tag{1}
$$

The algorithm consists of 7 phases as k_1, k_2, k_3, k_4, k_5, k_6, k_7 and y. When the DP algorithm is examined structurally, although there is not much difference between RK5-Butcher algorithm and DP algorithm, apart from RK5-Butcher, k_7 parameter has been added into DP algorithm and hence it has been provided to produce more precise results. Even if adding k_7 parameter to DP algorithm provides more precise results, it also makes it difficult to implement the algorithm in software and hardware. Therefore, it causes more chip resources to be consumed in real-time digital-based applications. The step size is determined as 0.01 [34].

3. 4-D Hyperchaotic System, Phase Portraits and Time Series

Continuous-time chaotic systems have been expressed using differential equations. The differential equations of 4-D hyperchaotic system newly presented to the literature are given in Eq. (2) [35].

$$
\begin{cases}
\dot{x} = a_1(y - x)\\
\dot{y} = -xz + a_2 y - 5w + 1\\
\dot{z} = xy - a_5 z\\
\dot{w} = a_7 y
\end{cases}
\tag{2}
$$

There are 4 state variables as *x*, *y*, *z* and *w* related to presented hyperchaotic system. The initial conditions and system parameters, which are very important variables in chaotic systems, significantly change the dynamic behaviour of the system. In this study, the system parameters related to 4-D hyperchaotic system have been determined as *a1=30, a2 =20, a5=3, a7=0*. Besides, the initial conditions of 4-D hyperchaotic system have been determined as *x(0)= 0.1, y(0)= 0.1, z(0)= 0.1* and *w(0)= 0.1*. In this study, 4-D hyperchaotic system, presented its continuous-time differential equation to the literature, has been coded in MATLAB using Dormand-Prince algorithm to obtain the ANN-based model. *x*, *y*, *z* and *w* time series obtained from the Dormand-Prince-based model of the 4-D hyper-chaotic system have been illustrated in Figure 1. Also Figure 2 shows *x-y*, *x-z*, *y-z*, *x-w* and *x-y-z* phase portraits obtained from Dormand-Prince–based model of the 4-D hyper-chaotic system. As can be seen from the phase portraits of

the 4-D hyper-chaotic system, the system exhibits chaotic characteristic.

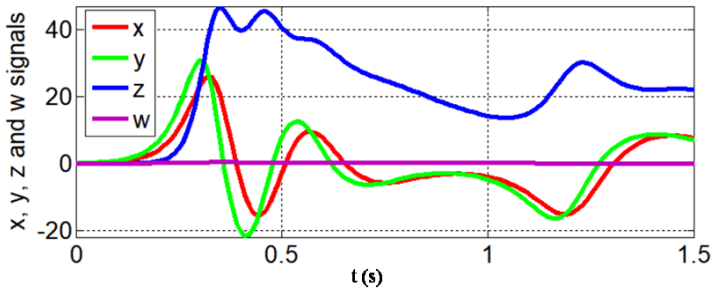

Figure 1.: *x*, *y*, *z* and *w* time series of 4-D hyperchaotic system using Dormand-Prince algorithm.

