Available online at www.dergipark.org.tr/en

INTERNATIONAL ADVANCED RESEARCHES
and
ENGINEERING JOURNAL

Journal homepage: www.dergipark.org.tr/en/pub/iarej

e-ISSN: 2618-575X

International Open Access

Volume 04
Issue 03

December, 2020

Research Article

CCII current conveyor and dormand-prince-based chaotic oscillator designs for secure communication applications

Murat Alçın [a], ***Murat Tuna*** [b,*], ***İhsan Pehlivan*** [c] ***and İsmail Koyuncu*** [d]

[a]*Afyon Kocatepe Universtiy, Faculty of Technology, Department of Mechatronics, Afyonkarahisar 03200, Turkey*
[b]*Kırklareli Universtiy, Technical Sciences Vocational High School, Department of Electrical and Energy, Kirklareli 39000, Turkey*
[c]*Sakarya University of Applied Sciences, Faculty of Technology, Department of Electrical and Electronics Engineering, Sakarya 54187, Turkey*
[d]*Afyon Kocatepe Universtiy, Faculty of Technology, Department of Electrical and Electronics Engineering, Afyonkarahisar 03200, Turkey*

ARTICLE INFO

Article history:
Received 13 June 2020
Revised 05 August 2020
Accepted 05 September 2020

Keywords:
CCII
Dormand-Prince algorithm
FPGA
Nonlinear oscillator
VHDL

ABSTRACT

Chaos is one of the important research areas in recent years. The chaotic signal generator is one of the most basic structure in the chaos-based researches and applications. In this study, Sundarapandian-Pehlivan Chaotic Oscillator (SPCO) designs have been implemented in 2 different platforms as analog-based using Second-Generation Current Controlled Current Conveyor (CCII) and FPGA-based with one of the chaotic oscillator that has been presented to the literature namely Sundarapandian-Pehlivan system. The structure used for the design of CCII-based chaotic oscillator and the results obtained from the study have been presented. In the second phase, the design of SPCO has been realized in order to utilize for running in FPGA chips using Dormand-Prince (DP) numeric algorithm. The design has been coded in VHDL using 32-bit IEEE-754-1985 floating point representation. The designed system has been tested by synthesizing it in Xilinx ISE Design Tools program. Then, the test results obtained from DP-based SPCO structure have been presented. In the last phase, the designed system has been synthesized for VIRTEX-7 FPGA. FPGA chip resource consumption values that obtained after the Place-Route process are presented. According to the results, the maximum operating frequency of DP-based SPCO unit on FPGA is obtained as 362.608 MHz. In future studies, the designs of Pseudo Random Number Generator (RNG) and True RNG can be performed using DP-based SPCO unit implemented successfully in this study.

1. Introduction

Chaotic systems (CSs) are one of the areas that have been studied extensively in recent years [1–3]. CSs can be defined as nonlinear systems that exhibit an infinite number of non-periodic oscillations with a specific order within certain intervals [4,5]. The sensitivity to the initial conditions (ICs) and the system parameters are among the important characteristics of the CSs [6,7]. For this reason, in CSs, the slight changes in the parameter values or the ICs of the system may significantly affect the dynamic behavior of the system in time domain [8,9].

The chaotic signal generator is one of the basic structures used in chaos-based studies and applications [10–13]. Nowadays, the chaotic signal generators and Chaotic Oscillators (COs) can be created by using different platforms. The circuits constructed by using analog electronic circuit elements can be given as examples for one of the most basic structures of CO structures [14,15]. In these circuits, active circuit elements are used together with basic elements such as resistors and capacitors, which are passive circuit elements [15,16]. Operational Amplifier (Op-Amp), Operational Transconductance Amplifier (OTA) and Second-Generation Current Controlled Current Conveyor (CCII) can be given as examples for the analog circuit elements that used in the CO designs [17–19]. The Op-

* Corresponding author. Tel.: +90-288-214-1845; Fax: +90-288-214-1495.
E-mail addresses: muratalcin@aku.edu.tr (*M. Alçın*), murat.tuna@klu.edu.tr (*M. Tuna*), ipehlivan@subu.edu.tr (*İ. Pehlivan*), ismailkoyuncu@aku.edu.tr (*İ. Koyuncu*)
ORCID: 0000-0002-2874-7048 *(M. Alçın)*, 0000-0003-3511-1336 *(M. Tuna)*, 0000-0001-6107-655X *(İ. Pehlivan)*, 0000-0003-4725-4879 *(İ. Koyuncu)*
DOI: 10.35860/iarej.752321

Amp has high-gain and it is a direct coupling element that can perform many linear/non-linear signal processing operations. Today, there are Op-Amp elements operating at frequencies higher than 100 MHz and higher power values than 100 W. Although it is quite difficult to integrate Op-Amp elements with other analog or digital structures on the same chip, it is very suitable for integrating OTA element with analog or digital structures on the same chip. Besides, OTA elements have very high bandwidths compared to Op-Amp elements [20,21]. CCII elements can operate in a wider frequency band than Op-Amp and OTA elements. Also, CCII elements are active circuit elements that have more flexible use in terms of circuit synthesis [22,23].

Another method used for CO design is the structure of Artificial Neural Network (ANN)-based CO design [24–26]. ANN is a mathematical modeling process of the thinking and decision making feature of the human brain. In this method, ANN is first trained by using the training set obtained from the CO [27]. Then ANN model is implemented on a digital platform by taking the weight and bias values obtained from the training as reference [28]. In ANN-based CO designs, many digital sources have been used, and since they contain non-linear transfer functions, their operating frequencies are generally low [28–32].

