CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.3 / No.1 / 2021 / pp.20-28
https://doi.org/10.51537/chaos.783548

FPGA-based Dual Core TRNG Design Using Ring and Runge-Kutta-Butcher based on Chaotic Oscillator

Murat Alcin [*,1], **Murat Tuna** [β,2], **Pakize Erdogmus** [γ,3] **and Ismail Koyuncu** [§,4]
[*] Department of Mechatronics Engineering, Faculty of Technology, Afyon Kocatepe University, Afyon, 03200, Turkey, [β] Department of Electrical, Technical Sciences Vocational School, Kırklareli University, Kırklareli, 39000, Turkey, [γ] Department of Computer Engineering, Faculty of Engineering, Düzce University, Düzce, 81620, Turkey, [§] Department of Electrical Electronics Engineering, Faculty of Technology, Afyon Kocatepe Uni., Afyon, 03200, Turkey.

ABSTRACT Despite the fact that chaotic systems do not have very complex circuit structures, interest in chaotic systems has increased considerably in recent years due to their interesting dynamic properties. Thanks to the noise-like properties of chaotic oscillators and the ability to mask information signals, great efforts have been made in recent years to develop chaos-based TRNG structures. In this study, a new chaos-based Dual Entropy Core (DEC) TRNG with high operating frequency and high bit generation rate was realized using 3D Pehlivan-Wei Chaotic Oscillator (PWCO) structure designed utilizing RK5-Butcher numerical algorithm on FPGA and ring oscillator structure. In the FPGA-based TRNG model of the system, 32-bit IQ-Math fixed-point number standard is used. The developed model is coded using VHDL. The designed TRNG unit was synthesized for Virtex-7 XC7VX485T-2FFG1761 chip produced by Xilinx. Then, the statistics of the parameters of FPGA chip resource usage and unit clock speed were examined. The data processing time of the TRNG unit was achieved by using the Xilinx ISE Design Tools 14.2 simulation program, with a high bit production rate of 437.043 Mbit/s. In addition, number sequences obtained from FPGA-based TRNG were subjected to the internationally valid statistical NIST 800-22 Test Suite and all the randomness tests of NIST 800-22 Test Suite were successful.

KEYWORDS
Chaotic systems
Ring oscillator
RK5-Butcher
Dual core TRNG
FPGA

INTRODUCTION

The term chaos is used to describe the dynamic behavior of simple dynamical systems, which appears to be complex and very different from what was predicted (Akgul *et al.* 2016b; Tuna and Fidan 2018). The behavior of these systems has a non-periodic property and can easily be confused with random behavior (Akkaya *et al.* 2018; Rivera-Blas *et al.* 2019). Chaotic systems are sensitive to initial conditions, complex and irregular in appearance, and occur in deterministic nonlinear time-dependent systems (Dursun and Kaşifoğlu 2018; Tuna *et al.* 2019a; Bonny and Elwakil 2018). Although chaotic systems do not have very complex circuit structures, since they have interesting dynamical properties, the interest in chaotic systems has been increased in recent years (Alçın *et al.* 2016; Koyuncu *et al.* 2019; Öztürk and Kiliç 2014). The basic structure to be used in chaos-based engineering applications is a chaos generator that produces the necessary chaotic signal (Adiyaman *et al.* 2020; Akgul *et al.* 2016a; Li *et al.* 2005). Thus, secure communication, cryptographic and random number generators, in which chaotic signals are used as entropy sources, have been proposed (Taskiran and Sedef 2020; Akgul *et al.* 2019; Benkouider *et al.* 2020; Bonny *et al.* 2019).

Ring oscillators are the oscillators consisting of an odd number of NOT gates connected cascade (Koyuncu *et al.* 2020). The output of each gate is connected to the input of the next gate, and the output of the last gate is connected to the input of the first gate. Ring oscillators generate a square

Manuscript received: 22 August 2020,
Revised: 25 December 2020,
Accepted: 27 January 2021.

[1] muratalcin@aku.edu.tr (**Corresponding Author**)
[2] murattuna@klu.edu.tr
[3] pakizeerdogmus@duzce.edu.tr
[4] ismailkoyuncu@aku.edu.tr

wave having a frequency depending on the delay of the ring (Koyuncu *et al.* 2020; Tuncer 2016). Therefore, the frequency of the obtained square wave varies according to the static and dynamic factors in the elements forming the ring. That is, the frequencies of the signals produced by two equally arranged oscillators will not be the same. This shows that ring oscillators can be used to generate random bits that differ in the frequencies of the signals they produce (Tuna *et al.* 2019b). Most of the integrated circuit (I.C.) applications and Field Programmable Gate Array (FPGA) based True Random Number Generators (TRNG) use the ring oscillator structures as the source of randomness (Kaya 2020; Buchovecka *et al.* 2017; Garipcan and Erdem 2019; Yoo *et al.* 2010; Avaroglu and Tuncer 2020; Bonny and Nasir 2019).