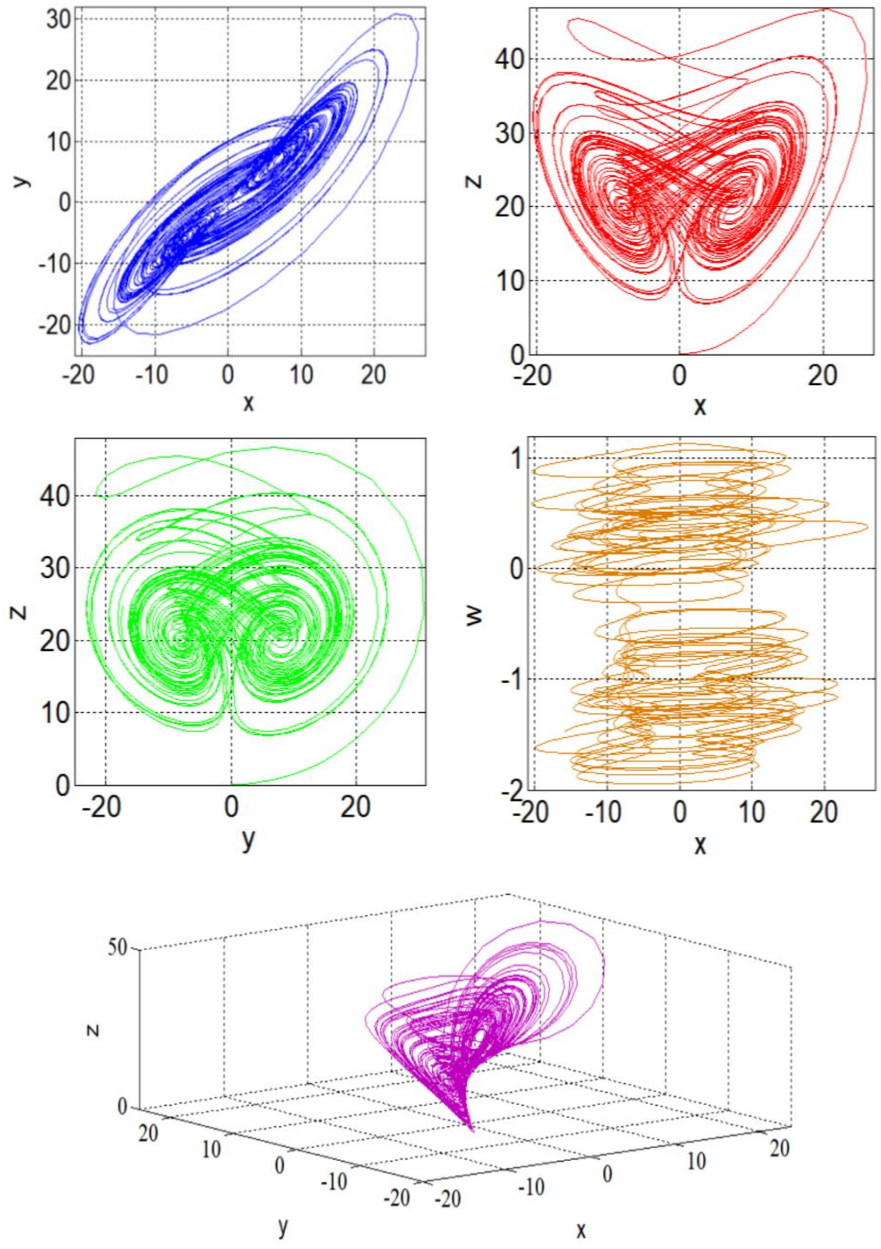

Figure 2. *x-y*, *x-z*, *y-z*, *x-w* and *x-y-z* phase portraits of 4-D hyperchaotic system obtained from Matlab-based Dormand-Prince algorithm

4. Design and Implementation of ANN-Based 4-D Hyperchaotic System on FPGA

In this section, the novel hyperchaotic system in (2) has been modelled using MATLAB environment and then ANN-Based 4-D hyperchaotic system on FPGA has been created using the obtained values of biases and weights from the Feed Forward ANN (FFANN) model. Since online training usually results in loss of efficiency in hardware implementation, offline training has been selected. After the training, the suitable network parameters (weights and biases) have been obtained and then these parameters have been deployed on FPGA [36], [37]. A numeric solution of this system has been accomplished for using FFANN training [38]. The model has 3 layers namely, input, hidden, and an output layer. Since there are 4 state variables in (2), FFANN model has 4 neurons in both input and output layers. Tangent Sigmoid (TanSig) activation function has been used for the hidden layer due to having the success in modelling the nonlinear dynamic systems with hyperchaotic behaviour. Linear (PureLin) function has also been used for the activation function of the output layer. Table 1 illustrates the FFANN model structure of the 4-D hyperchaotic system used for ANN-based modeling.

Table 1 The FFANN model parameters of the 4-D hyperchaotic system used for ANN-based modeling.