One of the most preferred methods in the literature is that the COs, which is given in a continuous time, is implemented on a digital platform in discrete time using numerical algorithms [33,34]. For this purpose, Euler, Heun, 4th order Runge-Kutta, Runge-Kutta-Butcher and Dormand-Prince are the numerical algorithms used in the literature [28,35,36]. Among these algorithms, the algorithm that can show the most characteristic of the CSs and produce the closest results is the Dormand-Prince (DP) algorithm [37,38].

In the literature, there are different chaotic oscillator designs implemented using ANN-based and different numerical algorithms on FPGA, which is one of the digital platforms in recent years (Table 1). Alçın et al. [24] implemented the Pehlivan–Uyaroglu chaotic system based on ANN on FPGA and obtained the operating frequency as 266 MHz. They discussed the chip statistics and simulation results of the design and presented them to the literature. Yu et al. [22] designed the 4-D Chua chaotic system in 32-bit IEEE-754-1985 floating point number format with VHDL on FPGA using RK4 numerical algorithm and present the operating frequency of the system as 180 MHz. Rajagopal et al. [33] designed a new 3-D chaotic system on FPGA using Runge-Kutta Butcher (RK5B) numerical algorithm in 32-bit IEEE-754-1985 floating point number format. Koyuncu et al. [37] designed the SEA chaotic oscillator as 16I-16Q, 14I-14Q, 12I-12Q, 10I_10Q, 8I-8Q fixed point based on FPGA using DP numerical algorithm and obtained the operating frequencies of the systems in the range of 344-366 MHz. Seker et al. [36] carried out the DP numerical algorithm-based design of the chaotic oscillator on FPGA in 32-bit IEEE-754-1985 floating point number format and present the operating frequency of the design as 316 MHz. Senouci et al. [39] transformed multiple chaotic systems they designed under Matlab/Simulink into FPGA code using MATLAB HDL Coder and Fixed Point Toolbox and realized their designs on the digital platform. Operating frequencies of systems embedded on FPGAs vary between 27-72 MHz. Kizmaz et al. [40] carried out the designs and chaos analysis of Memristor-Based Simplest Chaotic system based on DP numerical algorithm in their study. Koyuncu et al. [41] performed the Dormand-Prince based chaotic oscillator design with golden proportion balance point on FPGA. In this study, SPCO chaotic system was designed on Xilinx Virtex-7 FPGA using DP numerical algorithm in 32-bit IEEE 754-1985 floating point number format. The operating frequency of the system designed in VHDL was obtained as 362 MHz.

In this presented study, Sundarapandian-Pehlivan CS has been implemented using two different platforms: CCII-based as analog and numerically as DP-based on FPGA chip. In the second part of this study, information about Sundarapandian-Pehlivan CS is given. In the third part, CCII-based SPCO design is presented. In the fourth part, DP-based CO structure and FPGA chip statistics are presented. In the last part, the results obtained from the study were evaluated.

2. SPCO System

CSs or oscillators are divided into two groups, continuous time and discrete time. Examples of continuous-time COs are Sprott A, Burke-Shaw, Lorenz, Chua, Rössler, Pehlivan-Uyaroglu and Pehlivan-Wei COs [39,43,44]. Continuous-time COs are expressed by differential equations. The differential equations of the nonlinear autonomous SPCO used in this study are given below [42].

$$\begin{aligned} \dot{x} &= a.y - x \\ \dot{y} &= -b.x - z \\ \dot{z} &= c.z + x.y^2 - x \end{aligned} \tag{1}$$

The parameters a, b and c in these equations can take different values according to the design. In this study, the ICs for Dormand-Prince-based SPCO are taken as x (0) = 0, y (0) = 0 and z (0) = 0.1 and the parameters of the system are taken as a = 1.5, b = 0.4 and c = 0.4. The change in system parameters and oscillator ICs in COs completely changes the dynamic behavior of the system.

Table 1. Technical characteristics of FPGA-based COs designs in the literature in recent years.

Paper	Method	Number Format	Clock (MHz)
[1]	Euler numerical algor.	Fixed (7;25)	-------
[2]	Heun-RK4 numerical algor.	Floating	390
[3]	RK4 numerical algor.	Fixed (16;16)	373
[13]	RK4 numerical algor.	Fixed (8;24)	112
[22]	RK4 numerical algor.	Floating	180
[24]	ANN-based design	Floating	266
[26]	ANN-based design	Floating	231
[27]	ANN-based design	Floating	240
[28]	ANN-based design	Floating	272
[31]	ANN-based design	Floating	231
[33]	RK5B numerical algor.	Floating	-------
[34]	3-D COs, Forward Euler and RK4	Fixed (8;24)	-------
	4-D Hyper-COs, Forward Euler and RK4	Fixed (8;24)	-------
	3-D COs, Simulink HDL-Coder	Fixed (8;24)	48
	4-D Hyper-COs, Simulink HDL-Coder	Fixed (8;24)	41
[37]	DP numerical algor.	Fixed (16;16)	344-366
[38]	DP numerical algor.	Floating	316
[39]	Simulink HDL-Coder	Fixed	27-72
[40]	DP numerical algor.	Fixed	-----
[41]	DP numerical algor.	Floating	316
[42]	RK4 numerical algor.	Floating	293
This	DP numerical algor.	Floating	362

There are many methods developed in the literature for the chaos analysis of COs. One of the most preferred methods of these methods are chaotic phase portraits and time series methods. Phase portraits of x-y, y-z, x-z and x-y-z obtained using the DP numerical algorithm for the SPCO used in this study are given in Figure 1, Figure 2, Figure 3 and Figure 4, respectively. Besides, the x-y-z time series obtained using the DP numerical algorithm for the SPCO is presented in Figure 5.

