IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 30, 2019, accepted November 19, 2019, date of publication November 28, 2019,
date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956573

Design and FPGA Implementation of a Pseudorandom Number Generator Based on a Four-Wing Memristive Hyperchaotic System and Bernoulli Map

FEI YU[1], **LIXIANG LI**[1], **BINYONG HE**[1], **LI LIU**[1], **SHUAI QIAN**[1], **YUANYUAN HUANG**[1], **SHUO CAI**[1], **YUN SONG**[1], **QIANG TANG**[1], **QIUZHEN WAN**[2], **AND JIE JIN**[3]

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China
[2]College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China
[3]School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

Corresponding authors: Fei Yu (yufeiyfyf@csust.edu.cn) and Yuanyuan Huang (snailhyy@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61504013, Grant 61702052, Grant 61772087, and Grant 61901169, in part by the Natural Science Foundation of Hunan Province under Grant 2019JJ50648, Grant 2016jj2005, Grant 2019JJ40190, and Grant 2017JJ3254, in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 18A137, Grant 18B162, and Grant 16B212, and in part by the National Key Research and Development Project under Grant 2018YFE0111200.

ABSTRACT Random numbers are widely used in the fields of computer, digital signature, secure communication and information security. Especially in recent years, with the large-scale application of smart card and the demand of information security, the demand for high-quality random number generator is increasingly urgent. With the development of the theory of non-linear systems, the design of pseudorandom number generator (PRNG) for chaotic behavior of non-linear systems provides a new theoretical basis and implementation method. This paper presents a PRNG based on a no-equilibrium four-wing memristive hyperchaotic system (FWMHS) and its implementation on Field Programmable Gate Array (FPGA) board. In order to increase the output throughput and the statistical quality of the generated bit sequences, we propose the PRNG design which uses a dual entropy sources architecture with FWMHS and Bernoulli map. Simulation and experimental results verifying the feasibility of the FWMHS are also given. Then, the proposed PRNG system is modeled and simulated on the Vivado 2018.3 platform, and implemented on the Xilinx ZYNQ-XC7Z020 FPGA evaluation board. The maximum operating frequency has been achieved as 135.04 MHz with a speed of 62.5 Mbit/s. Finally, we have experimentally verified that the binary data obtained by this dual entropy sources architecture pass the tests of NIST 800.22, ENT and AIS.31 statistical test suites with XOR function post-processing for a high throughput speed. The security analysis is carried out by means of dynamical degradation, key space, key sensitivity, correlation and information entropy. Statistical tests and security analysis show that it has good pseudorandom characteristics and can be used in chaos-based cryptographic applications at hardware or software implementation.

INDEX TERMS Pseudorandom number generator (PRNG), FPGA, four-wing memristive hyperchaotic system (FWMHS), Bernoulli map, security analysis.

I. INTRODUCTION

With the popularity of personal computers and networks, electronic commerce, digital signatures, multimedia communications and other fields have developed rapidly. At the same time, information security issues in these areas are also getting more and more attention [1]–[5]. Almost all fields of cryptography require values that are unknown to an attacker [6]–[10]. Obviously, the best choice is random number (RN). Random number generators (RNGs) are widely used in many fields. For example, in asymmetric (public key) algorithms such as RSA, DSA and Diffie-Hellman, they are used to generate public or private keys and generate keys in symmetric and hybrid cryptosystems [11]–[14].

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

RNG can be basically divided into two categories: true random number generator (TRNG) by using physical process (non-deterministic) and pseudorandom number generator (PRNG) by using mathematical algorithms (deterministic) [15]. TRNG uses processes of non-deterministic physical properties, such as thermal noise, photon noise, quantum random processes, frequency jitter in oscillators and chaotic oscillator [16]–[19]. They can be digitally sampled and post-processing techniques can be implemented to improve randomness. TRNG should be non-reproducible, unpredictable and statistically unbiased. PRNG uses deterministic digital processes through digital algorithms, which are based on algorithms that generate pseudo-random decision sequences from an initial value called seed in mathematical processes. PRNG should have good statistical characteristics, fast execution time, repeatability and reproducibility, and its security must be based on the difficulty of solving related mathematical problems [15], [20].