Network Structure	4x8x4
Activation function	Hidden Layer: TanSig Output Layer: Purelin
Training function	Trainlm
Performance function	MSE: 2.58E-07
Number of state variables (inputs)	10000
Number of epoches	200.000
Fault tolerance	1E-15

As seen in Table 1, at the end of training the performance function reaches 2.58E-07 MSE (Mean Square Error) value when 8 neurons have been used for the hidden layer. It is possible to decrease the MSE value by increasing the number of neurons in the hidden layer, but it leads to slower speed and larger size which are inconvenient for real time ANN implementations [39]. Besides, there is a trade-off between the number of hidden neurons used in FFANN model structure and the resource used in FPGA. Therefore, the acceptable value of MSE can be used for the implementation of ANN-based 4-D hyperchaotic system.

Since FPGA not only offers parallelism but also flexible designs, savings in cost and design cycle, it is quite suitable for ANN implementations. That's why FPGA has been employed in the design of an ANN-based 4-D hyperchaotic system. The designed architecture has been described using VHDL (Very High Speed Integrated Circuits Hardware Description Language) with 32-bit floating-point arithmetic since floating-point arithmetic has high precision with a lower number of bits [3], [40]. As TanSig has infinite exponential series, the direct implementation of TanSig is not suitable. For the approximation of TanSig activation function, CORDIC and LUT-based exponent calculator have been used [24], [28], [41]. The digital implementation of the 4-D hyperchaotic system has been designed with respect to weight and bias values obtained from FFANN model. The block diagram of the FFANN-based 4-D hyperchaotic system on Virtex XC6VHX255T-3FF1155 FPGA is given in Figure 3.

The implemented design has been simulated and synthesised by using Xilinx Project Navigator. Xilinx ISE 14.7 has been used as a synthesis tool for implementing the design in Xilinx Virtex 6 FPGA. The result of the design of FFANN operation timing diagram has been shown in Figure 4. It takes only 146 clock cycles for the applied 4-8-4 FFANN to calculate the next output. The chip statistics for the FFANN-based 4-D hyperchaotic system on FPGA are given in Table 2. The minimum clock period of the FFANN-based 4-D hyperchaotic system unit on FPGA is 4.152 ns.

Table 2. The chip statistics summary for the FFANN-based 4-D hyperchaotic system on FPGA

FPGA Chip	**Virtex-6**	**%**
Slice Register Number	103705	32
LUTs Number	104701	66
Number of DSP48E1s	8	1
Bonded IOBs Number	259	58
Maximum Clock Frequency (MHz)	240.861	