3. CCII-based SPCO Design

In this section, SPCO has been modeled using OTA elements and the circuit design of the modeled system has been realized with ORCAD PSPICE program.

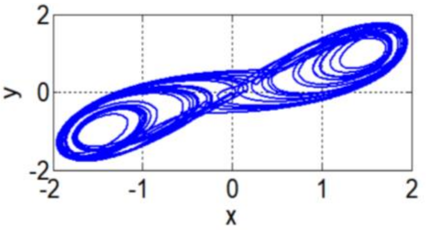


Figure 1. x-y phase portrait of Dormand-Prince-based SPCO

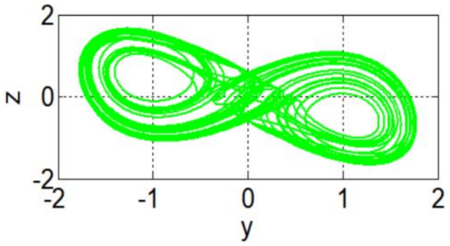


Figure 2. y-z phase portrait of Dormand-Prince-based SPCO

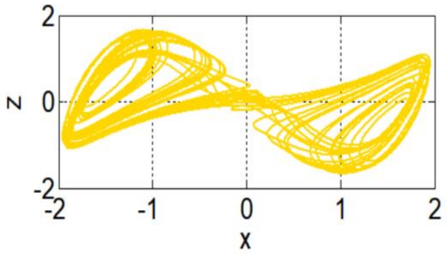


Figure 3. x-z phase portrait of Dormand-Prince-based SPCO

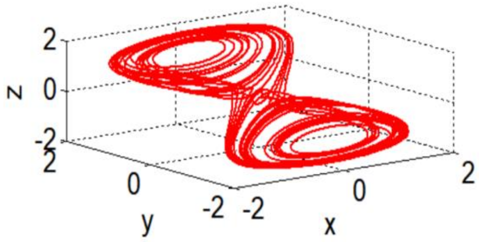


Figure 4. x-y-z phase portrait of Dormand-Prince-based SPCO

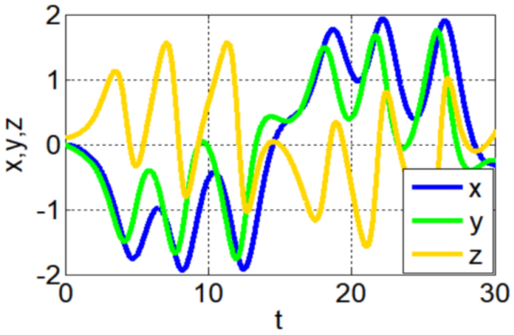


Figure 5. x-y-z time series of Dormand-Prince based SPCO

In the circuit, 14 AD844 CCII, 2 AD633 multiplier, three capacitors and 10 resistance elements with different values have been used. Below is the CO circuit diagram performed with OTA in Figure 6 using the ORCAD PSPICE program.

In Figure 7, x-y, y-z and x-z phase portraits obtained from ORCAD are given, according to the ICs of the CCII-based CO, as x(0) = 0, y(0) = 0 and z(0) = 0.1 and the system parameters as a=1.5, b=0.4 and c=0.4, between the time intervals from 0 to 100ms. In addition, Fast Fourier Transformation (FFT) analysis results of the signals obtained from CCII-based SPCO are presented in Figure 8.

