CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.2 / No.1 / 2020 / pp.23-30

The Runge Kutta-4 based 4D Hyperchaotic System Design for Secure Communication Applications

Murat Alcin *,[1]
*Department of Mechatronics Engineering, Faculty of Technology, Afyon Kocatepe University, Afyon 03200, Turkey.

ABSTRACT It is a common fact that warranting high security is the leading isssue in the communication systems. Since there will always be possible threats in attacking the communication channels, the trends to determine a suitable method to quarantee high security for communication systems will continue. This paper presents an FPGA-based 4D hyperchaotic system to be utilized in the communication systems. Since the system has 9 terms, two of which are nonlinear, it is one of the among simplest 4D hyperchaotic systems. Fourth order Runge-Kutta numeric algorithm (RK4) has been utilized to get the discrete time tantamount of the given hyperchaotic system. IEEE 32-bit 754-1985 floating point standart of single precision has been used for defining the numbers. The whole design has been coded in Very High–Speed Integrated Circuit Hardware Description Language (VHDL). The implemented hyperchaotic system has been simulated and synthesised by utilizing Xilinx ISE Design Tools 14.7 software in a Xilinx Virtex-7 XC7VX330T chip. After the Place-Route process, chip area consumption statistics and the clock frequency parameters were got and analyzed. Finally, the maximum clock frequency of the 4D hyperchaotic oscillator reaches 350.733 MHz so the proposed design can be utilized as a chaotic oscillator in enhancing chaos-based communication systems on FPGA.

KEYWORDS
Runge Kutta-4
4D Hyperchaotic system
FPGA
Secure communication

INTRODUCTION

A chaos based communication system is one of those security methods in modern communication area Murillo-Escobar *et al.* (2017); Kocamaz *et al.* (2018). Chaos has been advanced by scientists, engineeers and mathematicians Bonilla *et al.* (2016); Liu *et al.* (2013). It is the discipline that is interested in the behaviours of nonlinear dynamic systems. Due to having the characteristics of deterministic, aperiodic, nonlinear and sensitivity to initial states, the behaviours of the chaotic systems can not be predicted for long time period Tuna and Fidan (2016); Pamuk (2013); Tuna and Fidan (2018); Pehlivan *et al.* (2019). Hence, the systems that have a strong chaotic oscillator operating in chaotic regime as an entropy source are obviously required in every engineering fields that deal with chaotic systems especially, in the communication systems Alçın *et al.* (2018); Koyuncu and Özcerit (2017); Pehlivan and Zhouchao (2012). These systems have been largely utilized in the areas of cryptology, industrial electronics, wireless communication, Artificial Neural Networks (ANN), secure communication, biomedical, optic electronics, biophysics, mathematics, electromagnetic, mechatronics, fuzzy logic, optimization and data transmission Koyuncu (2016); Avaroğlu *et al.* (2015a); Alçın *et al.* (2016); Tuncer (2016); Jahanshahi *et al.* (2018); Xu *et al.* (2018). In literature, the hardware-based implementations of chaotic systems have been carried out by using different technologies including ASIC (Application Specific Integrated Circuit), Microprocessors, PIC (Peripheral Interface Controller) and FPGA (Field Programmable Gate Array) Zhang *et al.* (2008); Zuppa (2010); Akgul (2017); Koyuncu and Şeker (2019).

FPGA provides higher parallelism and speed than the alternative processors. FPGA is a predefined reconfigurable Integrated Circuit (IC) device without the need of any external components. FPGA can be used to realize any function that is needed. Its configuration is broadly described by a Hardware Description Language (HDL). FPGA has been used for hardware-based applications where the execution speed of the operations is most required. Therefore, FPGA-based architectures are suitable for hardware implementation of chaotic or hyperchaotic systems Lai *et al.* (2018); Rajagopal *et al.* (2018). It is too important to define the sensitivity of the numbers as high as possible during numeric solution of the chaotic systems. Since, the sensitivity has a huge effect on the solutions of the chaotic systems, the RK-4 numeric algorithm,

Manuscript received: 1 April 2020,
Revised: 12 April 2020,
Accepted: 1 May 2020.

[1] muratalcin@aku.edu.tr (**Corresponding Author**)

which supplies higher precision, has been used for the numeric solution of the chaotic systems. Prior to the hardware-based implementation of a chaotic system in FPGA, the number format and the number sensitivity must be kept in mind. The rise of the sensitivity of the numbers dramatically increases the resources utilized. Well-recognized of the number representations concerning precision are floating and fixed point representations Elmanfaloty and Abou-Bakr (2019). In this study, since the requirement for sensitivity is high, therefore rather than fixed point representation, single precision IEEE-754 floating point representation has been utilized.

