IQ-Math Based Designing of Fourth Order Runge-Kutta Algorithm on FPGA and Performance Analysis According to ANN Approximation

Murat alçın
Afyon Kocatepe University
15 PUBLICATIONS **48** CITATIONS
SEE PROFILE

Murat Tuna
Kirklareli University
29 PUBLICATIONS **77** CITATIONS
SEE PROFILE

Ismail Koyuncu
Afyon Kocatepe University
40 PUBLICATIONS **325** CITATIONS
SEE PROFILE

ISSN: 2350-0328

International Journal of AdvancedResearch in Science, Engineering and Technology

Vol. 5, Issue 8 , August 2018

IQ-Math Based Designing of Fourth Order Runge-Kutta Algorithm on FPGA and Performance Analysis According to ANN Approximation

Murat Alçın, Murat Tuna*, İsmail Koyuncu

Assistant Professor, Department of Mechatronics Engineering, Afyon Kocatepe University, Afyon, Turkey
Assistant Professor, Department of Electrical, Technical Sciences Vocational School, Kırklareli University, Kırklareli, Turkey
Assistant Professor, Department of Electrical and Electronics Engineering, Afyon Kocatepe University, Afyon, Turkey

ABSTRACT: In this paper, the design of the fourth order Runge-Kutta (RK4) algorithm has been performed using 32-bit IQ-Math (16I-16Q) fixed point number format in VHDL on FPGA. The designed system has been implemented into 3-B Jerk chaotic system. The design has been synthesized in Xilinx ISE Design Tools 14.7 programmer and it has been implemented in Xilinx Virtex–6 FPGA chip.The performance analyses have been carried out by evaluating the maximum operating frequencies and chip statistics that obtained from Place&Route process of FPGA-based designs. The comparative analyses between RK4 numeric-based Jerk chaotic system designed in fixed point number format on FPGA and ANN-based Jerk chaotic system on FPGA existing in literature have been performed. Accordingly, it has been observed that the numeric-based Jerk chaotic oscillator has not only greater operating frequency but also lower chip resource ratio. In future, chaos based various engineering applications can be carried out utilizing the Jerk chaotic system model implemented in FPGA-based fixed point number format.

KEYWORDS: RK4 Algorithm, IQ-Math number standard, FPGA, Artificial Neural Networks, VHDL**.**

I.INTRODUCTION

The chaotic systems have sensitive dependence on initial conditions, irregular looking and they arise in deterministic nonlinear time-varying systems. When the definition of chaos is investigated, it has been observed that it has a system that has sensitive dependence on initial conditions exponentially and deterministic behavior, is nonlinear, a periodic dynamical systems in long term [1].On the other hand the short definition is that it is a science branch enabling the explanation of nonlinear phenomena and being described as the organization of irregularity. Additionally, it is a progress having its own organization. Especially, one of the remarkable points is that chaos is not randomness. One of the most complex steady state behavior known in dynamical systems is chaos [2]. Chaos related studies are some part of the theory of nonlinear dynamical systems. Since chaotic systems have interesting dynamical properties despite having in complex circuit structures, the attraction related to different application areas of chaotic systems have been increasing, in recent years [3–5].Artificial Neural Networks (ANN), biomedical, communication, optic electronics and electromagnetic, image processing, fuzzy logic, power electronics, optimization, robot control and mechatronics can be exemplified for these application areas [6–12]. Due to different features of chaotic signals, these systems have been used in electric-electronic engineering field in the areas of cryptography [13] for data security, composing chaos-based secure communication mechanisms [14], chaotic noise generators [15], chaos-based encryption [16] and chaotic random number generators [17,18].

From the viewpoint of increasing and extension of chaos-based engineering applications, it is need to diversify, make more flexible and functional of presented chaotic circuit models. The application of new circuit technics using FPGA (Field Programmable Gate Array) in flexible electronic circuit design makes the applications used in these models more flexible and useful [19,20]. FPGA-based chaotic circuit models are extremely convenient with respect to programmability and re-configurability [21,22]. Because the signal generation in different mode with respect to parameter variations in chaotic systems and the implementation of such systems with different nonlinear functions alternatively are possible so the chaotic circuit models having these features can be designed in programmable and reconfigurable manner without hardware complexity [23]. The studies related to digital FPGA-based modeling of chaotic oscillators have been paid attention in literature [24–26].