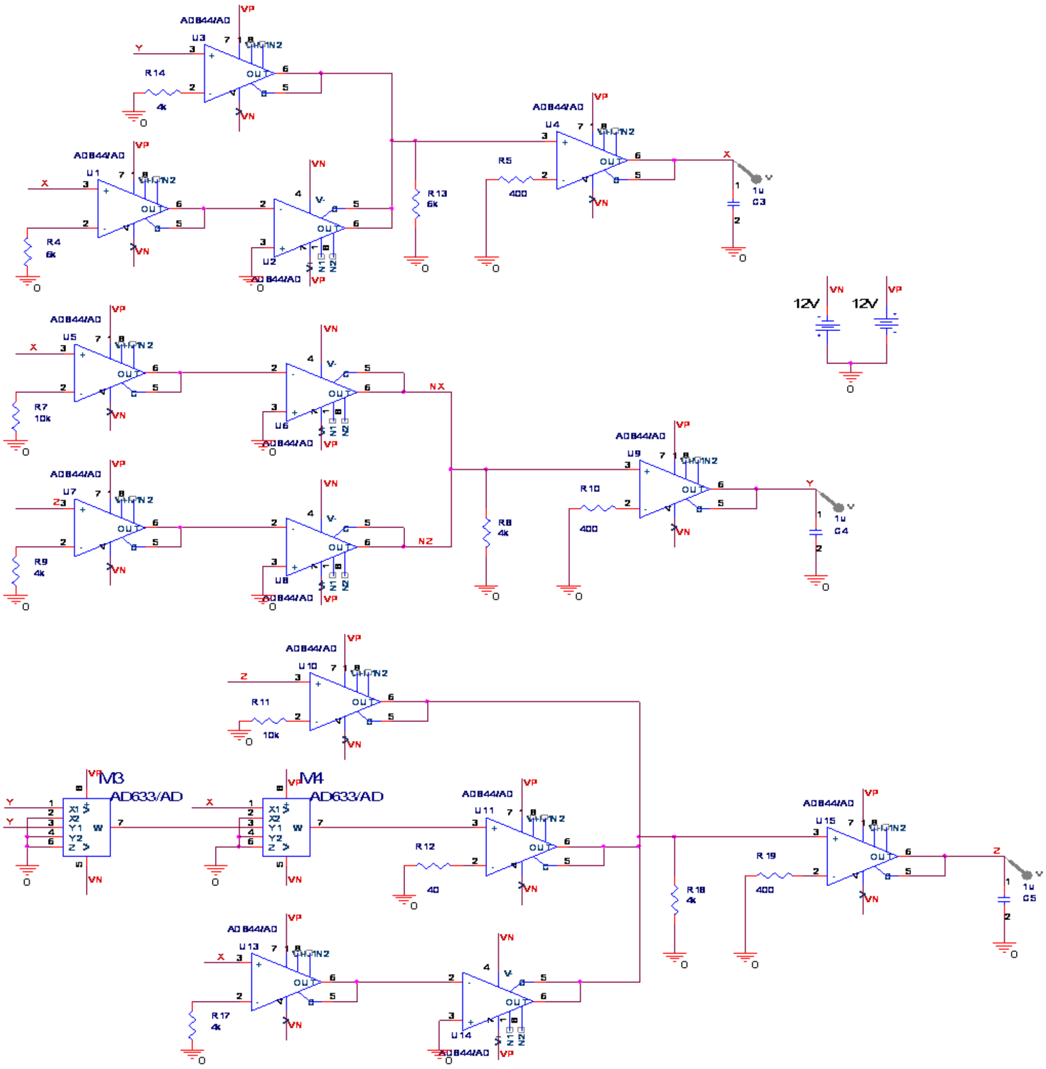
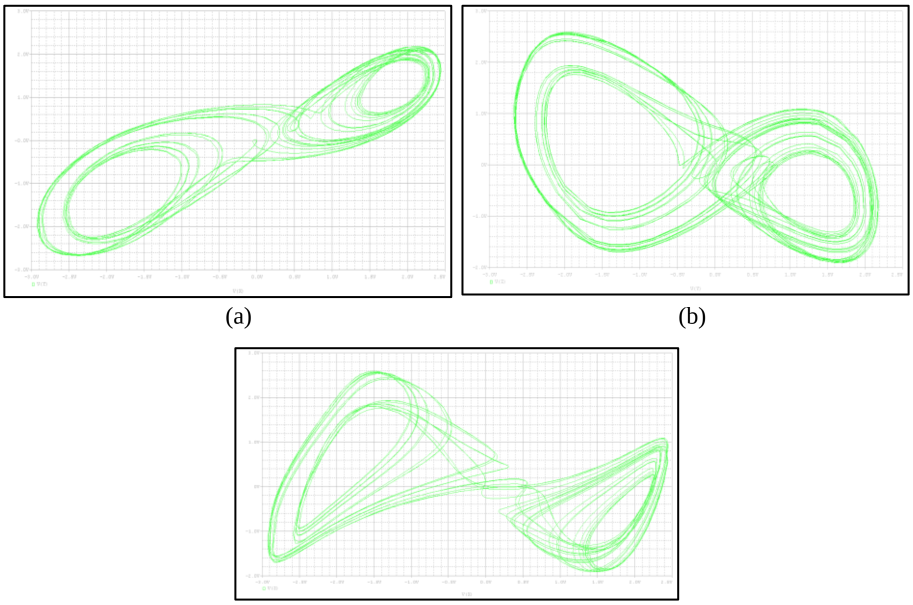


Figure 6. Circuit schema of CCII-based SPCO design using ORCAD-Pspice program


(a)

(b)

(c)

Figure 7. (a) x-y, (b) y-z and (c) x-z phase portraits of CCII-based SPCO designed using ORCAD-Pspice program

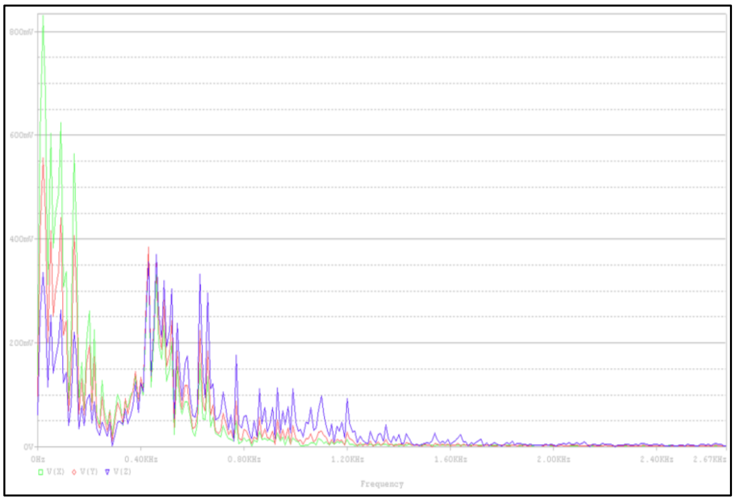


Figure 8. FFT analysis results of the CCII-based SPCO designed using ORCAD-Pspice program

4. Dormand-Prince-based SPCO on FPGA

In this part of the study, the SPCO is modeled to work on FPGA using the DP. DP is given in Equation (2). DP consists of seven steps of k1, k2, k3, k4, k5, k6 and k7. IEEE-754-1985 32-bit floating point representation was used for the design and the design was coded in VHDL. Xilinx ISE Design Tools (ISE-DTs) program was used for the design synthesis, testing and Place-Route operations. Xilinx IP-Core Generator is used for the basic arithmetic operations including division, addition etc. used in the design. The first-order block diagram of DP based SPCO unit designed to work on FPGA chip is given in Figure 9.