Systems that do not have autocorrelation at their output using hardware or software methods and produce numbers that are statistically independent from each other are called Random Number Generators (RNG) (Coskun *et al.* 2019; Gupta *et al.* 2019; Prakash *et al.* 2020). These generators are structures that can generate outputs at the level of randomness, where the next data cannot be predicted with the help of previous data. Because of these features, RNG is used in many different areas. TRNG is a device that produces a sequence of numbers such that they cannot be predicted. The random numbers produced by TRNG is a safe method since it is difficult to generate the same numbers. For this reason, considerable efforts are being made in the field of developing hardware-based random number generation structures with FPGA and general purpose microprocessors (Koyuncu and Özcerit 2017; Öztürk and Kılıç 2019). FPGA is a programmable integrated circuit (IC) whose internal structure can be changed any number of time with respect to desired function (Koyuncu and Özcerit 2017; Alcin 2020). So, FPGA is used for rapid prototype development. FPGA is commonly used nowadays because it presents great flexibility in the design stage, and it has parallel processing capability. The advantages of faster implementation and having higher density, make FPGAs possible to implement complex systems including numerical calculations. Programmable FPGA chips have an important potential to improve information security capacity in applications such as cryptology and secure communication, which require high performance and processing power, due to their high speed and capacity (Hagras and Saber 2020; Alcin *et al.* 2019; Koyuncu and Şeker 2019).

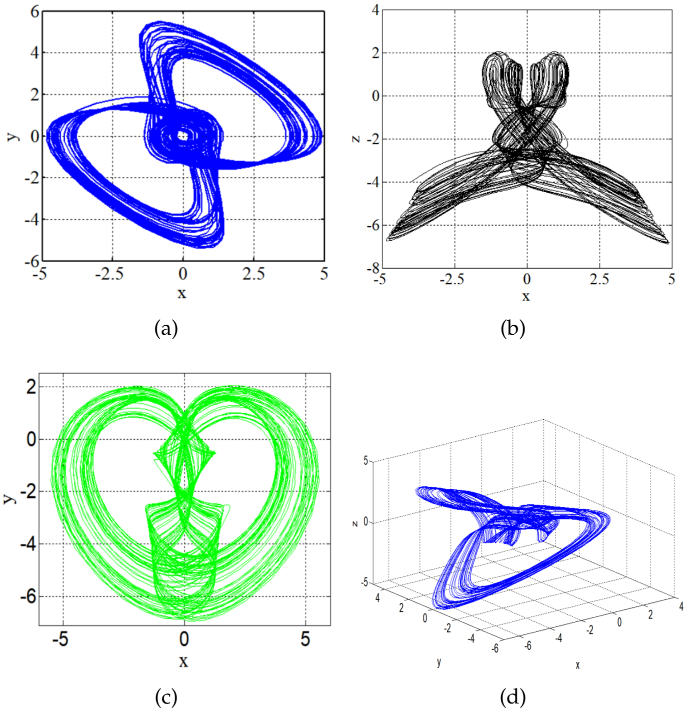
In the second part of the study, two and three dimensional phase portraits obtained from the modeling of the 3D PWCO system, one of the chaotic oscillators presented to the literature, using Runge-Kutta-Butcher algorithm (RK5-Butcher) are presented. In the third chapter, Dual Entropy Core (DEC) TRNG design using Ring and RK5-Butcher based PWCO on FPGA and the results obtained from the design are given. In the last part, the results obtained from the study are discussed.

THE 3D PWCO SYSTEM

Chaotic systems are expressed using differential equations. The differential equation for the continuous-time 3D PWCO system is given in Eq. (1) (Koyuncu *et al.* 2014).

$$\begin{aligned}\dot{x} &= y(1-z)\\ \dot{y} &= y(1+z)-\alpha x\\ \dot{z} &= \alpha - xy - y^2\end{aligned} \tag{1}$$

Here α is the system parameter for PWCO. The change of this value greatly changes the dynamic behavior of the system. In this study, α has been set to 2.1 for the PWCO modeled using the RK5-Butcher algorithm. Initial conditions are needed for the system to work. In this study, the initial conditions for PWCO modeled by using the RK5-Butcher algorithm are taken as $x(0) = -3.9, y(0) = 0.90$, and $z(0) = -4.1$. Two-dimensional $x-y, x-z, y-z$ and three-dimensional $x-y-z$ phase portraits for the PWCO oscillator modeled using the RK5-Butcher algorithm are presented in Figure 1.



(a)

(b)

(c)

(d)

Figure 1 2D a) x-y, b) x-z, c) y-z and d) x-y-z phase portraits of RK5-Butcher based 3D PWCO

PWCO AND RING BASED DEC TRNG ON FPGA

In this section, the DEC TRNG design, which is implemented utilizing PWCO oscillator that created using Fifth Order Runge-Kutta Butcher Algorithm (RK5-B) numerical algorithm and Ring oscillator on FPGA, has been implemented. The discretized mathematical model of PWCO using the RK5-Butcher algorithm is given in Eq. 2. Here, the expansion of variables $\kappa_1 \ldots \kappa_6$, $\lambda_1 \ldots \lambda_1$ and $\xi_1 \ldots \xi_1$ is given in Eq. 3. Although the RK5-B has a similar structure with the RK4 (Fourth Order Runge-Kutta Algorithm) algorithm, this algorithm also produces more precise solutions than the RK4 and Euler algorithms since it has fifth and sixth order terms (Tlelo-Cuautle *et al.* 2015; Pano-Azucena *et al.* 2018; Sambas *et al.* 2020).