In the last years FPGA, having competency of parallel processing, is the leading candidate of embedded system environments for applications of chaotic systems. Chaotic oscillators have been extensively used in many areas; for instance random number generators, secure communication, synchronization and data encryption. In this paper, since the proposed chaotic oscillator has been implemented in real time, this study has demonstrated that this oscillator can be utilized in future studies related to secure communication. In literature there have been numerous hardware-based chaotic system applications on FPGAs.

Prakash et al. designed a novel four-dimensional hyperchaotic system based on FPGA for enhancing embedded chaos-based engineering applications Prakash *et al.* (2020). Vaidyanathan et al. implemented a new 4-D two-disk dynamical system based on ANN that demonstrates hyperchaos and hidden attractor on FPGA. Then they realize a new high-speed True Random Number Generator (TRNG) utilizing FPGA-based design of this system Vaidyanathan *et al.* (2020). A new dual entropy core TRNG architecture based on chaos and ring on FPGA was introduced to literature by Koyuncu et al. The proposed TRNG includes 3D continuous time autonomous chaotic oscillator and ring oscillator. By uniting these oscillators they implemented a dual entropy core TRNG unit that has throughput ranging 464 Mbps Koyuncu *et al.* (2020). Akgul et al. has proposed a 3D chaotic system without equilibrium points and they have realized the circuit application of that system. In addition they have modelled this chaotic system on Labview FPGA Akgul *et al.* (2016).

In another study presented to the literature by Tuna et al., a hyper jerk multiscroll chaotic system is utilized in the FPGA implementation of ANN-Ring-based TRNG Tuna *et al.* (2019b). Also, Rajagopal et al. has been implemented chaotic memristor Hopfield neural network system on FPGA by utilizing RK4 numeric algorithm. The system based on FPGA has been designed in VHDL Rajagopal *et al.* (2019). To enhance embedded chaos-based engineering applications, Lü-Chen chaotic system is designed on FPGA by utilizing Heun algorithm in VHDL by Tuna et al. Tuna *et al.* (2019a). In addition to these studies, Avaroglu et al. presented a chaos-based post-processing technique as an alternative to other post-processing techniques in the literature Avaroğlu *et al.* (2015b). Random number generation of LFSR based stream encryption algorithms and their hardware implementations have been proposed by Tuncer et al. Tuncer and Avaroğlu (2017). Table 1 demonstrates the technical properties of the implemented chaotic systems-based signal generators on FPGA.

Remaining sections of the study are organized as follows: the dynamics of the 4D hyperchaotic system are introduced in Section 2; design parameters and structure of the proposed 4D hyperchaotic system on FPGA are explained in Section 3. Lastly, results are discussed in conclusion section.

DYNAMICS OF THE 4D HYPERCHAOTIC SYSTEM

The dynamics of the 4D Hyperchaotic system has been presented as follows:

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

where x, y, z and w are the state variables. The 4D Hyperchaotic System demonstrates hyperchaotic behavior when $a_1 = 30, a_2 = 20, a_5 = 3, a_7 = 0.1$, and initial states as $x(0), y(0), z(0), w(0)) = 0.1, 0.1, 0.1, 0.1$. Thus, a nominal set of parameters for which the system in (1) demonstrates hyperchaos has been selected as $a_1 = 30, a_2 = 20, a_5 = 3, a_7 = 0.1$.

FPGA-BASED 4D HYPERCHAOTIC SYSTEM

FPGA-based chaotic oscillator studies are more important in the literature Tuna *et al.* (2019a); Tlelo-Cuautle *et al.* (2016); Koyuncu *et al.* (2013); Rashtchi and Nourazar (2015). In this part, the RK4-Based 4D HCO system was modelled as a model of FPGA-based using RK4 numeric algorithm with 32-bit IEEE 754-1985 floating point notation of single precision and was designed with VHDL. The units including adder, multiplier, and subtractor complying with the floating-point standard were produced using IP CORE Generator that has been created with Xilinx ISE Design Tools. The top-level block schema of RK4-Based 4D HCO unit designed on FPGA is given in Fig. 1.