In recent years, different studies in many field have been performed using hardware-based ANN [27]. ANN can be modeled in two ways namely, hardware-based and software-based. Two different transfer functions namely, linear and nonlinear have been used. Nonlinear transfer functions include exponential operations, for this reason the hardware implementation of nonlinear transfer functions are quite difficult. Different structures like ASIC (Application Specific Integrated Circuits), DSP (Digital Signal Processor), CNN (Cellular Neural Network) and FPGA chips have been used in the hardware implementation of ANNs [28–31]. Real time ANN-based chaotic oscillator designs have been carried out due to the features of FPGA like reprogrammability, high hardware capacity [32] and paralell processing [11,33–35].

In this paper, firstly 3-B Jerk chaotic system has been coded in 32-bit IQ-Math (16I-16Q) fixed point number format with VHDL using the fourth order Runge-Kutta (RK4) algorithm on FPGA. The designed 3-B Jerk chaotic system has been tested using Xilinx ISE 14.7 design tools and synthesized for Virtex-6 FPGA chip. For this reason, in the second part of the study, the mathematical model of Jerk chaotic system and the phase portraits obtained from this numerical model have been given. In the third part of the study, the RK4-based model of Jerk chaotic system has been presented in IQ-Math fixed point number format on FPGA. It is purposed to design and implement the Jerk chaotic system as hardware in a simpler, faster and more effective way through fixed point number design on FPGA [36]. In the last part, the test results and chip statistics obtained from this study have been compared and evaluated with a different study that presents the same chaotic system designed using ANN structure in 32-bit IEEE 754-1985 floating point number standart on FPGA.

II.JERK CHAOTIC OSCILLATOR AND ITS DISCRETIZED MODEL

A) Mathematical model of Jerk chaotic system

Chaotic systems can be divided into two parts namely, discrete and continuous time. The logistic map can be given as an example of discrete time chaotic systems. Continuous time chaotic systems is described with differential equations [37]. There exist many chaotic systems having different features and being carried out many studies using these chaotic systems, therefore new chaotic systems have been presented to the literature day by day. The differential equations of 3-B Jerk chaotic system have been given in Eq. 1 [38].

$$\begin{aligned} \dot{x}1 &= x2 \\ \dot{x}2 &= x3 \\ \dot{x}3 &= \alpha \cdot x1 - \beta \cdot x2 - x3 - x1^2 - x2^2 \end{aligned} \qquad (1)$$

In the given differential equation α and β are the system parameters and they are defined as α=7.5 and β=4 in Eq. 1. Besides, x1(0)=0.2, x2(0)=0.6 and x3(0)=0.4 are the initial conditions. When a continuous time nonlinear contains one nonlinear term and two variables at least, this dynamic system can demonstrate chaotic features. If this nonlinear system meets these requirements, chaotic analyses can be carried out for this system. However these requirements do not need to be met in discrete time chaotic systems. Various methods including the investigating the system’s phase portraits, analyzing the time series, Poincare mapping, power spectrum, bifurcation diagram and Lyapunov exponents spectrum have been developed for chaotic analyzing a system. 3-B Jerk chaotic system that is presented in this paper has been modeled numerically using RK4 algorithm. Fig 1 illustrates x1-x2, x1-x3, x2-x3 phase portraits of 3-B Jerk system and time series of x1, x2, and x3 state variables. The designed chaotic Jerk oscillator has 7 terms and 2 parameters.