$$
\begin{aligned}
&x(k+1) = x(k) + \tfrac{1}{90}\Delta h \left[7\kappa_1(k) + 32\kappa_3(k) + 12\kappa_4(k) + 32\kappa_5(k) + 7\kappa_6(k)\right] \\
&y(k+1) = y(k) + \tfrac{1}{90}\Delta h \left[7\lambda_1(k) + 32\lambda_3(k) + 12\lambda_4(k) + 32\lambda_5(k) + 7\lambda_6(k)\right] \\
&z(k+1) = z(k) + \tfrac{1}{90}\Delta h \left[7\zeta_1(k) + 32\zeta_3(k) + 12\zeta_4(k) + 32\zeta_5(k) + 7\zeta_6(k)\right]
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
&\kappa_1 = f(x(k), y(k), z(k)) \\
&\lambda_1 = g(x(k), y(k), z(k)) \\
&\xi_1 = \delta(x(k), y(k), z(k)) \\
&\kappa_2 = f(x(k) + \tfrac{1}{4}\Delta h\kappa_1, y(k) + \tfrac{1}{4}\Delta h\lambda_1, z(k) + \tfrac{1}{4}\Delta h\xi_1) \\
&\lambda_2 = g(x(k) + \tfrac{1}{4}\Delta h\kappa_1, y(k) + \tfrac{1}{4}\Delta h\lambda_1, z(k) + \tfrac{1}{4}\Delta h\xi_1) \\
&\xi_2 = \delta(x(k) + \tfrac{1}{4}\Delta h\kappa_1, y(k) + \tfrac{1}{4}\Delta h\lambda_1, z(k) + \tfrac{1}{4}\Delta h\xi_1) \\
&\kappa_3 = f(x(k) + \tfrac{1}{8}(\Delta h(\kappa_1 + \kappa_{2)}, y(k) + \tfrac{1}{8}(\Delta h(\lambda_1 + \lambda_{2)}, z(k) + \tfrac{1}{8}(\Delta h(\xi_1 + \xi_{2)}) \\
&\lambda_3 = g(x(k) + \tfrac{1}{8}(\Delta h(\kappa_1 + \kappa_{2)}, y(k) + \tfrac{1}{8}(\Delta h(\lambda_1 + \lambda_{2)}, z(k) + \tfrac{1}{8}(\Delta h(\xi_1 + \xi_{2)}) \\
&\xi_3 = \delta(x(k) + \tfrac{1}{8}(\Delta h(\kappa_1 + \kappa_{2)}, y(k) + \tfrac{1}{8}(\Delta h(\lambda_1 + \lambda_{2)}, z(k) + \tfrac{1}{8}(\Delta h(\xi_1 + \xi_{2)}) \\
&\kappa_4 = f(x(k) - \tfrac{1}{2}\Delta h\kappa_2 + \Delta h\kappa_3, y(k) - \tfrac{1}{2}\Delta h\lambda_2 + \Delta h\lambda_3, z(k) - \tfrac{1}{2}\Delta h\xi_2 + \Delta h\xi_3) \\
&\lambda_4 = g(x(k) - \tfrac{1}{2}\Delta h\kappa_2 + \Delta h\kappa_3, y(k) - \tfrac{1}{2}\Delta h\lambda_2 + \Delta h\lambda_3, z(k) - \tfrac{1}{2}\Delta h\xi_2 + \Delta h\xi_3) \\
&\xi_4 = \delta(x(k) - \tfrac{1}{2}\Delta h\kappa_2 + \Delta h\kappa_3, y(k) - \tfrac{1}{2}\Delta h\lambda_2 + \Delta h\lambda_3, z(k) - \tfrac{1}{2}\Delta h\xi_2 + \Delta h\xi_3) \\
&\kappa_5 = f(x(k) + \tfrac{3}{16}\Delta h\kappa_1 + \tfrac{9}{16}\Delta h\kappa_4, y(k) + \tfrac{3}{16}\Delta h\lambda_1 + \tfrac{9}{16}\Delta h\lambda_4, z(k) + \tfrac{3}{16}\Delta h\xi_1 + \tfrac{9}{16}\Delta h\xi_4) \\
&\lambda_5 = g(x(k) + \tfrac{3}{16}\Delta h\kappa_1 + \tfrac{9}{16}\Delta h\kappa_4, y(k) + \tfrac{3}{16}\Delta h\lambda_1 + \tfrac{9}{16}\Delta h\lambda_4, z(k) + \tfrac{3}{16}\Delta h\xi_1 + \tfrac{9}{16}\Delta h\xi_4) \\
&\xi_5 = \delta(x(k) + \tfrac{3}{16}\Delta h\kappa_1 + \tfrac{9}{16}\Delta h\kappa_4, y(k) + \tfrac{3}{16}\Delta h\lambda_1 + \tfrac{9}{16}\Delta h\lambda_4, z(k) + \tfrac{3}{16}\Delta h\xi_1 + \tfrac{9}{16}\Delta h\xi_4) \\
&\kappa 6 = f(x(k) - \tfrac{3}{7}\Delta h\kappa_1 + \tfrac{2}{7}\Delta h\kappa_2 + \tfrac{12}{7}\Delta h\kappa_3 - \tfrac{12}{7}\Delta h\kappa_4 + \tfrac{8}{7}\Delta h\kappa_5, y(k) + -\tfrac{3}{7}\Delta h\lambda_1 + \tfrac{2}{7}\Delta h\lambda_2 + \\
&\tfrac{12}{7}\Delta h\lambda_3 - \tfrac{12}{7}\Delta h\lambda_4 + \tfrac{8}{7}\Delta h\lambda_5, z(k) - \tfrac{3}{7}\Delta h\xi_1 + \tfrac{2}{7}\Delta h\xi_2 + \tfrac{12}{7}\Delta h\xi_3 - \tfrac{12}{7}\Delta h\xi_4 + \tfrac{8}{7}\Delta h\xi_5) \\
&\lambda 6 = g(x(k) - \tfrac{3}{7}\Delta h\kappa_1 + \tfrac{2}{7}\Delta h\kappa_2 + \tfrac{12}{7}\Delta h\kappa_3 - \tfrac{12}{7}\Delta h\kappa_4 + \tfrac{8}{7}\Delta h\kappa_5, y(k) + -\tfrac{3}{7}\Delta h\lambda_1 + \tfrac{2}{7}\Delta h\lambda_2 + \\
&\tfrac{12}{7}\Delta h\lambda_3 - \tfrac{12}{7}\Delta h\lambda_4 + \tfrac{8}{7}\Delta h\lambda_5, z(k) - \tfrac{3}{7}\Delta h\xi_1 + \tfrac{2}{7}\Delta h\xi_2 + \tfrac{12}{7}\Delta h\xi_3 - \tfrac{12}{7}\Delta h\xi_4 + \tfrac{8}{7}\Delta h\xi_5) \\
&\xi 6 = \delta(x(k) - \tfrac{3}{7}\Delta h\kappa_1 + \tfrac{2}{7}\Delta h\kappa_2 + \tfrac{12}{7}\Delta h\kappa_3 - \tfrac{12}{7}\Delta h\kappa_4 + \tfrac{8}{7}\Delta h\kappa_5, y(k) + -\tfrac{3}{7}\Delta h\lambda_1 + \tfrac{2}{7}\Delta h\lambda_2 + \\
&\tfrac{12}{7}\Delta h\lambda_3 - \tfrac{12}{7}\Delta h\lambda_4 + \tfrac{8}{7}\Delta h\lambda_5, z(k) - \tfrac{3}{7}\Delta h\xi_1 + \tfrac{2}{7}\Delta h\xi_2 + \tfrac{12}{7}\Delta h\xi_3 - \tfrac{12}{7}\Delta h\xi_4 + \tfrac{8}{7}\Delta h\xi_5)
\end{aligned}
\tag{3}
$$