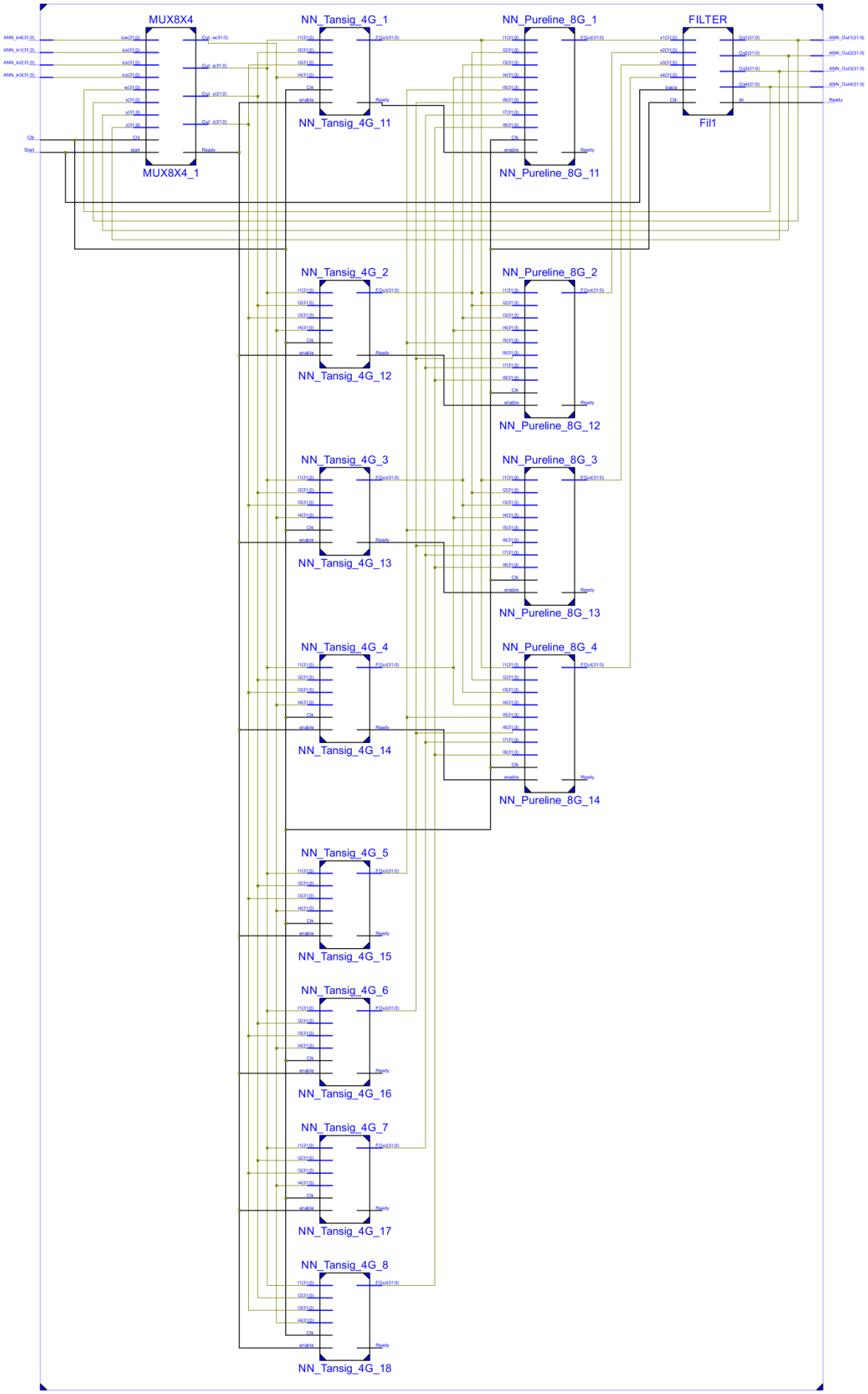

Figure 3. The Top-level block diagram of the FFANN-based 4-D hyperchaotic system on FPGA.

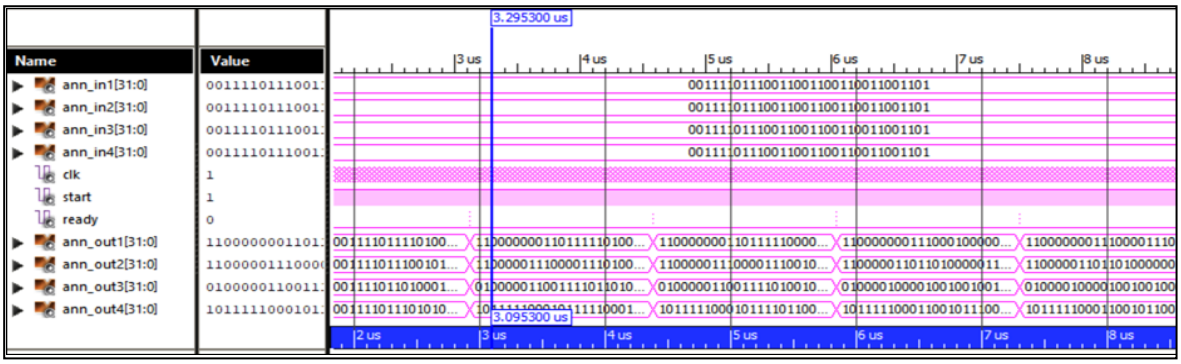

Figure 4. The result of the design of FFANN operation timing diagram.