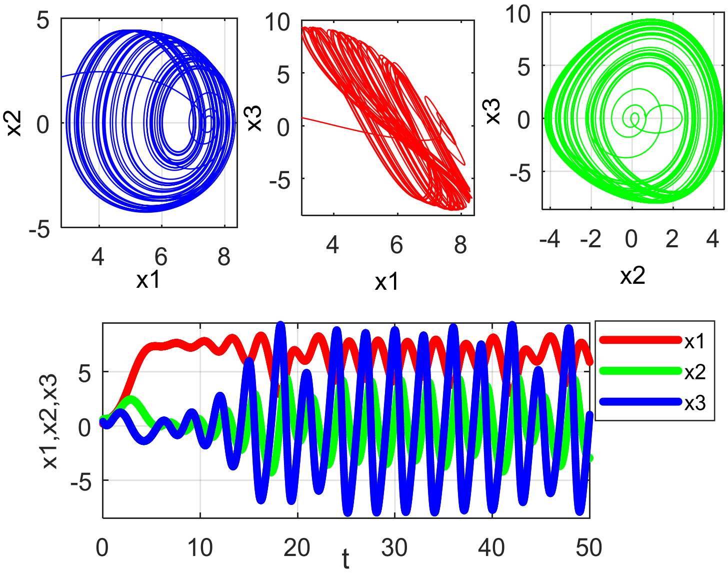

Fig 1.x_1-x_2, x_1-x_3, x_2-$x_3$phase portraits of 3-B Jerk system and time series of x_1-x_2-x_3 state variables

B) The discretized model of Jerk chaotic oscillator

In this part, the discretized model of RK4 numeric algorithm has been obtained to be used for chaotic Jerk oscillator on FPGA and then all processes have been carried out with respect to this step. The initial values *x1(i)*, *x2(i)* and *x3(i)* have been defined as *x1(i)=0.2*, *x2(i)=0.6* and *x3(i)=0.4*. Eq. 3 and Eq. 4 present the discretized mathematical model of chaotic Jerk oscillator using RK4 numeric algorithm with respect to f_1, f_2 and f_3 functions in Eq. 2. In these equations, κ_1, κ_2, κ_3 and κ_4 parameters indicate the coefficients related to first equation of chaotic system; λ_1, λ_2, λ_3 and λ_4 parameters show the coefficients related to second equation of chaotic system; ξ_1, ξ_2, ξ_3 and ξ_4 parameters give the coefficients related to third equation of chaotic system. After the increment of the step size (Δh), the next values of *x1(i)*, *x2(i)* and *x3(i)* (*x1(i+1)*, *x2(i+1)* and *x3(i+1)*) have been calculated by replacing these coefficients in RK4 algorithm. At the end of each iteration, the output values of the system, *x1(i+1)*, *x2(i+1)* and *x3(i+1)* have been utilized as not only outputs but also the initial conditions of the algorithm for the next iteration.

$$\begin{aligned}\dot{x}1 &= f_1(t,x1,x2,x3) = x2\\ \dot{x}2 &= f_2(t,x1,x2,x3) = x3\\ \dot{x}3 &= f_3(t,x1,x2,x3) = \alpha\cdot x1-\beta\cdot x2 - x3 - x1^2 - x2^2\end{aligned} \tag{2}$$

$$\begin{aligned}x1(i+1) &= x1(i)+\frac{1}{6}\Delta h\left[\kappa_1(i)+2\kappa_2(i)+2\kappa_3(i)+\kappa_4(i)\right]\\ x2(i+1) &= x2(i)+\frac{1}{6}\Delta h\left[\lambda_1(i)+2\lambda_2(i)+2\lambda_3(i)+\lambda_4(i)\right]\\ x3(i+1) &= x3(i)+\frac{1}{6}\Delta h\left[\xi_1(i)+2\xi_2(i)+2\xi_3(i)+\xi_4(i)\right]\end{aligned} \tag{3}$$

$$
\begin{aligned}
\kappa_1 &= f_1(x1(i), x2(i), x3(i)) \\
\lambda_1 &= f_2(x1(i), x2(i), x3(i)) \\
\xi_1 &= f_3(x1(i), x2(i), x3(i)) \\
\kappa_2 &= f_1(x1(i) + \frac{1}{2}\Delta h \cdot \kappa_1,\ x2(i) + \frac{1}{2}\Delta h \cdot \lambda_1,\ x3(i) + \frac{1}{2}\Delta h \cdot \xi_1) \\
\lambda_2 &= f_2(x1(i) + \frac{1}{2}\Delta h \cdot \kappa_1, x2(i) + \frac{1}{2}\Delta h \cdot \lambda_1, x3(i) + \frac{1}{2}\Delta h \cdot \xi_1) \\
\xi_2 &= f_3(x1(i) + \frac{1}{2}\Delta h \cdot \kappa_1, x2(i) + \frac{1}{2}\Delta h \cdot \lambda_1, x3(i) + \frac{1}{2}\Delta h \cdot \xi_1) \\
\kappa_3 &= f_1(x1(i) + \frac{1}{2}\Delta h \cdot \kappa_2, x2(i) + \frac{1}{2}\Delta h \cdot \lambda_2, x3(i) + \frac{1}{2}\Delta h \cdot \xi_2) \\
\lambda_3 &= f_2(x1(i) + \frac{1}{2}\Delta h \cdot \kappa_2, x2(i) + \frac{1}{2}\Delta h \cdot \lambda_2, x3(i) + \frac{1}{2}\Delta h \cdot \xi_2) \\
\xi_3 &= f_3(x1(i) + \frac{1}{2}\Delta h \cdot \kappa_2, x2(i) + \frac{1}{2}\Delta h \cdot \lambda_2, x3(i) + \frac{1}{2}\Delta h \cdot \xi_2) \\
\kappa_4 &= f_1(x1(i) + \Delta h \cdot \kappa_3, x2(i) + \Delta h \cdot \lambda_3, x3(i) + \Delta h \cdot \xi_3) \\
\lambda_4 &= f_2(x1(i) + \Delta h \cdot \kappa_3, x2(i) + \Delta h \cdot \lambda_3, x3(i) + \Delta h \cdot \xi_3) \\
\xi_4 &= f_3(x1(i) + \Delta h \cdot \kappa_3, x2(i) + \Delta h \cdot \lambda_3, x3(i) + \Delta h \cdot \xi_3)
\end{aligned}
\tag{4}
$$