The top level block diagram of the designed structure is given in Figure 2. Random number sequences with high throughput and high operating frequency obtained from the proposed structure; they can be used in cryptography and secure communication areas that require fast, secure and intensive processing. The designed chaotic DEC TRNG unit was synthesized for the Virtex-7 VC707 chip produced by Xilinx, and the statistics of the parameters of FPGA chip resource usage and the clock speeds of the units were analyzed. The data processing time of TRNG units was obtained using Xilinx ISE Design Tools 14.2 simulation program.

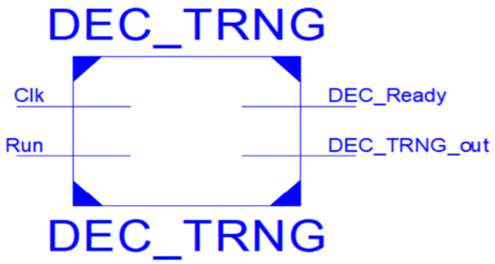


Figure 2 The top-level block diagram of the FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO.

Figure 3 shows the block diagram of the proposed FPGA-based DEC TRNG unit. RK5-Butcher-based TRNG unit designed on FPGA consists of 5 parts: $x3mux$, $PWKS_RK5$ oscillator, Quantization unit, Ring oscillator and Art unit. In the design, the $x3mux$ unit is basically a multiplexer (MUX) structure developed for the control of the start signals required by the $PWKS_RK5$ unit, which has 3 dependent variables.

Quantization process was realized by taking the last 23 bits of the fractional part of each 32-bit number in the fixed point number standard produced by the fixed point number based chaotic oscillator unit. The RN signals obtained from the output of this unit are the signals that carry random numbers. The sh signal indicates that random signals are received from the unit output. These two signals are transmitted to the ART unit. Post processing is applied for the signals obtained here.

In the presented study, XOR process was applied as the post process and the results obtained were sent to the output of the system. The random numbers produced by the ring oscillator and the random numbers produced by the RK5-Butcher algorithm based PWCO-based TRNG unit are subjected to XOR processing in the $ART - PROCESSING$ unit.

In TRNG structures subjected to XOR process presented in the literature, as a result of the XOR process, the bit production rate is reduced by half. However, unlike the studies presented in the literature, in the XOR process presented in this study, since random numbers are generated from two different sources and subjected to the XOR process, there is no decrease in the bit production rate in the high speed DEC TRNG using Ring and RK5-Butcher algorithm based 3D PWCO on FPGA design.

In Fig. 4, the third level block diagram of high speed DEC TRNG using Ring and RK5-Butcher based on PWCO on FPGA is presented.