5. Conclusion

In this study, 4-D hyper-chaotic system newly proposed to the literature, has been implemented on FPGA chip with IEEE-754-1985 floating-point number standard using Multi-layer Feed-Forward ANN-based structure. In this paper, a 4-D hyper-chaotic system has been modelled using Dormand-Prince numeric algorithm on MATLAB. Feed-Forward Multi-layer 4X8X4 network structure has been created using the data set obtained from Matlab-based numeric model. 2.58E-07 MSE value has been obtained from the training phase of ANN-based 4-D hyper-chaotic system. After the successful training of the ANN-based 4-D hyper-chaotic system, ANN-based 4-D hyper-chaotic system on FPGA has been coded in VHDL with 32-bit IEEE-754-1985 floating-point number standard by taking the networks structure, bias and weight values as reference. The implemented ANN-based 4-D hyper-chaotic system on FPGA has been tested by synthesizing it with Xilinx ISE Design Suite. The maximum operating frequency of ANN-based 4-D hyper-chaotic system got after from Place&Route operation for Virtex XC6VHX255T-3FF1155 FPGA chip, is 240.861 MHZ. One of the most basic structures used in chaos-based studies and chaotic engineering applications is a chaotic oscillator that produces chaotic signal. This study has shown that ANN-based 4-D Hyper chaotic oscillator designed to work on FPGA can be successfully used in chaotic engineering applications such as synchronization, secure communication and random number generator. For future works, chaotic synchronization applications and pseudo/true random number generators can be performed using real time ANN-based 4-D hyper-chaotic system on FPGA that presented in this paper.

References

[1] E. Avaroğlu, T. Tuncer, A. B. Özer, B. Ergen, and M. Türk, "A novel chaos-based post-processing for TRNG," *Nonlinear Dyn.*, vol. 81, no. 1–2, pp. 189–199, Jul. 2015, doi: 10.1007/s11071-015-1981-9.

[2] 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 Integr. Circuits Signal Process.*, vol. 102, pp. 445–456, Dec. 2020, doi: 10.1007/s10470-019-01568-x.

[3] I. Dalkiran and K. Danis, "Artificial neural network based chaotic generator for cryptology," *Comp Sci*, vol. 18, no. 2, pp. 225–240, 2010, doi: 10.3906/elk-0907-140.

[4] M. Tuna and C. B. Fidan, "A Study on the importance of chaotic oscillators based on FPGA for true random number generating (TRNG) and chaotic systems," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 33, no. 2, pp. 469–486, 2018, doi: 10.17341/GUMMFD.71479.

[5] M. Alçın, M. Tuna, and İ. Koyuncu, "IQ-Math Based Designing of Fourth Order Runge-Kutta Algorithm on FPGA and Performance Analysis According to ANN Approximation," *Int. J. Adv. Sci. Eng. Technol.*, vol. 5, no. 8, pp. 6523–6530, 2018.

[6] A. M. Garipcan and E. Erdem, "Implementation and Performance Analysis of True Random Number Generator on FPGA Environment by Using Non-periodic Chaotic Signals Obtained from Chaotic Maps," *Arab. J. Sci. Eng.*, vol. 44, no. 11, pp. 9427–9441, 2019, doi: 10.1007/s13369-019-04027-x.

[7] I. Koyuncu, A. T. Ozcerit, and I. Pehlivan, "Implementation of FPGA-based real time novel chaotic oscillator," *Nonlinear Dyn.*, vol. 77, no. 1–2, pp. 49–59, Jul. 2014, doi: 10.1007/s11071-014-1272-x.

[8] A. D. Pano-Azucena, E. Tlelo-Cuautle, G. Rodriguez-Gomez, and L. G. De La Fraga, "FPGA-based implementation of chaotic oscillators by applying the numerical method based on trigonometric polynomials," *AIP Adv.*, vol. 8, no. 7, pp. 075217–12, Jul. 2018, doi: 10.1063/1.5038583.

[9] V. Vembarasan and P. Balasubramaniam, "Chaotic synchronization of Rikitake system based on T-S fuzzy control techniques," *Nonlinear Dyn.*, vol. 74, no. 1–2, pp. 31–44, Oct. 2013, doi: 10.1007/s11071-013-0946-0.

[10] L. Li, Z. Liu, D. Zhang, and H. Zhang, "Controlling chaotic robots with kinematical redundancy.," *Chaos*, vol. 16, no. 1, p. 013132, Mar. 2006, doi: 10.1063/1.2178447.

[11] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan, and A. Zengin, "Secure image encryption algorithm design using a novel chaos based S-Box," *Chaos, Solitons & Fractals*, vol. 95, pp. 92–101, Feb. 2017, doi: 10.1016/J.CHAOS.2016.12.018.

[12] 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," *Int. J. Circuit Theory Appl.*, vol. 47, no. 3, pp. 365–378, Mar. 2019, doi: 10.1002/cta.2581.