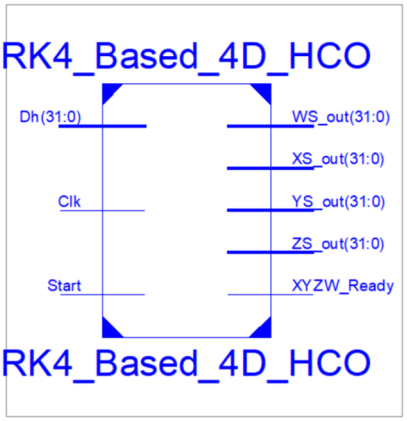

Figure 1 The top-level block schema of the RK4-Based 4D HCO system on FPGA

Table 1 The technical properties of chaotic systems performed on FPGA in recent years.

References	Used algorithm	Used number standard	FPGA properties	Operating Frequency (MHz)
Koyuncu et al., Koyuncu and Özcerit (2017)	Numeric Algorithm (RK4)	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex-6 XC6VLX240T-1-FF1156	293.815
Koyuncu I, Koyuncu (2016)	Artificial Neural Network-based	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex6 XC6VHX255T-3FF1923	304.534
Avaroglu et al., Avaroğlu *et al.* (2015a)	Numeric Algorithm (RK5)	IEEE 32-bit 754-1985 Floating point	Xilinx XC6VLX550T	339.000
Alçın et al., Alçın *et al.* (2016)	Artificial Neural Network-based	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex6 XC6VCX240T	266.429
Vaidyanathan et al., Vaidyanathan *et al.* (2020)	Artificial Neural Network-based	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex6 XC6VLX 240T-1FF1156	167.026
Rajagopal et al., Rajagopal *et al.* (2019)	Numeric Algorithm (RK4)	IEEE 32-bit 754-1985 Floating point	Xilinx XC6VLX240T-1-FF1156	231.616
Şeker et al., Koyuncu and Şeker (2019)	Numeric Algorithm (Dormand Prince)	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex-6 XC6VLX240T-1FF1156	316.756
Rajagopal et al., Rajagopal *et al.* (2017)	Numeric Algorithm (RK5)	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex-6 XC6VLX240T-1FF1156	325.759
This article	Numeric Algorithm (RK4)	IEEE 32-bit 754-1985 Floating point	Xilinx Virtex-7 XC7VX330T	350.733

Fig. 2 gives second level block schema of the RK4-Based 4D HCO system on FPGA. It consists of a $x4mux$ multiplexer unit and RK4-based Hyperchaotic oscillator unit (HCO RK4). The aim to use $x4mux$ unit is to enable the initial states taking their values in the system startup phase from $y0x1$, $y0x2$, $y0x3$, and $y0x4$ signals which are 32 bit initial signals with floating point number representation assigned by the user and is to enable them taking their values from the output of RK4-Based 4D HCO system unit. The *Shys* signal having one bit is '1' if the hyperchaotic system produces the output, otherwise it is '0'. Thus, when HCO RK4 unit produces the first output value, *Shys* will be '1' and this signal makes $x4mux$ use the values generated by the hyperchaotic system instead of using the initial values assigned by the user.

The discretised model of Four-Dimensional Hyperchaotic system utilizing RK4 algorithm was shown in (3) by organizing in terms of f, g, δ, and σ functions in (2). $x(k+1), y(k+1), z(k+1)$, and $w(k+1)$, which exemplify the values of the hyperchaotic system while step size rises as Δh, were computed by replacing initial states and parameters of the system in RK4 algorithm. $x(k+1), y(k+1), z(k+1)$, and $w(k+1)$ values, that are the system outputs in each recursion, were utilized as both the system outputs and system's new initial states for the next recursion.

$$\begin{aligned} \dot{x} &= f(t,x,y,z,w) = a_1 y - a_1 x \\ \dot{y} &= g(t,x,y,z,w) = -xz + a_2 y - 5w + 1 \\ \dot{z} &= \delta(t,x,y,z,w) = xy - a_5 z \\ \dot{w} &= \sigma(t,x,y,z,w) = a_7 y \end{aligned} \tag{2}$$

$$\begin{aligned} x\,(k+1) &= x(k) + \Delta h\ (a_1(y\,(k) - x(k))) \\ y\,(k+1) &= y(k) + \Delta h\ (-x\,(k)z(k) + a_2 y\,(k) - 5w(k) + 1) \\ z\,(k+1) &= z(k) + \Delta h\ (x(k)y(k) - a_5 z(k)) \\ w\,(k+1) &= w(k) + \Delta h\ (a_7 y(k)) \end{aligned} \tag{3}$$