$$
\begin{aligned}
& y_{i+1} = y_i + h(\frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6) \\
& k_1 = F(x_i, y_i) \\
& k_2 = F(x_i + \frac{h}{5}, y_i + \frac{h}{5}k_1) \\
& k_3 = F(x_i + \frac{3}{10}h, \left(y_i + \frac{3}{40}k_1 + \frac{9}{40}k_2 \right) * h) \\
& k_4 = F(x_i + \frac{4}{5}h, \left(y_i + \frac{44}{45}k_1 - \frac{56}{15}k_2 + \frac{32}{9}k_3 \right) * h) \\
& k_5 = F(x_i + \frac{8}{9}h, \left(y_i + \frac{19372}{6561}k_1 - \frac{25360}{2187}k_2 + \frac{64448}{6561}k_3 - \frac{212}{729}k_4 \right) * h) \\
& k_6 = F(x_i + h, \left(y_i + \frac{9017}{3168}k_1 - \frac{355}{33}k_2 + \frac{46732}{5247}k_3 + \frac{49}{176}k_4 - \frac{5103}{18656}k_5 \right) * h) \\
& k_7 = F(x_i + h, \left(y_i + \frac{35}{384}k_1 + 0 * k_2 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6 \right) * h)
\end{aligned}
\tag{2}
$$

Dormand-Prince-based SPCO unit on FPGA has 1-bit Clk and 1-bit Run input signals. The design has 32-bit X_out, Y_out, Z_out and 1-bit R_ready output signals. Clk signal allows the units in the design to work synchronously.

Run signal is used to provide the first run. When the '1' signal arrives to the Run input of the design, the X_out, Y_out and Z_out outputs start to produce their initial values as a result of the 360 clock pulse. At this moment, the value of R_Ready becomes '1'. In cases where there is no output in the X_out, Y_out and Z_out, the R_ready signal produces a '0' output.

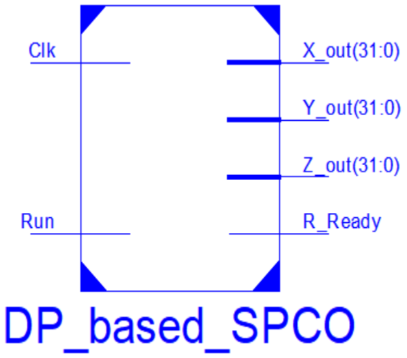


Figure 9. The first-order block diagram of Dormand-Prince based SPCO unit

The second-order block diagram of DP based SPCO unit designed to work on the FPGA chip is given in Figure 10. There are 10 units in the design, 6X3MUX, K1, K2, K3, K4, K5, K6, K7, ys, and Filter. The 6X3MUX unit is designed to choose between the initial values assigned by the designer and the value produced when the system starts producing results. When the design first starts working, it sends the initial values of the 6X3MUX unit to the system.

When the system starts producing the first results, it sends these values to the system as the initial value for the system to produce the next results. The K1, K2, K3, K4, K5, K6 and K7 units are designed to calculate the values found in the DP algorithm. The ys unit calculates the values of the DP algorithm using the values produced by the K1, K2, K3, K4, K5, K6 and K7 units. It then sends these values to the Filter unit. Since the designed system produces results in a 360 clock pulse period, the Filter unit is used to filter the values that come before this moment. When it comes to the 360th clock pulse, it transfers the values received to the output of the designed DP-based SPCO unit. At the same time, the unit sends a '1' signal to the R_Ready output.