Here, the structure of the $PWKS_RK5$ unit is given in more detail. The $PWKS_RK5$ oscillator generates the chaotic signals that TRNG needs and transfers these values to 32-bit x_out, y_out and z_out signals. When the chaotic oscillator produces an output, the 1-bit RNG_Ready signal becomes ”1” and sends the values produced by the 3D PWCO to the Quantization unit. All units used in these designs such as multiplier, adder, and subtractor were created using the IP Core generator developed with Xilinx ISE Design Tools.

FPGA-based DEC TRNG Design using Ring and RK5-Butcher based on PWCO unit is synthesized and tested for Xilinx Virtex-7 XC7VX485T-2FFG1761 FPGA chip. Figure 5. presents the test bench results for the FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO unit, whose code was written in VHDL.

FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO unit has been synthesized and then after the Place-Route processes, XC7VX330T-2-FFG-1157 FPGA chip statistics have been obtained. As can be observed from the chip statistics in the Table 1, the maximum clock frequency of the FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO unit reaches 437.043 MHz.

As can be observed from the literature, it is necessary to examine and to test the randomness and statistical properties of the random numbers produced by random number generators (Rezk *et al.* 2019; Murillo-Escobar *et al.* 2017). For this purpose, various statistical tests developed in the literature are used. At this stage, the new chaotic DEC TRNG developed on FPGA has been subjected to the NIST 800-22 statistical tests in order to be used safely in cryptographic applications (Etem and Kaya 2020). This test itself consists of 16 separate subtests. In order for the tested bit stream to be accepted as successful, it must pass all tests successfully. The orders in which the 16 tests in the NIST 800-22 test are run is completely optional. However, the Frequency Test is recommended to be applied first as it gives basic clues about the existence of nonrandom regions in a sequence. If this test fails, it is likely that other tests will also fail. The most complex test in terms of time criteria is the Linear Complexity test.

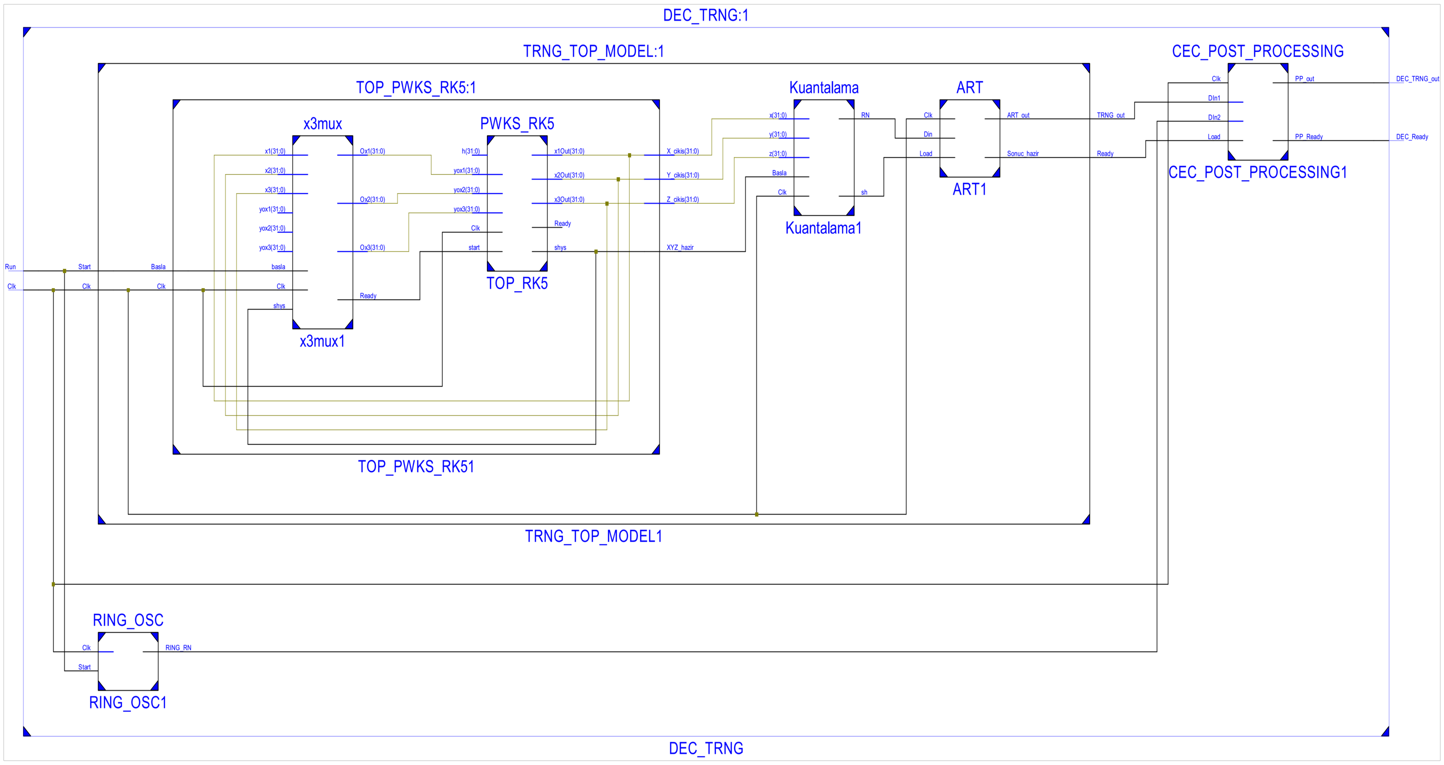


Figure 3 The second level block diagram of the FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO.