C) The design of numeric-based Jerk oscillator on FPGA

In this step, Jerk chaotic system has been modeled using RK4 numerical algorithm with 32 bit IQ-Math fixed point number format (16I-16Q) in FPGA and has been coded in VHDL as a hardware definition language. The designed unit has been synthesized for VIRTEX-6 chip VC6VLX240T tool FF1156 package and tested. The multiplier, adder and subtractor units, that are suitable with fixed point number standard and have been utilized in the performed designs, have been created using IP CORE Generator which is developed with Xilinx ISE Design. The top level block diagram of RK4-based chaotic Jerk oscillator designed on FPGA is given in Fig 2. *Start* and *Clk* are one bit signals that have been located in the inputs of the units. They are responsible for timing the sub-units inside the units and maintaining the synchronization between the units and their connected system. *Δh* which identifies the precision of the algorithm, states the step size parameter. The initial values (x1(0), x2(0), x3(0)) and the step size (*Δh*) have been applied from the outside of the implementation to maintain more flexible design. The initial values are needed for the system startup. On the other hand, the system parameters of the chaotic system have been integrated into the implementation by defining 32 bit numeric values so as to decrease FPGA chip source consumption. The designed RK4-based chaotic oscillator include three 32-bit output signals, namely *X1_out*, *X2_out* and *X3_out*and one bit *Ready* signal for illustrating the output signals are ready.

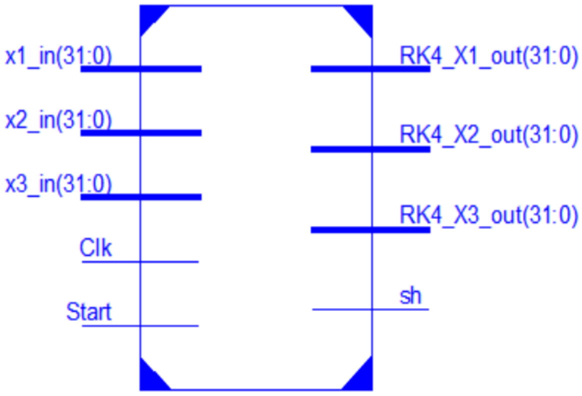

Fig 2. The top level block diagram of FPGA-based chaotic Jerk oscillator unit.

Fig 3 presents the second top level block diagram of chaotic Jerk unit. The second top level block diagram of the system contains *MuxUnit*, *Chaotic Jerk Unit* and *Filter Unit.* The purpose of using *Mux Unit* in the design is to enable taking the initial condition values from $x1_0$, $x2_0$ and $x3_0$ signals which are the initial condition values assigned by the user in fixed point number format when the system is at startup and to enable taking these values from *Filter Unit* at any step thereafter. *Filter Unit* has been used for hinder chaotic oscillator producing the unwanted signals, stated in other words it has been used for filtering. As the chaotic system produces result, *Ready* signal takes the value of *'1'* otherwise *Ready* signal produces *'0'.* In this way, while the system produces the first results, *Ready*signal takes the value of *'1'* , transmits this signal to *Mux Unit* and permits *Mux Unit* utilizing the produced values of chaotic system on behalf of utilizing the initial values devoted by the user.