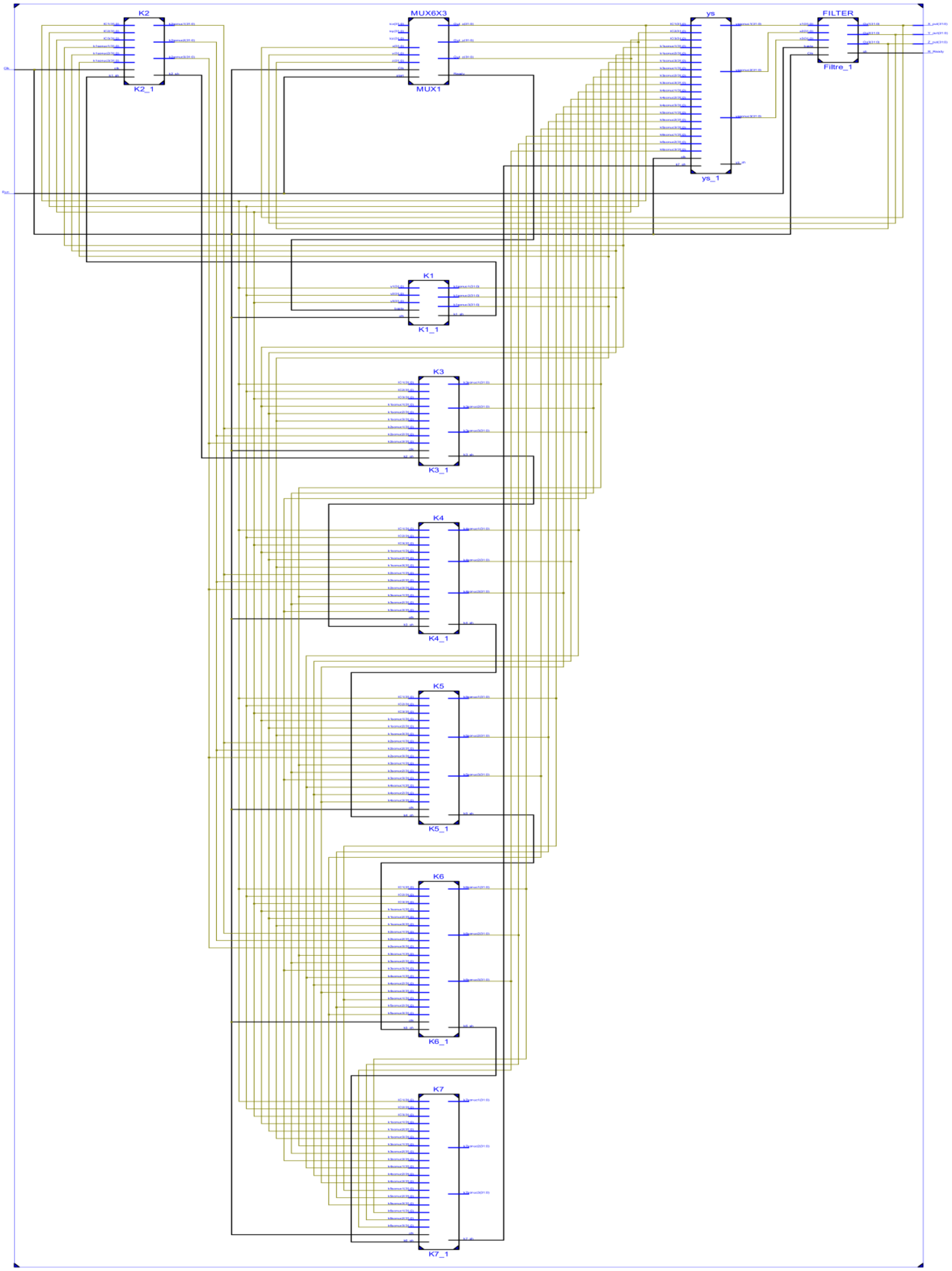


Figure 10. The second-order block diagram of Dormand-Prince based SPCO unit

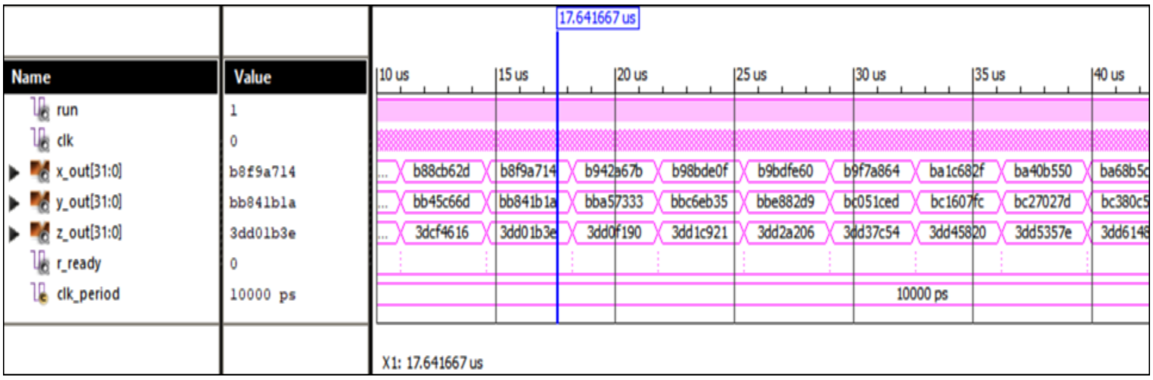


Figure 11. The results of Dormand-Prince based SPCO unit using Xilinx ISE-DTs

Table 2. The usage statistics for VIRTEX-7 chip of DP based SPCO unit

Logic Utilization	Used	Available	Utilization (%)
Number of Slice Registers	154,979	607,200	25
Number of DSP48E1s	150	2,800	5
Number of Slice LUTs	156,371	303,600	51
Number of bonded IOBs	99	700	14

Dormand-Prince based SPCO unit, which was designed to work on FPGA chip, has been tested by using Xilinx ISE-DTs. The test results of DP based SPCO unit using Xilinx ISE-DTs are given in Figure 11. A clock pulse signal and Run input signal generated at 10 ns intervals were sent as '1' to the design. After receiving the Run signal, the presented design produces the first results after 360 clock pulse interval. After this clock pulse, it continues to produce results in every 360 clock pulse. The design works according to the 32-bit IEEE-754-1985 floating point standard. However, in order to evaluate the design results more easily, the results are shown in hexadecimal format.

The Place & Route process was performed for the VIRTEX-7 XC7VX485T-2FFG1761 FPGA chip to be utilized in the designed DP based SPCO unit using Xilinx ISE-DTs program. After the Place+Route process, the obtained FPGA chip resource utilization report is presented in Table 2. The maximum operating frequency of the presented design on the Virtex-7 FPGA chip was obtained as 362.608 MHz. In other words, the minimum clock period of the design is 2.758 ns.

4. Conclusions

In this study, using the SPCO, which is one of the CSs presented to the literature, CO designs have been implemented to be utilized in chaos-based engineering applications on two separate platforms with CCII analog electronic circuit element and FPGA chip. The structure used for the CCII-based CO design and the results of the study are presented. Then SPCO was designed to work on FPGA chips using the DP. The design uses 32-bit floating point representation. The design was coded in VHDL and the designed system was synthesized in Xilinx ISE-DTs program. The designed system has been tested by preparing a test bench in VHDL. In the study, test results obtained from DP based SPCO structure are given. The SPCO unit designed in the last part of the study was synthesized for the VIRTEX-7 FPGA. Source usage values of FPGA chip obtained after the Place+Route of the unit are presented in the study. According to the obtained results, the maximum operating frequency of DP-based SPCO unit on FPGA is approximately 362 MHz. The chaotic signal generator is one of the basic structures used in chaos-based studies and applications. In this study, SPCO designs have been successfully implemented in 2 different platforms as analog-based using CCII and digital-based using FPGA chip. Synchronization, secure communication and Pseudo/True RNG applications can be implemented with DP based SPCO unit proposed in this study in future studies.