Table 1 The area utilization report of FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO unit on Virtex-7.

Utilization for 7VX485TFFG1761-2 Device	Used	Available	Utilization %
Number of Slice Registers	85.763	607.200	14
Number of Slice LUTs	85.294	303.600	28
Number of fully used LUT-Flip Flop Pairs	69.011	102.046	67
Number of Inputs/Outputs	4	700	1
Number of BUFG/BUFGCTRLs	1	32	3
Latency (ns)	702	-	-
Min. clock period (ns)	2.288	-	-

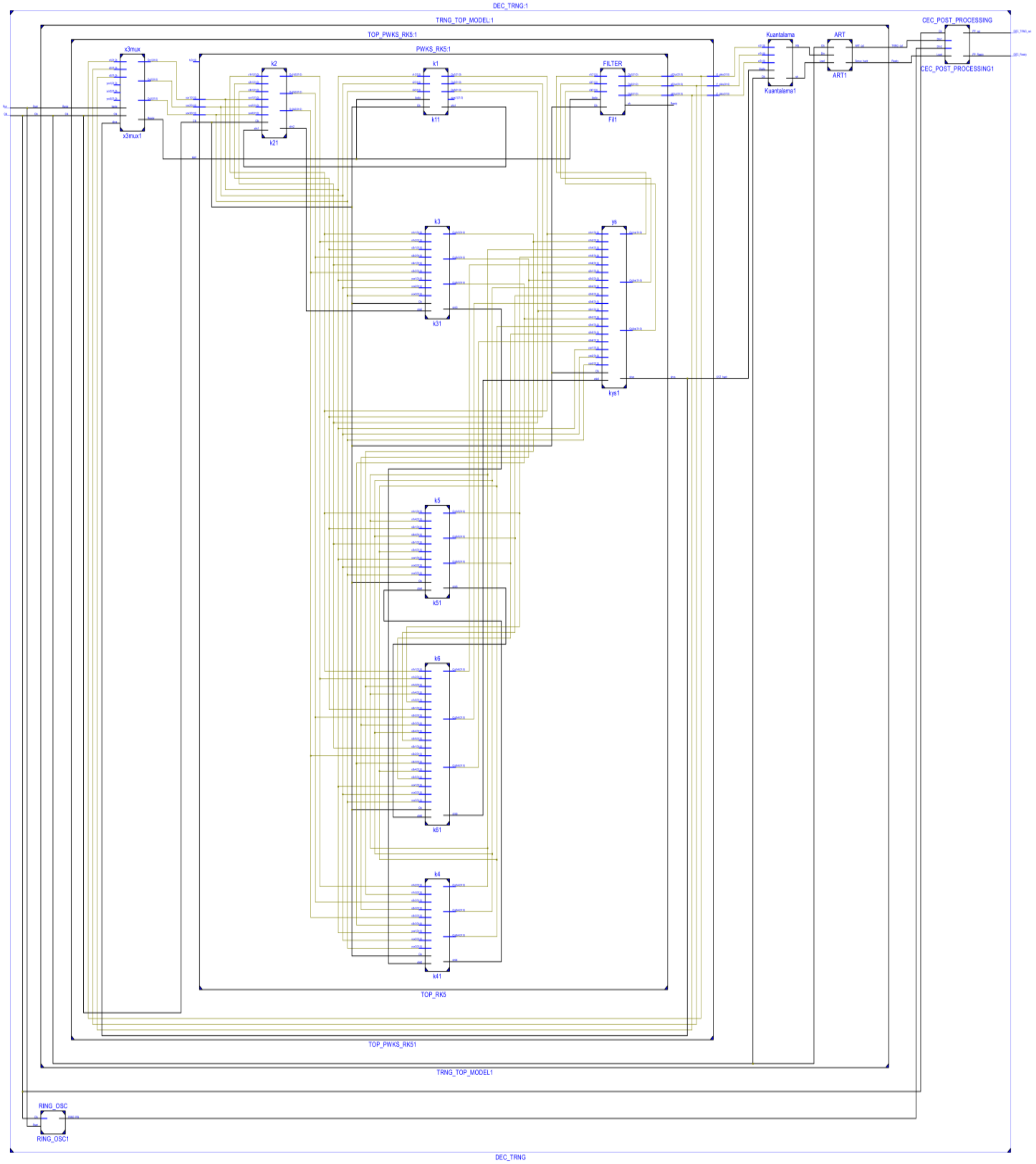
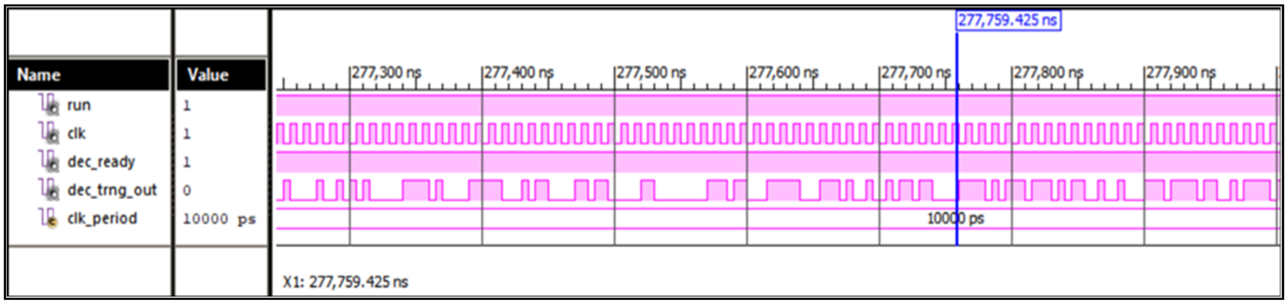


Figure 4 The third level block diagram of FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO.