Fig. 3 gives 3^{rd} degree block schema of RK4-based novel 4D Hyperchaotic oscillator unit. There have been 5 units including Multiplier, MUX, Adder, Filter, and f unit used in oscillator structure. The function of MUX unit is enabling the initial states and system parameters needed by the Four-Dimensional Hyperchaotic system to be assigned as the Start signal is active. That means, MUX unit makes a selection between the initial states of the system assigned by user and $x(k+1), y(k+1), z(k+1), w(k+1)$ signals obtained from the system output. The selected initial states have been utilized as initial states for calculating the next recursion. The initial states, which are one of the inputs of MUX unit, represent the initial states of the hyperchaotic system. The f unit in system calculates the equations of the novel 4D Hyperchaotic system using control signals of MUX unit. Signals coming from f unit and Δh value, i.e. the step size of algorithm, are multiplied by Multiplier Unit. Then the Adder Unit sums the values of the initial conditions and sends the results to the Filter Unit. Preventing hyperchaotic oscillator to produce any unwanted signals has been granted by Filter Unit, in other words it is used for filtering. The RK4-based new Four-Dimensional Hyperchaotic oscillator unit on FPGA operates in pipeline and produces the first output after 53 clock cycle.

New 4D Hyperchaotic oscillator unit based on RK4 algorithm has been simulated and synthesised in the Xilinx Virtex-7 XC7VX330T chip and the chip statistics related to FPGA area usage and the parameters of the clock frequency were analyzed. The data operation time of novel 4D Hyperchaotic oscillator unit was assessed with Xilinx ISE Design Tools 14.7 program. The values connected with time series of XS_out, YS_out, ZS_out and WS_out signals corresponding to x, y, z, and w signals got from FPGA realization using Hyper-chaotic oscillator in ISE Design Tools have been shown in 32 bit floating point format. Hyper-chaotic oscillator unit's results based on RK4 algorithm of operation timing diagram got from Xilinx ISE simulator have been illustrated in Fig. 4.

4D Hyperchaotic oscillator unit based on RK4 algorithm was synthesised and then following the Place-Route processes, the statistics of Xilinx Virtex-7 family XC7VX330T-2-FFG-1157 FPGA chip were got. As shown in the Table 2, the maximum clock frequency of the 4D Hyperchaotic oscillator reaches 350.733 MHz.