Because of the sensitivity of chaos to initial conditions and the unpredictability of long-term behavior, it is proved that chaos exists in almost all engineering fields [21]–[27]. With the further study of chaos theory, chaotic systems have been widely used in many fields, such as synchronization [28]–[32], secure communication [33]–[38], cryptography [39]–[42], artificial neural network [43]–[47], image processing [48]–[52] and RNGs [53]–[57]. Among them, chaos generator is one of the most fundamental structures in the application of chaos engineering. Recently, many literatures have introduced the design method of PRNGs based on chaos theory [56], [57]. In [56], a PRNG integrated circuit is designed with an adoption of Logistic map and a feedback mechanism is introduced to increase the cycle length of the digitized chaotic map. The simulation results show that the nonlinear feedback Boolean function can be a suitable randomness source and can pass the standard statistical tests. In [57], a chaos-based logistic map PRNG is designed using a reseeding-mixing method which can extend the system period length and enhance the statistical properties. The reseeding-mixing PRNG is implemented in the TSMC 0.18-μm 1P6M CMOS process that attains the best throughput rate of 6.4 Gb/s compared with other nonlinear PRNGs. The generated random sequences pass the NIST SP800.22 statistical tests including ratio test and U-value test. Considering the technologies of [56] and [57], chaotic PRNG based on IC can achieve the highest performance. However, the implementation based on IC do not guarantee a flexible use. In addition, the prototype design and test cost of the system are also very high [58]–[63]. Only in the case of mass production, the system cost can be greatly reduced. The structures of the Field Programmable Gate Array (FPGA) chip are relatively flexible because it can run in parallel [64]. Moreover, the design and test cycle cost of the FPGA chips are extremely low. In order to increase and expand engineering applications based on chaos, current chaotic systems should be diversified and supported by flexible architecture [54]. With the digitalization and reconfiguration of the FPGA, the chaotic system and its application can be more flexible.

In recent years, there exist several studies related to FPGA-based PRNG designs of chaotic systems [20], [64]–[66]. In [20], four chaotic maps: Bernoulli shift map, tent, zigzag, and Borujeni maps are selected to implement a PRNG. The binary sequences obtained from these maps are analyzed with FPGA, and the randomness of the generated binary sequences with the NIST test suite 800.22 both in floating point and fixed point arithmetic is validated. In [65], a novel FPGA-based PRNG is proposed by using Lorenz and Lü chaotic systems. These two systems are used to generate four different three-dimensional chaotic attractors. The output attractor of the proposed PRNG can be reconfigured using an effective hard-wire shift and multiplexing scheme during real-time operation. In [66], a high-speed PRNG model is implemented on the FPGA by using enhanced Henon map, which has been passed by a comprehensive security analysis. The random bit set generated by PRNG is further verified by NIST 800.22 statistical tests, which proves that the design can be used in cryptographic applications.

As a complex dynamic behavior, hyperchaos is more complex than chaotic behavior, and has higher application potential in the fields of secure communication and information security [67]–[70]. Reference [71] presents a PRNG using a hyperchaotic system with bigger Lyapunov exponent. Then the self-shrinking generator is used to perturb the hyperchaotic sequences to decrease the period degeneration and improve the performance of the sequences. In order to generate high quality true RNs at a fast rate, a new PRNG based on a hyperchaotic multi-scroll piecewise linear system is introduced in [72]. The generated random numbers were evaluated using the NIST statistical test suite. The results show that the method can generate RNs at a high rate while guaranteeing the statistical quality.