[13] E. Avaroğlu, İ. Koyuncu, A. B. Özer, and M. Türk, "Hybrid pseudo-random number generator for cryptographic systems," *Nonlinear Dyn.*, vol. 82, no. 1–2, pp. 239–248, Oct. 2015, doi: 10.1007/s11071-015-2152-8.

[14] U. E. Kocamaz, S. Çiçek, and Y. Uyaroğlu, "Secure Communication with Chaos and Electronic Circuit Design Using Passivity-Based Synchronization," *J. Circuits, Syst. Comput.*, vol. 27, no. 04, p. 1850057, Apr. 2018, doi: 10.1142/S0218126618500573.

[15] İ. Koyuncu, Y. Oğuz, H. Çimen, T. Özer, and M. Tuna, "Design and Implementation of Artificial Neural Network-Based 3-D Novel Jerk Chaotic Oscillator on FPGA," *3rd International Conference on Engineering Technology and Applied Sciences*, Skopje, Macedonia,

pp. 1–6, 2018.

[16] Ü. Çavuşoglu, Y. Uyaroglu, and İ. Pehlivan, "Design of A Continuous-Time Autonomous Chaotic Circuit and Application of Signal Masking," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 29, no. 1, pp. 79–87, Mar. 2014, doi: 10.17341/gummfd.73592.

[17] M. Tuna, M. Alçın, İ. Koyuncu, C. B. Fidan, and İ. Pehlivan, "High speed FPGA-based chaotic oscillator design," *Microprocess. Microsyst.*, vol. 66, pp. 72–80, Apr. 2019, doi: 10.1016/J.MICPRO.2019.02.012.

[18] R. A. Elmanfaloty and E. Abou-Bakr, "Random property enhancement of a 1D chaotic PRNG with finite precision implementation," *Chaos, Solitons and Fractals*, vol. 118, pp. 134–144, Jan. 2019, doi: 10.1016/j.chaos.2018.11.019.

[19] F. Katırcıoğlu, İ. Koyuncu, M. Kelek, Y. Oğuz, and M. Şen, "FPGA-Based Design of Gaussian Membership Function for Real-Time Fuzzy Logic Applications," *V. International Multidisciplinary Congress Of Eurasia*, Barselona, Spain, 2018.

[20] M. Tuna, C. B. Fidan, I. Koyuncu, and I. Pehlivan, "Real time hardware implementation of the 3D chaotic oscillator which having golden-section equilibra," *2016 24th Signal Processing and Communication Application Conference (SIU)*, IEEE, Zonguldak, Turkey, pp. 1309–1312, May-2016.

[21] S. Vaidyanathan *et al.*, "A novel ANN-based four-dimensional two-disk hyperchaotic dynamical system, bifurcation analysis, circuit realisation and FPGA-based TRNG implementation," *Int. J. Comput. Appl. Technol.*, vol. 62, no. 1, pp. 20–35, 2020, doi: 10.1504/IJCAT.2020.103921.

[22] E. Tlelo-Cuautle, J. J. Rangel-Magdaleno, A. D. Pano-Azucena, P. J. Obeso-Rodelo, and J. C. Nunez-Perez, "FPGA realization of multi-scroll chaotic oscillators," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 27, no. 1–3, pp. 66–80, Oct. 2015, doi: 10.1016/j.cnsns.2015.03.003.

[23] M. Stipcevic, "Quantum random number generators and their applications in cryptography," *Proc. Artic.*, vol. 8375, no. 2012, pp. 1–6, May 2012, doi: 10.1117/12.919920.

[24] M. Alçın, İ. Pehlivan, and İ. Koyuncu, "Hardware design and implementation of a novel ANN-based chaotic generator in FPGA," *Opt. - Int. J. Light Electron Opt.*, vol. 127, no. 13, pp. 5500–5505, Jul. 2016, doi: 10.1016/j.ijleo.2016.03.042.

[25] K. Rajagopal, A. Karthikeyan, and A. K. Srinivasan, "FPGA implementation of novel fractional-order chaotic systems with two equilibriums and no equilibrium and its adaptive sliding mode synchronization," *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2281–2304, Mar. 2017, doi: 10.1007/s11071-016-3189-z.