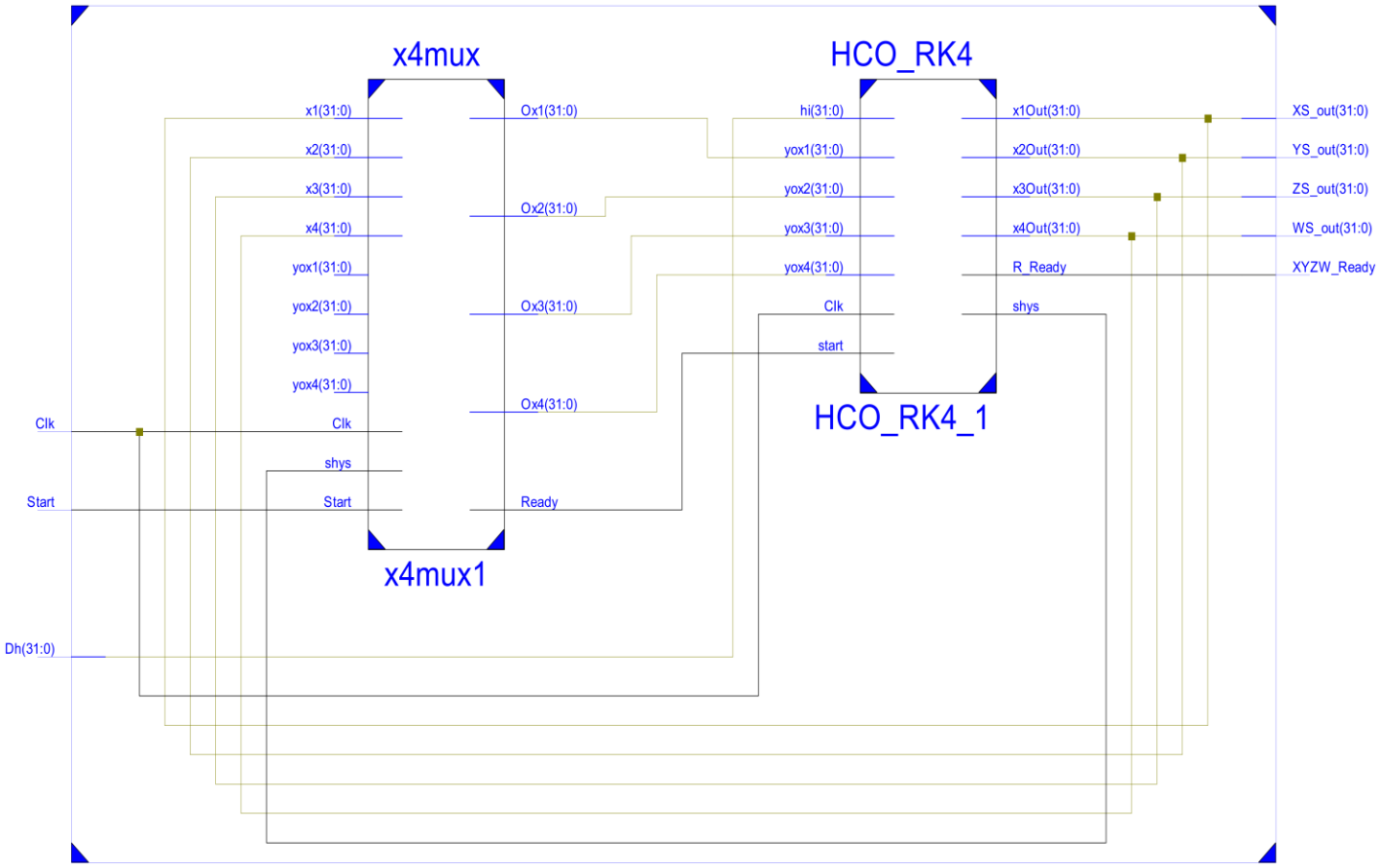

Figure 2 Second level block schema of the novel hyperchaotic oscillator unit on FPGA.

Table 2 The area utilization report of RK4-based 4D Hyperchaotic oscillator on FPGA.

Logic Utilization	Used	Available	Utilization (%)
Number of Slice Registers	57,580	408,000	14
Number of Slice LUTs	57,230	204,000	28
Number of fully used LUT-FF pairs	46,393	68,417	67
Number of bonded IOBs	163	600	27
Number of BUFG/ BUFGCTRLs	1	32	3