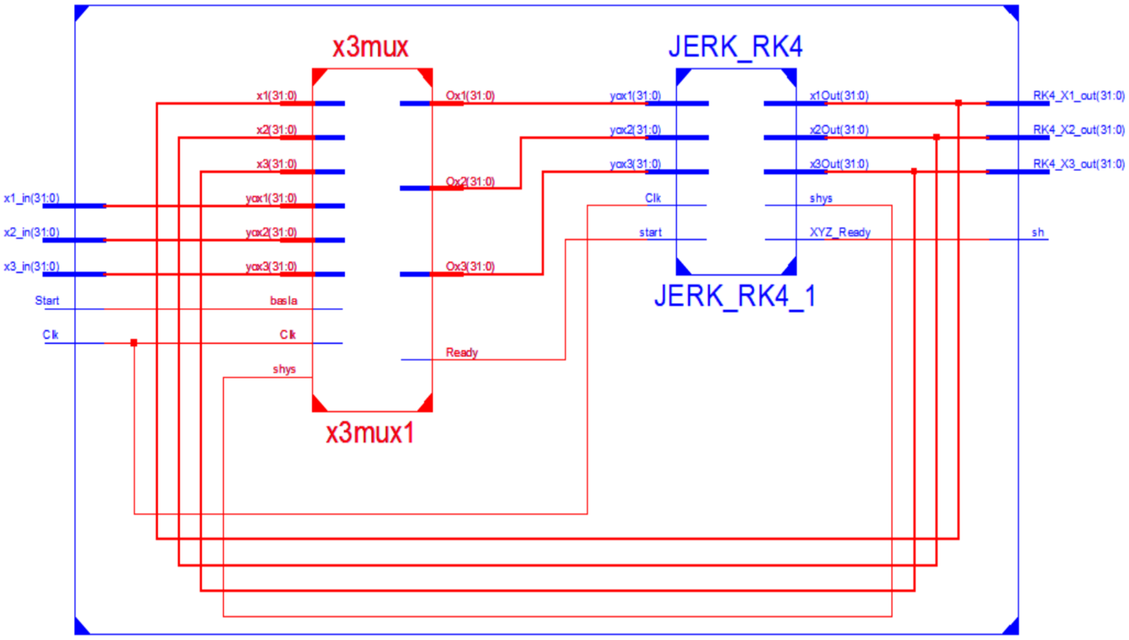

Fig 3. The second top level block diagram of FPGA-based chaotic Jerk unit.

The RK4-based chaotic Jerk chaotic oscillator includes 7 unit namely *Mux*, k_1, k_2, k_3, k_4, *ys* and *Filter Units.* The k_1, k_2, k_3 and k_4 units compute κ_σ, ξ_σ and λ_σ values in the discreted model of the system for $\sigma=1\ldots4$. The $x1(i+1)$, $x2(i+1)$ and $x3(i+1)$values shown in Eq. 3 in RK4 algorithm are computed in *ys* unit. A *Filter* unit has been utilized for avoiding undesired signals which are arriving to the output while the chaotic oscillator has not generated any result. Thus and so the whole undesired signals are filtered and thereby only the desired signals have been sent to the output. The signals obtained from the Filter unit present not only the signal outputs of the system (*X1_out*, *X2_out* and *X3_out*) but also the new initial conditions that is going to be sent to *Mux* unit for the calculation of the next algorithm. The unit runs in pipelined manner and chaotic oscillator generates the first outputs after 137 clock cycles.

III.EXPERIMENTAL RESULTS

A) The comparison of chaotic Jerk oscillator on FPGA and ANN-based design

RK4-based Jerk oscillator unit has been synthesized for Xilinx Virtex–6 family VC6VLX240T chip and the statistics of parameters related to clock frequencies of units and FPGA chip resource utilization have been examined. The data processing time of designed embedded chaotic oscillator units have been obtained using Xilinx ISE Design Tools 14.2 simulation program. Here, the time series values of *X1_out*, *X2_out* and *X3_out* signals, which correspond to “*x1*”, “*x2*” and “*x3*” signals in chaotic oscillator unit obtained from FPGA implementation using ISE Design Tools, have been illustrated in 32 bit fixed point number format. The results of RK4-based Jerk chaotic system designed on FPGA in fixed point number format and obtained from Xilinx ISE simulator have been given in Fig. 4.