The memristor is a non-linear element proposed by Chua [73] in 1971 to describe the relationship between charge and flux based on the completeness of the circuit. It was successfully implemented by Strukov *et al.* [74]. Memristor is a new type of electronic device. Because of its unique memory, the memristor chaotic system constructed by memristor has more complex nonlinear dynamic phenomena than the general chaotic system [75]–[80]. The memristor chaotic system not only shows sensitivity in circuit parameters, but also varies with the initial value of memristor. Using the nonlinearity and memory characteristics of memristors as feedback terms of hyperchaotic systems, complex nonlinear dynamic phenomena (like four-wing attractor) can be generated, which provides a new development space for the design of hyperchaotic systems [81]–[86]. Moreover, the random sequence generated by the memristor hyperchaotic system is more difficult to predict than the general chaotic signal. In [87], a PRNG based on a 1-D memristor-based Cellular Automata (CAs) array is presented. The memristor device is considered to be the basic module of a CA cell circuit implementation, performing as a combined memory and processing element to implement CA-based circuits. Up to now, no PRNG based on memristor hyperchaotic system has been proposed with the author's knowledge.

In the present paper, based on a no-equilibrium four-wing memristive hyperchaotic system (FWMHS) introduced in [88], a PRNG used in FPGA-based design is proposed. Most importantly, the novel PRNG uses a dual entropy sources architecture with FWMHS and Bernoulli map to increase the output throughput and the statistical quality of the generated bit sequences. The results obtained from FPGA-based FWMHS have been compared with the computer-based results. Then, the proposed PRNG is modeled and simulated on Vivado 2018.3 platform, and implemented on Xilinx ZYNQ-XC7Z020 FPGA evaluation board while the XOR function is used for post-processing. The experimental results are consistent with expectations. The bit sequence of the proposed PRNG successfully passes all statistical tests of NIST 800.22, ENT and AIS.31 test suites, and the ultimate output bit rate is 62.5 Mbps. Finally, the security analysis is carried out by means of dynamical degradation, key space, key sensitivity, correlation and information entropy.

The rest of the paper is organized as follows. The mathematical models of a no-equilibrium FWMHS and Bernoulli map are illustrated in Section II. In Section III, the FPGA-based model of FWMHS is introduced and simulation results of FPGA-based model are presented. In Section IV, the architecture of the PRNG is illustrated, and the implementation of the proposed PRNG on the FPGA platform is discussed. Experimental and statistics tests results for the validation of the architecture are also presented. The security analysis are presented in Section V. Finally, this paper is concluded in Section VI.

II. THE MATHEMATICAL MODELS OF FWMHS AND BERNOULLI MAP

A. A NO-EQUILIBRIUM FWMHS

The four-wing hyperchaotic system based on a memristor is a research hotspot at present. Many different four-wing hyperchaotic systems based on a memristor have been introduced successively [81], [82], [88]. Recently, a no-equilibrium FWMHS was proposed in [88] by considering a control parameter. The FWMHS shows both the no equilibrium and line equilibrium conditions for different values of the control parameter. This system is given by the following equation

$$\begin{cases} \dot{x} = ax + byz, \\ \dot{y} = cy + dxz - pyW(w) - Q, \\ \dot{z} = ez + fxy + gxw, \\ \dot{w} = -y, \end{cases} \tag{1}$$

where

$$W(w) = m + 3nw^2, \tag{2}$$

and where x, y, z, w are state variables and $W(w)$ is the memductance, that represents the non-linearity of the flux-controlled memristor with parameters m and n. Q is the control parameter, when $Q \neq 0$, the FWMHS exhibits hyperchaotic attractors without equilibrium points; When $Q = 0$, the system has a line equilibrium in $(0, 0, 0, w)$ and this line equilibrium is consisting of infinite unstable [88]. To obtain a hyperchaotic behavior, system parameters are chosen as $a = 0.35, b = -10, c = -0.6, d = 0.3, e = -1.6, f = 2, g = 0.1, m = 0.1, n = 0.01, p = 0.2$ and $Q = 0.01$ and the initial conditions are determined as $[0.1, 0.1, 0.1, 0.1]$, the system has a typical four-wing hyperchaotic attractor, as shown in Fig. 1.