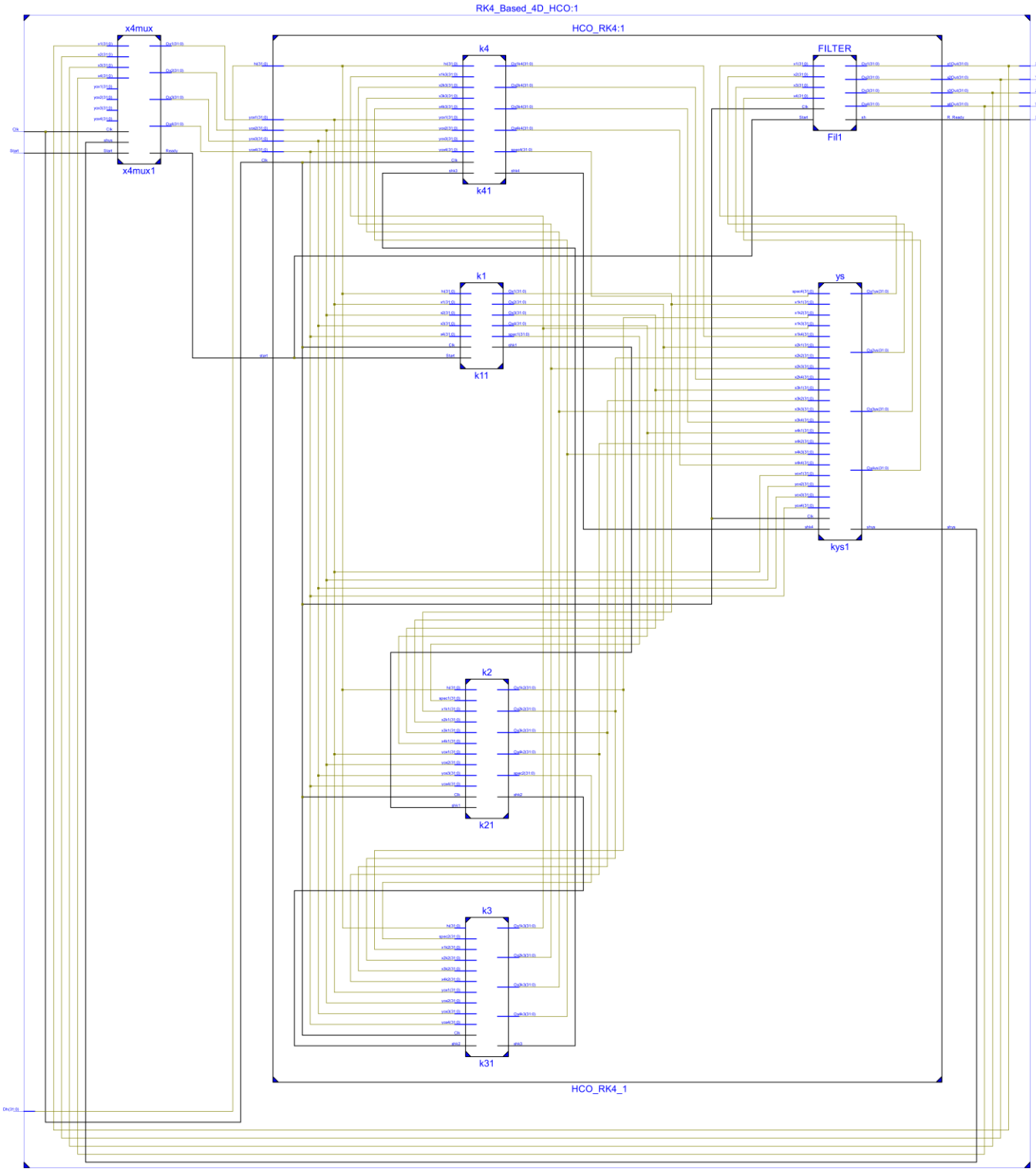

Figure 3 The 3^{rd} degree block schema of RK4 based new Four-Dimensional Hyperchaotic oscillator unit.

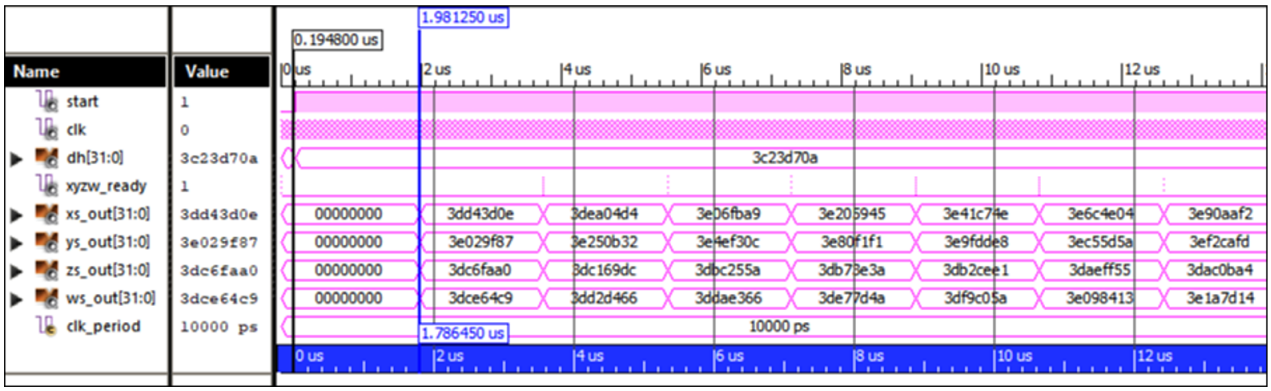

Figure 4 The results of RK4-based new 4D Hyperchaotic oscillator unit operation timing diagram got in Xilinx ISE Simulator.

RESULTS AND DISCUSSION

In this study, a 4D hyperchaotic system based on FPGA to be utilized in the communication systems is presented. The 4D hyperchaotic system has 9 terms, two of which are nonlinear, therefore it is one of the among simplest Four-Dimensional hyperchaotic systems. For the discrete time tantamount of the given Four-Dimensional hyperchaotic system, RK4 numeric algorithm has been choosen. To define the numbers used in the implementation process, IEEE 32-bit 754-1985 floating point representation of single precision has been preferrred due to the advantage of the precision it has. All design operation has been described using VHDL. A Xilinx Virtex-7 XC7VX330T chip in the Xilinx ISE Design Tools 14.7 software has been utilized for the simulating and synthesizing the implemented 4D hyperchaotic system. The chip area consumption statistics and the clock frequency parameters were acquired and analyzed following the Place-Route process. The maximum clock frequency of the 4D hyperchaotic oscillator achieves 350.733 MHz. In adddition, since minimum clock period of the proposed hyperchaotic oscillator is approximately 2.85 nsn, the proposed oscillator generates 4x350,000 results in 1 sn. As a result, the recommended design can be used as a chaotic oscillator in enhancing chaos-based communication systems on FPGA.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