Declaration

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article. The author(s) also declared that this article is original, was prepared in accordance with international publication and research ethics, and ethical committee permission or any special permission is not required.

References

1. Chang, D., Li, Z., Wang, M., and Y. Zeng, *A Novel Digital Programmable Multi-Scroll Chaotic System and Its Application in FPGA-Based Audio Secure Communication*, AEU-International Journal of Electronics and Communications, 2018. **88**(2018): p. 20–29.
2. Tuna, M., and C.B. Fidan, *Electronic Circuit Design, Implementation and FPGA-Based Realization of a New 3D Chaotic System with Single Equilibrium Point*, Optik - International Journal for Light and Electron Optics, 2016. **127**(24): p. 11786–11799.

3. Alçın, M., Tuna, M., and İ. Koyuncu, I*Q-Math Based Designing of Fourth Order Runge-Kutta Algorithm on FPGA and Performance Analysis According to ANN Approximation*, International Journal of Advanced Research in Science, Engineering and Technology, 2018. **5**(8): p. 6523–6530.

4. Tuncer, T., *The Implementation of Chaos-Based PUF Designs in Field Programmable Gate Array*, Nonlinear Dynamics, 2016. **86**(2): p. 975–986.

5. Id, A., Shaukat Id, S., Id, A.A., Id, A.E., Aziz, S., Id, S., and J. Ahmad Id, *Chaos Theory and Its Application: An Essential Framework for Image Encryption*, Chaos Theory and Applications, 2020. **2**(1): p. 17–22.

6. Tuna, M., and C.B. Fidan, *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, 2018. **33**(2): p. 469–486.

7. Bonny, T., and Q. Nasir, *Clock Glitch Fault Injection Attack on an FPGA-Based Non-Autonomous Chaotic Oscillator*, Nonlinear Dynamics, 2019. **96**(3): p. 2087–2101.

8. Avaroğlu, E., Tuncer, T., Özer, A.B., Ergen, B., and M. Türk, *A Novel Chaos-Based Post-Processing for TRNG*, Nonlinear Dynamics, 2015. **81**(1–2): p. 189–199.

9. Akgul, A., Moroz, I., Pehlivan, I., and S. Vaidyanathan, *A New Four-Scroll Chaotic Attractor and Its Engineering Applications*, Optik - International Journal for Light and Electron Optics, 2016. **127**(13): p. 5491–5499.

10. Avaroğlu, E., *Pseudorandom Number Generator Based on Arnold Cat Map and Statistical Analysis*, Turkısh Journal of Electrıcal Engıneerıng & Computer Scıences, 2017. **25**(1): p. 633–643.

11. Garipcan, A.M., and E. Erdem, *A TRNG Using Chaotic Entropy Pool as a Post-Processing Technique: Analysis, Design and FPGA Implementation*, Analog Integrated Circuits and Signal Processing, 2020. **103**(2020): p. 391–410.

12. Akgül, A., Zahid Yıldız, M., Faruk Boyraz, Ö., Güleryüz, E., Kaçar, S., and B. Gürevin, *Microcomputer-Based Encryption of Vein Images with a Non-Linear Novel System*, Journal of the Faculty of Engineering and Architecture of Gazi University, 2020. **35**(3): p. 1369–1385.

13. Bonny, T., Al Debsi, R., Majzoub, S., and A.S. Elwakil, *Hardware Optimized FPGA Implementations of High-Speed True Random Bit Generators Based on Switching-Type Chaotic Oscillators*, Circuits, Systems, and Signal Processing, 2019. **38**(3): p. 1342–1359.

14. Çam Taşkıran, Z.G., and H. Sedef, *Memristör Tabanlı Kaotik Rössler Devresi Gerçeklemesi*, Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 2019. **35**(2): p. 765–774.

15. Sundarapandian, V., and I. Pehlivan, *Analysis, Control, Synchronization, and Circuit Design of a Novel Chaotic System*, Mathematical and Computer Modelling, 2012. **55**(7–8): p. 1904–1915.

16. Akgul, A., Hussain, S., and I. Pehlivan, *A New Three-Dimensional Chaotic System, Its Dynamical Analysis and Electronic Circuit Applications*, Optik-International Journal for Light and Electron Optics, 2016. **127**(18): p. 7062–7071.

17. Lin, Y., Wang, C., and H. He, *A Simple Multi-Scroll Chaotic Oscillator Employing CCIIs*, Optik-International Journal for Light and Electron Optics, 2015. **126**(7–8): p. 824–827.

18. Munoz-Pach, J.M., Campos-Lop, W., Tlelo-Cuau, E., and C. Sanchez-Lo, *OpAmp-, CFOA- and OTA-Based Configurations to Design Multi-Scroll Chaotic Oscillators*, Trends in Applied Sciences Research, 2012. **7**(2): p. 168–174.

19. Bulut, G.G., Emin Şahin, M., and H. Güler, *An Implementation of Chaotic Circuits with Multisim-LabVIEW*, International Advanced Researches and Engineering Journal, 2018. **02**(03): p. 304–308.

20. Dar, M.R., Kant, N.A., and F.A. Khanday, *Realization of Fractional-Order Double-Scroll Chaotic System Using Operational Transconductance Amplifier (OTA)*, Journal of Circuits, Systems and Computers, 2018. **27**(1): p. 1850006.

21. Singh, J.P., Koley, J., Akgul, A., Gurevin, B., and B.K. Roy, *A New Chaotic Oscillator Containing Generalised Memristor, Single Op-Amp and RLC with Chaos Suppression and an Application for the Random Number Generation*, European Physical Journal: Special Topics, 2019. **228**(10): p. 2233–2245.