Name	Value
run	1
clk	1
dec_ready	1
dec_trng_out	0
clk_period	10000 ps

Figure 5 The operation timing diagram of FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO unit obtained from Xilinx ISE Simulator.

Table 2 TThe NIST test results of FPGA-based DEC TRNG design using Ring and RK5-Butcher based on PWCO unit.

NIST 800-22 Statistical Tests	P-value	Result
Frequency Test	0.80568	Successful
Block Frequency Test	0.33645	Successful
Runs Test	0.75218	Successful
Longest Runs of One’s Test	0.834183	Successful
Binary Matrix Rank Test	0.73924	Successful
Discrete Fourier Transform (FFT) Test	0.48553	Successful
Non-Overlapping Template Matching Test	0.47564	Successful
Overlapping Template Matching Test	0.26366	Successful
Maurer’s “Universal Statistical” Test	0.67244	Successful
Linear Complexity Test	0.21416	Successful
Serial Test 1	0.33020	Successful
Serial Test 2	0.68817	Successful
Approximate Entropy Test	0.40933	Successful
Cumulative Sums (Forward) Test	0.87979	Successful
Random Excursions Test (for x=-3)	0.33195	Successful
Random Excursions Variant Test (for x= 3)	0.28495	Successful

1 million bits of data were collected and saved in a file for the system designed for testing. Then the bit file was subjected to 16 tests in the NIST Test Suite and all the sequences obtained were successful in all the randomness tests. In this test, some parameters of the random bit stream to be tested can be determined externally. P-value, which is one of the most important parameters in these tests, is accepted as a measure of the randomness of the random sequences subjected to the test. If a P-value for a test is determined to be equal to 1, then the sequence appears to have perfect randomness. A P-value of zero indicates that the sequence appears to be completely non-random. A significance level (α) can be chosen for the tests. Typically, the, α is chosen in the range $[0.001, 0.01]$. For this study, α parameter has been choosen as 0.01. As can be seen from the test results in Table 2, since the P-value $\geq$ 0.01, the obtained sequences are accepted randomly.

CONCLUSION

This paper presents a novel FPGA based Dual Core TRNG unit implemented in discrete time. In this direction, in the first stage, 3D PWCO has been modeled with RK5-Butcher numerical method and chaos analyses were performed by examining the dynamic behavior of the systems. Then, the PWCO was modeled on FPGA using the hardware description language as VHDL in accordance with the 32 bit IQ-Math fixed point number standard. RK5-Butcher numerical method was used in the modeling phase. The ring oscillator and PWCO designs were harvested in the post processing unit and the proposed TRNG design was implemented on FPGA. The proposed TRNG is capable of producing a high throughput of 437.043 Mbit/s after post-processing. Apart from the studies presented in the literature, post- processing has been performed without the decrease in the bit production rate. In the last part, number streams acquired from the presented TRNG unit have been applied to NIST 800-22 Test Suite. The test results have shown that the proposed TRNG unit can be used in the cryptographic systems. In addition, when this study is compared with other studies and methods presented in the literature, it offers very successful results in terms of both operating frequency and throughput.

CONFLICTS OF INTEREST

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

LITERATURE CITED

Adiyaman, Y., S. EMİROGLU, M. K. UÇAR, and M. YILDIZ, 2020 Dynamical analysis, electronic circuit design and control application of a different chaotic system. Chaos Theory and Applications **2**: 10–16.

Akgul, A., C. ARSLAN, and B. ARICIOĞLU, 2019 Design of an interface for random number generators based on integer and fractional order chaotic systems. Chaos Theory and Applications **1**: 1–18.

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

Akgul, A., S. Hussain, and I. Pehlivan, 2016b A new three-dimensional chaotic system, its dynamical analysis and electronic circuit applications. Optik **127**: 7062–7071.

Akkaya, S., İ. Pehlivan, A. Akgül, and M. Varan, 2018 Yeni bir kaos tabanlı rasgele sayı üreteci kullanan banka şifrematik cihazı tasarımı ve uygulaması. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi **33**: 1171–1182.

Alcin, M., 2020 The runge kutta-4 based 4d hyperchaotic system design for secure communication applications. Chaos Theory and Applications **2**: 23–30.

Alcin, M., I. Koyuncu, M. Tuna, M. Varan, and I. Pehlivan, 2019 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 **47**: 365–378.

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.

Avaroglu, E. and T. Tuncer, 2020 A novel s-box-based post-processing method for true random number generation. Turkish Journal of Electrical Engineering & Computer Sciences **28**: 288–301.

Benkouider, K., T. Bouden, and M. E. Yalcin, 2020 A snail-shaped chaotic system with large bandwidth: dynamical analysis, synchronization and secure communication scheme. SN Applied Sciences **2**: 1–15.