LITERATURE CITED

Akgul, A., 2017 An electronic card for easy circuit realisation of complex nonlinear systems. Electron. World **124**: 29–31.

Akgul, A., H. Calgan, I. Koyuncu, I. Pehlivan, and A. Istanbullu, 2016 Chaos-based engineering applications with a 3d chaotic system without equilibrium points. Nonlinear dynamics **84**: 481–495.

Alçın, M., İ. Pehlivan, and İ. Koyuncu, 2016 Hardware design and implementation of a novel ann-based chaotic generator in fpga. Optik **127**: 5500–5505.

Alçın, M., M. Tuna, and İ. Koyuncu, 2018 Iq-math based designing of fourth order runge-kutta algorithm on fpga and performance analysis according to ann approximation .

Avaroğlu, E., I. Koyuncu, A. B. Özer, and M. Türk, 2015a Hybrid pseudo-random number generator for cryptographic systems. Nonlinear Dynamics **82**: 239–248.

Avaroğlu, E., T. Tuncer, A. B. Özer, B. Ergen, and M. Türk, 2015b A novel chaos-based post-processing for trng. Nonlinear Dynamics **81**: 189–199.

Bonilla, L. L., M. Alvaro, and M. Carretero, 2016 Chaos-based true random number generators. Journal of Mathematics in Industry **7**: 1–17.

Elmanfaloty, R. A. and E. Abou-Bakr, 2019 Random property enhancement of a 1d chaotic prng with finite precision implementation. Chaos, Solitons & Fractals **118**: 134–144.

Jahanshahi, H., K. Rajagopal, A. Akgul, N. N. Sari, H. Namazi, *et al.*, 2018 Complete analysis and engineering applications of a megastable nonlinear oscillator. International Journal of Non-Linear Mechanics **107**: 126–136.

Kocamaz, U. E., S. Çiçek, and Y. Uyaroğlu, 2018 Secure communication with chaos and electronic circuit design using passivity-based synchronization. Journal of Circuits, Systems and Computers **27**: 1850057.

Koyuncu, I., 2016 Design and implementation of high speed artificial neural network based sprott 94 s system on fpga. International Journal of Intelligent Systems and Applications in Engineering **4**: 33–39.

Koyuncu, I. and A. T. Özcerit, 2017 The design and realization of a new high speed fpga-based chaotic true random number generator. Computers & Electrical Engineering **58**: 203–214.

Koyuncu, I., A. T. Ozcerit, I. Pehlivan, *et al.*, 2013 An analog circuit design and fpga-based implementation of the burke-shaw chaotic system. Optoelectronics and Advanced Materials-Rapid Communications **7**: 635–638.

Koyuncu, İ. and H. İ. Şeker, 2019 Implementation of dormand-prince based chaotic oscillator designs in different iq-math number standards on fpga. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi **23**: 859–868.

Koyuncu, I., M. Tuna, I. Pehlivan, C. B. Fidan, and M. Alçın, 2020 Design, fpga implementation and statistical analysis of chaos-ring based dual entropy core true random number generator. Analog Integrated Circuits and Signal Processing **102**: 445–456.

Lai, Q., X.-W. Zhao, K. Rajagopal, G. Xu, A. Akgul, *et al.*, 2018 Dynamic analyses, fpga implementation and engineering applications of multi-butterfly chaotic attractors generated from generalised sprott c system. Pramana **90**: 6.

Liu, Y., S. Pang, and D. Chen, 2013 An unusual chaotic system and its control. Mathematical and computer modelling **57**: 2473–2493.