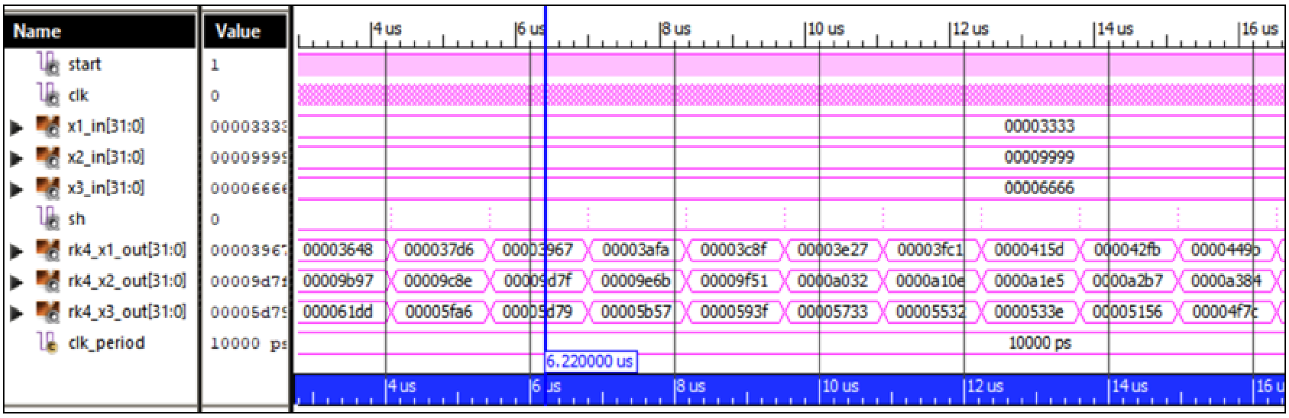

Fig 4. The Xilinx ISE simulator results of RK4-based Jerk chaotic oscillator unit.

Table 1 presents the chip statistics of Xilinx Virtex-6 VC6VLX240T-1ff1156 obtained from synthesizing process of Jerk chaotic oscillator designed on FPGA with respect to not only RK4-based 32 bit IQ-Math fixed point number format but also ANN-based floating point number format. As can be observed from the chip statistics, maximum operating frequency of RK4-based chaotic Jerk oscillator on FPGA reaches 373,128 MHz and chip utilization ratio is under %15. The chip statistics of ANN-based Jerk chaotic system have been given in 32 bit IEEE 754-1985 floating point number format on FPGA in 2018 by Koyuncu et al [38]. The maximum operating frequency of ANN-based Jerk oscillator unit on FPGA reaches 231,616 MHz. Besides, chip utilization ratio is greater than %30. Accordingly, it has been observed that the RK4-based Jerk chaotic system has not only greater operating frequency but also lower chip resource ratio than ANN-based Jerk chaotic system.

Table1. FPGA chip statistics of RK4-based and ANN-based Jerk system.

Device Utilization Summary (estimated values)	**RK4-Based Jerk chaotic system**	**ANN-Based Jerk chaotic system**
	Used / Utilization (%)	
Number of Slice Registers	5455 / 1	96264 / 31
Number of Slice LUTs	4828 / 3	88672 / 58
Number of fully used LUT-FF pairs	3736 / 57	5985 / 91
Number of bonded IOBs	195 / 32	195 /27
Number of BUFG/BUFGCTRLs	1 / 3	1 / 3
Number of DSP48E1s	136 / 32	8 / 1
Max. Clock Frequency (MHz)	**373,128**	**231,616**

IV.CONCLUSION

In this study, the implementation of 3-B Jerk chaotic system has been carried out using RK4 numerical algorithm in VHDL 32-bit IQ-Math (16I-16Q) fixed point number format on FPGA. RK4-based chaotic Jerk oscillator has been synthesized and tested for Xilinx Virtex-6 FPGA chip and the maximum operating frequency of the designed chaotic oscillator has been measured as 373,128 MHz. Then, the chip utilization statistics of ANN-based Jerk chaotic system has been compared with the chip utilization statistics of numerical-based Jerk chaotic system in fixed point number format. Accordingly, it has been understood that the numeric RK4-based chaotic Jerk system in fixed point number format has not only greater operating frequency but also lower chip resource ratio. Embedded chaos based engineering applications, including synchronization, data encryption, chaos-based secure communication and true random number generators with a lower cost and faster can be performed using the Jerk chaotic oscillator designed with 32 bit IQ-Math fixed point number format-based RK4 numerical algorithm in FPGA and presented to literature.