[26] I. Koyuncu, A. T. Ozcerit, and I. Pehlivan, "An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system," *Optoelectron. Adv. Materıals-Rapıd Communıcatıons*, vol. 7, no. 9, pp. 635–638, Sep. 2013.

[27] A. Akgul, H. Calgan, I. Koyuncu, I. Pehlivan, and A. Istanbullu, "Chaos-based engineering applications with a 3D chaotic system without equilibrium points," *Nonlinear Dyn.*, vol. 84, no. 2, pp. 481–495, Nov. 2015, doi: 10.1007/s11071-015-2501-7.

[28] K. Rajagopal, A. Akgul, S. Jafari, A. Karthikeyan, and I. Koyuncu, "Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses," *Chaos, Solitons & Fractals*, vol. 103, pp. 476–487, Oct. 2017, doi: 10.1016/J.CHAOS.2017.07.007.

[29] M. Tuna and C. B. Fidan, "Electronic circuit design, implementation and FPGA-based realization of a new 3D chaotic system with single equilibrium point," *Opt. - Int. J. Light Electron Opt.*, vol. 127, no. 24, pp. 11786–11799, 2016, doi: 10.1016/j.ijleo.2016.09.087.

[30] I. Koyuncu, M. Alcin, M. Tuna, I. Pehlivan, M. Varan, and S. Vaidyanathan, "Real-time high-speed 5-D hyperchaotic Lorenz system on FPGA," *Int. J. Comput. Appl. Technol.*, vol. 61, no. 3, pp. 152–165, 2019, doi: 10.1504/IJCAT.2019.102852.

[31] A. E. Matouk, "Dynamics and control in a novel hyperchaotic system," in *International Journal of Dynamics and Control*, 2019, vol. 7, no. 1, pp. 241–255, doi: 10.1007/s40435-018-0439-6.

[32] M. Tuna, C. B. Fidan, and İ. Koyuncu, *The Chaos-Based Dual Entropy Core TRNG on FPGA, with VHDL codes of Chaotic Systems*. LAP Lambert Academic Publishing, ISBN:978-613-9-9958-3, 2019.

[33] T. Tuncer, "The implementation of chaos-based PUF designs in field programmable gate array," *Nonlinear Dyn.*, vol. 86, no. 2, pp. 975–986, Oct. 2016, doi: 10.1007/s11071-016-2938-3.

[34] S. Chen, B. Li, and C. Zhou, "FPGA implementation of SRAM PUFs based cryptographically secure pseudo-random number generator," *Microprocess. Microsyst.*, vol. 59, pp. 57–68, Jun. 2018, doi: 10.1016/j.micpro.2018.02.001.

[35] P. Prakash *et al.*, "A Novel Simple 4-D Hyperchaotic System with a Saddle-Point Index-2 Equilibrium Point and Multistability: Design and FPGA-Based Applications," *Circuits, Syst. Signal Process.*, pp. 1–22, Feb. 2020, doi: 10.1007/s00034-020-01367-0.

[36] I. Koyuncu, "Design and Implementation of High Speed Artificial Neural Network Based Sprott 94 S System on FPGA," *Int. J. Intell. Syst. Appl. Eng.*, vol. 4, no. 2, p. 33, May 2016, doi: 10.18201/ijisae.97824.

[37] M. Rana, D. Abdu-Aljabar, and A. Lecturer, "Design and Implementation of Neural Network in FPGA," *J. Eng. Dev.*, vol. 16, no. 3, pp. 73–90, 2012.

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

[39] S. Sahin, Y. Becerikli, and S. Yazici, "Neural network implementation in hardware using FPGAs," in *Lecture Notes in Computer Science*, 2006, vol. 4234 LNCS, pp. 1105–1112, doi: 10.1007/11893295_122.

[40] I. Sahin and I. Koyuncu, "Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSig Activation Functions on FPGA," *Electron. Electr. Eng.*, vol. 120, no. 4, pp. 51–54, Apr. 2012, doi: 10.5755/j01.eee.120.4.1452.

[41] M. A. Çavuşlu, C. Karakuzu, S. Şahin, and M. Yakut, "Neural network training based on FPGA with floating point number format and it's performance," *Neural Comput. Appl.*, vol. 20, no. 2, pp. 195–202, 2011, doi: 10.1007/s00521-010-0423-3.