Murillo-Escobar, M., C. Cruz-Hernández, L. Cardoza-Avendaño, and R. Méndez-Ramírez, 2017 A novel pseudorandom number

generator based on pseudorandomly enhanced logistic map. Nonlinear Dynamics **87**: 407–425.
Pamuk, N., 2013 Determination of chaotic time series in dynamic systems. Journal of Balikesir University Institute of Science and Technology **15**: 77–91.
Pehlivan, İ., E. Kurt, Q. Lai, A. Basaran, and M. C. Kutlu, 2019 A multiscroll chaotic attractor and its electronic circuit implementation. Chaos Theory and Applications **1**: 29–37.
Pehlivan, I. and W. Zhouchao, 2012 Analysis, nonlinear control, and chaos generator circuit of another strange chaotic system. Turkish Journal of Electrical Engineering and Computer Science **20**: 1229–1239.
Prakash, P., K. Rajagopal, I. Koyuncu, J. P. Singh, M. Alcin, *et al.*, 2020 A novel simple 4-d hyperchaotic system with a saddle-point index-2 equilibrium point and multistability: Design and fpga-based applications. Circuits, Systems, and Signal Processing pp. 1–22.
Rajagopal, K., A. Akgul, S. Jafari, A. Karthikeyan, and I. Koyuncu, 2017 Chaotic chameleon: Dynamic analyses, circuit implementation, fpga design and fractional-order form with basic analyses. Chaos, Solitons & Fractals **103**: 476–487.
Rajagopal, K., S. Jafari, A. Karthikeyan, A. Srinivasan, and B. Ayele, 2018 Hyperchaotic memcapacitor oscillator with infinite equilibria and coexisting attractors. Circuits, Systems, and Signal Processing **37**: 3702–3724.
Rajagopal, K., M. Tuna, A. Karthikeyan, İ. Koyuncu, P. Duraisamy, *et al.*, 2019 Dynamical analysis, sliding mode synchronization of a fractional-order memristor hopfield neural network with parameter uncertainties and its non-fractional-order fpga implementation. The European Physical Journal Special Topics **228**: 2065–2080.
Rashtchi, V. and M. Nourazar, 2015 Fpga implementation of a real-time weak signal detector using a duffing oscillator. Circuits, Systems, and Signal Processing **34**: 3101–3119.
Tlelo-Cuautle, E., A. Pano-Azucena, J. Rangel-Magdaleno, V. Carbajal-Gomez, and G. Rodriguez-Gomez, 2016 Generating a 50-scroll chaotic attractor at 66 mhz by using fpgas. Nonlinear dynamics **85**: 2143–2157.
Tuna, M., M. Alçın, İ. Koyuncu, C. B. Fidan, and İ. Pehlivan, 2019a High speed fpga-based chaotic oscillator design. Microprocessors and Microsystems **66**: 72–80.
Tuna, M. and C. B. Fidan, 2016 Electronic circuit design, implementation and fpga-based realization of a new 3d chaotic system with single equilibrium point. Optik **127**: 11786–11799.
Tuna, M. and C. B. Fidan, 2018 A study on the importance of chaotic oscillators based on fpga for true random number generating (trng) and chaotic systems. Journal of the Faculty of Engineering and Architecture of Gazi University **33**: 469–486.
Tuna, M., A. Karthikeyan, K. Rajagopal, M. Alcin, and İ. Koyuncu, 2019b Hyperjerk multiscroll oscillators with megastability: Analysis, fpga implementation and a novel ann-ring-based true random number generator. AEU-International Journal of Electronics and Communications **112**: 152941.
Tuncer, T., 2016 The implementation of chaos-based puf designs in field programmable gate array. Nonlinear dynamics **86**: 975–986.
Tuncer, T. and E. Avaroğlu, 2017 Random number generation with lfsr based stream cipher algorithms. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 171–175, IEEE.
Vaidyanathan, S., I. Pehlivan, L. G. Dolvis, K. Jacques, M. Alcin, *et al.*, 2020 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 **62**: 20–35.
Xu, G., Y. Shekofteh, A. Akgül, C. Li, and S. Panahi, 2018 A new chaotic system with a self-excited attractor: entropy measurement, signal encryption, and parameter estimation. Entropy **20**: 86.
Zhang, Y., Z. Liu, and X. Zheng, 2008 A chaos-based image encryption asic using reconfigurable logic. In *APCCAS 2008-2008 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1782–1785, IEEE.
Zuppa, L. A., 2010 Chaotic logistic map implementation in the pic12f629 microcontroller unit. IFAC Proceedings Volumes **43**: 167–170.

How to cite this article: Alcin, M. The Runge Kutta-4 based 4D Hyperchaotic System Design for Secure Communication Applications. Chaos Theory and Applications, 2(1), 23-30, 2020.