REFERENCES

[1] Pamuk N. Determination of Chaotic Time Series in Dynamic Systems. BAÜ Fen Bil Enst Derg Cilt 2013;15:77–91.

[2] Yılmaz D, Güler NF. Kaotik Zaman Serisinin Analizi Üzerine bir Araştırma. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Derg 2006;21:759–79.

[3] Tuna M, Fidan CB. 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 2018;33:469–86.

[4] Alcin M. The Effect on Modelling Performance of Different Activation Functions for Feed Forward and Feedback Network Structures in Modeling of Chen Chaotic System. Int J Sci Technol Res 2017;3:60–70.

[5] Alçin M. The Modelling Performance Evolution of Recurrent Neural Networks Having Different Training Functions for Modelling Sprott H Chaotic System. Int J Res Innov Eng Sci Technol 2017;2:563–8.

[6] Wang X, Akgul A, Cicek S, Pham V-T, Hoang DV. A Chaotic System with Two Stable Equilibrium Points: Dynamics, Circuit Realization and Communication Application. Int J Bifurc Chaos 2017;27:1750130.

[7] Ji Y, Zhang M, Wang Y, Wang P, Wang A, Wu Y, et al. Microwave-Photonic Sensor for Remote Water-Level Monitoring Based on Chaotic Laser. Int J Bifurc Chaos 2014;24:1450032.

[8] Avaroğlu E, Koyuncu İ, Özer AB, Türk M. Hybrid pseudo-random number generator for cryptographic systems. Nonlinear Dyn 2015;82:239–48.

[9] Ashita S, Uma G, Deivasundari P. Chaotic dynamics of a zero average dynamics controlled DC–DC Ćuk converter. IET Power Electron 2014;7:289–98.

[10] Pomares J, Perea I, Torres F. Dynamic Visual Servoing With Chaos Control for Redundant Robots. IEEE/ASME Trans Mechatronics 2014;19:423–31.

[11] Alçın M, Pehlivan İ, Koyuncu İ. Hardware design and implementation of a novel ANN-based chaotic generator in FPGA. Opt - Int J Light Electron Opt 2016;127:5500–5.

[12] Koyuncu I, Alcin M, Pehlivan I. Electronic circuit realization and synchronization application of Sprott 94 S chaotic system for secure communication systems. 2013 21st Signal Process. Commun. Appl. Conf., IEEE; 2013, p. 1–4.

[13] Wu J, Liao X, Yang B. Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation. Signal Processing 2018;142:292–300.

[14] Çavuşoğlu Ü, Kaçar S, Pehlivan I, Zengin A. Secure image encryption algorithm design using a novel chaos based S-Box. Chaos, Solitons & Fractals 2017;95:92–101.

[15] Koyuncu I, Ozcerit AT, Pehlivan I. An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system. Optoelectron Adv Materıals-Rapıd Communıcatıons 2013;7:635–8.

[16] Çavuşoğlu Ü, Akgül A, Zengin A, Pehlivan I. The design and implementation of hybrid RSA algorithm using a novel chaos based RNG. Chaos, Solitons & Fractals 2017;104:655–67.

[17] Koyuncu İ, Turan Özcerit A. The design and realization of a new high speed FPGA-based chaotic true random number generator. Comput Electr Eng 2017;58:203–14.

[18] Bonny T, Al Debsi R, Majzoub S, Elwakil AS. Hardware Optimized FPGA Implementations of High-Speed True Random Bit Generators Based on Switching-Type Chaotic Oscillators. Circuits, Syst Signal Process 2018:1–18.

[19] Bonny T, Elwakil AS, Bonny T, Elwakil AS. FPGA realizations of high-speed switching-type chaotic oscillators using compact VHDL codes. Nonlinear Dyn n.d.

[20] Koyuncu I, Ozcerit AT, Pehlivan I. Implementation of FPGA-based real time novel chaotic oscillator. Nonlinear Dyn 2014;77:49–59.