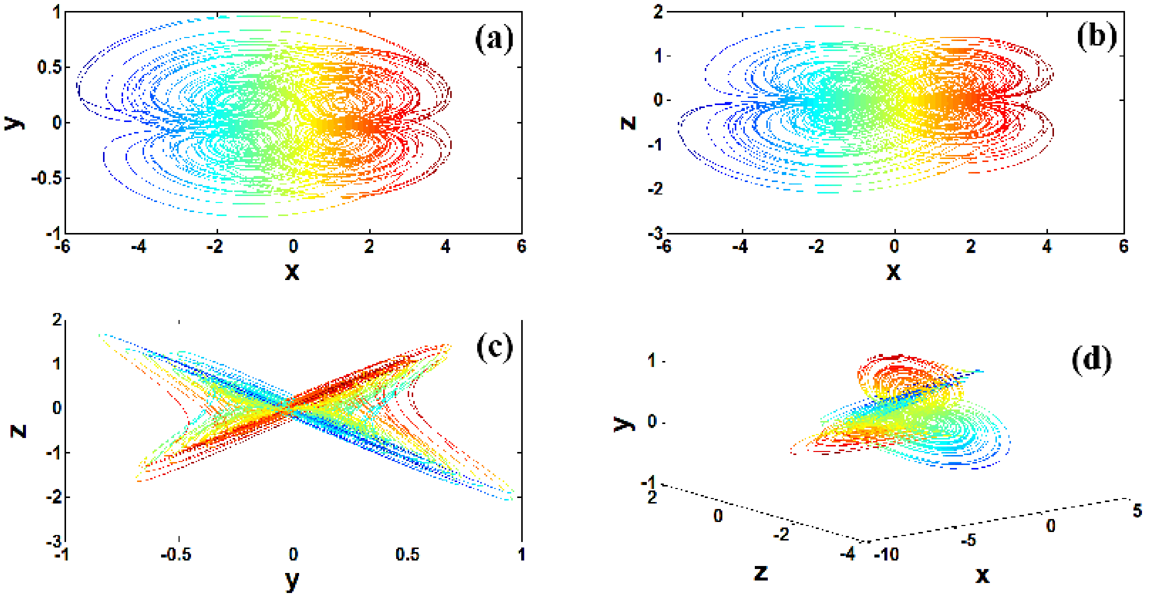


FIGURE 1. The results of FWMHS attractors obtained from Matlab: (a) $x - y$, (b) $x - z$, (c) $y - z$ phase portraits and (d) 3D $(x - y - z)$ chaotic phase portrait.

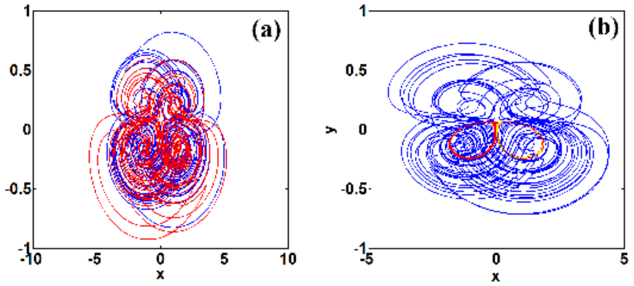


FIGURE 2. Multistable attractors for the conditions: (a) $Q = 0.0505$ and the initial conditions are $[-0.692, 0.04, -0.065, -0.375]$ and $[-2.38, 0.07, -0.281, 1.432]$ which are shown in blue and red respectively; (b) $Q = -0.0508$ and the initial conditions are $[2.062, 0.063, 0.028, -0.765]$, $[0.0224, 0.052, -0.005, -3.665]$ and $[-0.0224, 0.052, -0.005, -3.665]$ which are shown in blue, red and yellow.

With respect to performed analysis, the Lyapunov exponents have been calculated as $LE1 = 0.1032$, $LE2 = 0.0149$, $LE3 = 0$ and $LE4 = -1.996$ when $Q = 0.01$. As can be seen on the results, since the signs of the Lyapunov exponents $(LE1, LE2, LE3, LE4)$ are $(+, +, 0, -)$, respectively, the four-wing memristive chaotic system is hyperchaotic.