22. Yu, F., Shen, H., Liu, L., Zhang, Z., Huang, Y., He, B., Cai, S., Song, Y., Yin, B., Du, S., and Q, Xu, *CCII and FPGA Realization: A Multistable Modified Fourth-Order Autonomous Chua's Chaotic System with Coexisting Multiple Attractors*, Complexity, 2020. **5212601**: p. 1–17.

23. Hasan, S., and I.A. Khan, *Multi-Scroll Chaos Generator Using Current Conveyors*, 2009 International Multimedia, Signal Processing and Communication Technologies, 2009. Aligarh, India. p. 229–232.

24. Alçın, M., Pehlivan, İ., and İ. Koyuncu, *Hardware Design and Implementation of a Novel ANN-Based Chaotic Generator in FPGA*, Optik - International Journal for Light and Electron Optics, 2016. **127**(13): p. 5500–5505.

25. Alçin, M., *The Modelling Performance Evolution of Recurrent Neural Networks Having Different Training Functions for Modelling Sprott H Chaotic System*, International Journal on Research Innovations in Engineering Science and Technology(IJRIEST), 2017. **2**(10): p. 563–568.

26. Koyuncu, İ., Oğuz, Y., Çimen, H., Özer, T., 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, 2018. Skopje, Macedonia. p. 1–6.

27. Alcin, M., Erdogmus, P., 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, 2020. **8**(2): p. 102–108.

28. Vaidyanathan, S., Pehlivan, I., Dolvis, L.G., Jacques, K., Alcin, M., Tuna, M., and I. Koyuncu, *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, 2020. **62**(1): p. 20–35.

29. Koyuncu, İ., Şahin, İ., Gloster, C., and N.K. Sarıtekin, *A Neuron Library for Rapid Realization of Artificial Neural Networks on FPGA: A Case Study of Rössler Chaotic System*, Journal of Circuits, Systems and Computers, 2017. **26**(01): p. 1750015.

30. Dalkiran, I., and K. Danis, *Artificial Neural Network*

Based Chaotic Generator for Cryptology, Turkish Journal of Electrical Engineering & Computer Sciences, 2010. **18**(2): p. 225–240.

31. Alcin, M., Koyuncu, I., Tuna, M., Varan, M., 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, 2019. **47**(3): p. 365–378.
32. Tuna, M., Karthikeyan, A., Rajagopal, K., Alçın, M., and İ. Koyuncu, *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, 2019. **112**(2019): p. 152941–10.
33. Rajagopal, K., Akgul, A., Jafari, S., Karthikeyan, A., and I. Koyuncu, *Chaotic Chameleon: Dynamic Analyses, Circuit Implementation, FPGA Design and Fractional-Order Form with Basic Analyses*, Chaos, Solitons & Fractals, 2017. **103**(2017): p. 476–487.
34. Bonny, T., *Chaotic or Hyper-Chaotic Oscillator? Numerical Solution, Circuit Design, MATLAB HDL-Coder Implementation, VHDL Code, Security Analysis, and FPGA Realization*, Circuits, Systems, and Signal Processing, 2020. p. 1–28.
35. Dursun, M., and E. Kaşifoğlu, *Design and Implementation of the FPGA-Based Chaotic van Der Pol Oscillator*, International Advanced Researches and Engineering Journal, 2018. **02**(03): p. 309–314.
36. Karakaya, B., Turk, M.A., Turk, M., and A. Gulten, *Selection of Optimal Numerical Method for Implementation of Lorenz Chaotic System on FPGA*, International Advanced Researches and Engineering Journal, 2018. **02**(02): p. 147–152.
37. Koyuncu, İ., and H.İ. Seker, *Implementation of Dormand-Prince Based Chaotic Oscillator Designs in Different IQ-Math Number Standards on FPGA*, Sakarya University Journal of Science, 2019. **23**(5): p. 859–868.
38. Şeker, H.İ., Koyuncu, İ., Tuna, M., and M. Alçın, *Implementation of Dormand Prince Based SEA Chaotic Oscillator Design on FPGA*, V. Science Technology and Innovation Congress, 2019. Alanya, Turkey. p. 303–310.
39. Senouci, A., Bouhedjeur, H., Tourche, K., and A. Boukabou, *FPGA Based Hardware and Device-Independent Implementation of Chaotic Generators*, AEU-International Journal of Electronics and Communications, 2017. **82**(2017): p. 211–220.
40. Kizmaz, H., Kocamaz, U.E., and Y. Uyaroğlu, *Control of Memristor-Based Simplest Chaotic Circuit with One-State Controllers*, Journal of Circuits, Systems, and Computers, 2019. **28**(1): p. 1950007.
41. Koyuncu, İ., Şeker, H.İ., Tuna, M., and M. Alçın, *Dormand-Prince Tabanlı Kaotik Osilatör Tasarımının FPGA Üzerinde Gerçeklenmesi*, International Eurasian Conference on Science, Engineering and Technology, 2018. Ankara, Turkey. p. 1059–1065.
42. Koyuncu, İ., and A.T. Özcerit, *The Design and Realization of a New High Speed FPGA-Based Chaotic True Random Number Generator*, Computers & Electrical Engineering, 2017. **58**(2017): p. 203–214.
43. Pehlivan, I., and Y. Uyaroglu, *A New Chaotic Attractor from General Lorenz System Family and Its Electronic Experimental Implementation*, Turkish Journal of Electrical Engineering & Computer Sciences, 2010. **18**(2): p. 171–184.
44. Pehlivan, İ., and Z. Wei, *Analysis, Nonlinear Control, and Chaos Generator Circuit of Another Strange Chaotic System*, Turkısh Journal of Electrıcal Engıneerıng & Computer Scıences, 2012. **20**(2): p. 1229–1239.