[21] Qiu M, Yu S, Wen Y, Lü J, He J, Lin Z. Design and FPGA Implementation of a Universal Chaotic Signal Generator Based on the Verilog HDL Fixed-Point Algorithm and State Machine Control. Int J Bifurc Chaos 2017;27:1750040.

[22] Tuna M, Koyuncu I, Fidan CB, Pehlivan I. Real time implementation of a novel chaotic generator on FPGA. 2015 23nd Signal Process. Commun. Appl. Conf., IEEE; 2015, p. 698–701.

[23] Tuna M, Fidan CB, Koyuncu I, Pehlivan I. Real time hardware implementation of the 3D chaotic oscillator which having golden-section equilibra. 2016 24th Signal Process. Commun. Appl. Conf., IEEE; 2016, p. 1309–12.

[24] Sadoudi S, Azzaz MS, Djeddou M, Benssalah M. An FPGA Real-time Implementation of the Chen's Chaotic System for Securing Chaotic Communications. Int J Nonlinear Sci 2009;7:1749–3889.

[25] Rajagopal K, Akgul A, Jafari S, Karthikeyan A, Koyuncu I. Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses. Chaos, Solitons & Fractals 2017;103:476–87.

[26] Guang-Yi W, Xu-Lei B, Zhong-Lin. Design and FPGA Implementation of a new hyperchaotic system. Chinese Phys B 2008;17:3596–602.

[27] Alçın M, Pehlivan İ, Koyuncu İ. The Performance Analysis of Artificial Neural Network Based Shimizu-Morioka Chaotic System with Respect to Sample Numbers. Balk J Electr Comput Eng 2015;3:1–5.

[28] Sahin I, Koyuncu I. Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSig Activation Functions on FPGA. Electron Electr Eng 2012;120:51–4.

[29] Namjin Kim, Kehtarnavaz N, Yeary MB, Thornton S. DSP-based hierarchical neural network modulation signal classification. IEEE Trans Neural Networks 2003;14:1065–71.

[30] Yu B, Chan RHM, Mak T, Sun Y, Poon C-S. On-Chip Systolic Networks for Real-Time Tracking of Pairwise Correlations Between Neurons in a Large-Scale Network. IEEE Trans Biomed Eng 2013;60:198–202.

[31] KOYUNCU I, I. Implementation of High Speed Tangent Sigmoid Transfer Function Approximations for Artificial Neural Network Applications on FPGA. Adv Electr Comput Eng 2018;18:79–86.

[32] Koyuncu I, Cetin O, Katircioglu F, Tuna M. Edge dedection application with FPGA based Sobel operator. 2015 23nd Signal Process. Commun. Appl. Conf., IEEE; 2015, p. 1829–32.

[33] Koyuncu I. Design and Implementation of High Speed Artificial Neural Network Based Sprott 94 S System on FPGA. Int J Intell Syst Appl Eng 2016;4:33.

[34] Koyuncu İ, Şahin İ, Gloster C, Sarıtekin NK. A Neuron Library for Rapid Realization of Artificial Neural Networks on FPGA: A Case Study of Rössler Chaotic System. J Circuits, Syst Comput 2017;26:1750015.

[35] Lei Zhang. Artificial Neural Network model design and topology analysis for FPGA implementation of Lorenz chaotic generator. 2017 IEEE 30th Can. Conf. Electr. Comput. Eng., IEEE; 2017, p. 1–4.

[36] Tuna M, Koyuncu İ, Alçın M. Fixed and Floating point-Based High-Speed Chaotic Oscillator Design with Different Numerical Algorithms on FPGA. Int J Adv Res Electr Electron Instrum Eng 2018;7:3179–87.

[37] Tuna M, Fidan CB, Koyuncu İ, Pehlivan İ. FPGA-Based IQ-Math Number Format Suitable High Speed Chaotic Oscillator Design. 4th Int. Symp. Innov. Technol. Eng. Sci. 3-5 Nov 2016, Alanya/Antalya - Turkey: 2016, p. 354–63.

[38] Koyuncu İ, Oğuz Y, Çimen H, Özer T, Tuna M. Design and Implementation of Artificial Neural Network-Based 3-D Novel Jerk Chaotic Oscillator on FPGA. 3rd Int. Conf. Eng. Technol. Appl. Sci. July 17-21 2018 Skopje Maced., 2018, p. 1–6.