Bonny, T., R. Al Debsi, S. Majzoub, and A. S. Elwakil, 2019 Hardware optimized fpga implementations of high-speed true random bit generators based on switching-type chaotic oscillators. Circuits, Systems, and Signal Processing **38**: 1342–1359.

Bonny, T. and A. S. Elwakil, 2018 Fpga realizations of high-speed switching-type chaotic oscillators using compact vhdl codes. Nonlinear Dynamics **93**: 819–833.

Bonny, T. and Q. Nasir, 2019 Clock glitch fault injection attack on an fpga-based non-autonomous chaotic oscillator. Nonlinear Dynamics **96**: 2087–2101.

Buchovecka, S., R. Lórencz, F. Kodỳtek, and J. Buček, 2017 True random number generator based on ring oscillator puf circuit. Microprocessors and Microsystems **53**: 33–41.

Coskun, S., I. Pehlivan, A. AKGÜL, and B. GÜREVİN, 2019 A new computer-controlled platform for adc-based true random number generator and its applications. Turkish Journal of Electrical Engineering & Computer Sciences **27**: 847–860.

Dursun, M. and E. Kaşifoğlu, 2018 Design and implementation of the fpga-based chaotic van der pol oscillator. International Advanced Researches and Engineering Journal **2**: 309–314.

Etem, T. and T. Kaya, 2020 A novel true random bit generator design for image encryption. Physica A: Statistical

Mechanics and its Applications **540**: 122750.

Garipcan, A. M. and E. Erdem, 2019 Implementation and performance analysis of true random number generator on fpga environment by using non-periodic chaotic signals obtained from chaotic maps. Arabian Journal for Science and Engineering **44**: 9427–9441.

Gupta, R., A. Pandey, and R. K. Baghel, 2019 Fpga implementation of chaos-based high-speed true random number generator. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields **32**: e2604.

Hagras, E. A. and M. Saber, 2020 Low power and high-speed fpga implementation for 4d memristor chaotic system for image encryption. Multimedia Tools and Applications **79**: 23203–23222.

Kaya, T., 2020 A true random number generator based on a chua and ro-puf: design, implementation and statistical analysis. Analog Integrated Circuits and Signal Processing **102**: 415–426.

Koyuncu, İ., M. Alçin, M. Tuna, İ. Pehlivan, M. Varan, *et al.*, 2019 Real-time high-speed 5-d hyperchaotic lorenz system on fpga. International Journal of Computer Applications in Technology **61**: 152–165.

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, and I. Pehlivan, 2014 Implementation of fpga-based real time novel chaotic oscillator. Nonlinear Dynamics **77**: 49–59.

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.

Li, S., G. Chen, and X. Mou, 2005 On the dynamical degradation of digital piecewise linear chaotic maps. International journal of Bifurcation and Chaos **15**: 3119–3151.

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.

Öztürk, İ. and R. Kiliç, 2014 Cycle lengths and correlation properties of finite precision chaotic maps. International Journal of Bifurcation and Chaos **24**: 1450107.

Öztürk, I. and R. Kılıç, 2019 Higher dimensional baker map and its digital implementation with lsb-extension method. IEEE Transactions on Circuits and Systems I: Regular Papers **66**: 4780–4792.

Pano-Azucena, A., E. Tlelo-Cuautle, G. Rodriguez-Gomez, and L. De la Fraga, 2018 Fpga-based implementation of chaotic oscillators by applying the numerical method based on trigonometric polynomials. AIP Advances **8**: 075217.

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.

Rezk, A. A., A. H. Madian, A. G. Radwan, and A. M. Soliman, 2019 Reconfigurable chaotic pseudo random number generator based on fpga. AEU-international Journal of Electronics and Communications **98**: 174–180.

Rivera-Blas, R., S. A. Rodríguez Paredes, L. A. Flores-Herrera, and I. Adrián Romero, 2019 Design and implementation of a microcontroller based active controller for the synchronization of the petrzela chaotic system. Computation **7**: 40.

Sambas, A., S. Vaidyanathan, E. Tlelo-Cuautle, B. Abd-El-Atty, A. A. Abd El-Latif, *et al.*, 2020 A 3-d multi-stable system with a peanut-shaped equilibrium curve: Circuit design, fpga realization, and an application to image encryption. IEEE Access **8**: 137116–137132.

Taskiran, Z. and H. Sedef, 2020 Realization of memristor-based chaotic rossler circuit. J. Fac. Eng. Archit. Gazi Univ. **35**: 765–774.

Tlelo-Cuautle, E., J. Rangel-Magdaleno, A. Pano-Azucena, P. Obeso-Rodelo, and J. C. Nuñez-Perez, 2015 Fpga realization of multi-scroll chaotic oscillators. Communications in Nonlinear Science and Numerical Simulation **27**: 66–80.

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. Fidan, 2018 A study on the importance of chaotic oscillators based on fpga for true random number generating (trng) and chaotic systems .

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.

Yoo, S.-K., D. Karakoyunlu, B. Birand, and B. Sunar, 2010 Improving the robustness of ring oscillator trngs. ACM Transactions on Reconfigurable Technology and Systems (TRETS) **3**: 1–30.

How to cite this article: Alcin, M., Tuna, M., Erdogmus, P. and Koyuncu, I. FPGA-based Dual Core TRNG Design Using Ring and Runge-Kutta-Butcher based on Chaotic Oscillator. Chaos Theory and Applications, 3(1), 20-28, 2021.