Multistability refers to coexisting of multiple attractors depending on different initial conditions, which is the inherent characteristic of nonlinear systems and shows the rich multiple steady state of the nonlinear dynamic system. In order to study the multistability of the FWMHS, we change the initial conditions and control parameter Q, and keep the system other parameters unchanged to observe the attractor phenomenon, and find that the system has hidden and multistable attractors [88]. When $-0.05055 < Q < 0.0513$, we can see coexisting attractor with a limit cycle and coexisting attractors of $Q = 0.0505$ as shown in Fig. 2(a). We also observed coexisting period-8 oscillations of $Q = -0.0508$ is shown in Fig. 2(b).

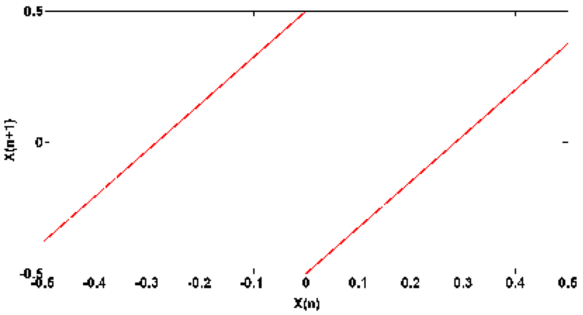


FIGURE 3. The graph of Bernoulli map function.

B. BERNOULLI MAP

Bernoulli map consists of two piecewise linear parts separated by a discontinuity point, as shown in Fig. 3. In mathematical terms, Bernoulli map is defined as

$$x_{n+1} = \begin{cases} Bx_n + 0.5, & x < 0 \\ Bx_n - 0.5, & x \geq 0 \end{cases} \tag{3}$$

where B is the chaos control parameter, and when $1 \leq B \leq 1.4$, the sequence of Bernoulli map will appear multiple periodic points, and when $1.4 < B \leq 2$, the map will appear chaotic state, and all trajectory points will be linked together. Bernoulli map not only has uniform probability density distribution function, but also has similar time probability density distribution and statistical probability density distribution, which makes it have good ergodicity.

Random bits are generated by means of a threshold comparator, which is the defined as

$$b(n) = B(x_n, T_h) = \begin{cases} 0, & x_n \leq T_h \\ 1, & x_n > T_h \end{cases} \tag{4}$$

where T_h is a threshold parameter, and the comparator uses to divide the phase space into two bit generating partitions. In order to achieve the best quantization effect, according to the distribution characteristics of Bernoulli's iteration value, T_h is 0. When $B = 1.75$, the sequence generated iteratively by Bernoulli map, as shown in Fig. 4(a), and by sampling and quantizing Bernoulli map, the random bits stream is generated, as shown in Fig. 4(b).

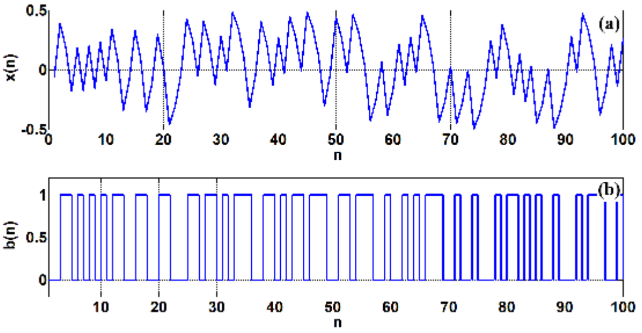


FIGURE 4. The graphs of Bernoulli map: (a) after sampling; (b) after comparison.

III. THE FPGA-BASED MODEL OF FWMHS

As an important kind of integrated circuit chip, FPGA is widely used in communication, information security, industry, automobile, Internet of Things, artificial neural networks, consumer electronics, image processing, artificial intelligence and chaotic systems design [89]–[94]. There are many algorithms to solve differential equations in literature, such as Euler, Heun, RK4 and RK5 butcher. RK4 is a derivative of Runge-Kutta basic model, which is used to solve ordinary differential equations with high accuracy, and mostly has proved itself superior to other solutions.

A. RK4 ALGORITHM

In this study, RK4 algorithm has been used for performing FPGA-based digital model of the FWMHS. Eq. (5) gives the mathematical equation of this method. Where iteration step $\Delta h = 0.001$, K_1, K_2, K_3 and K_4 represent the slope values at four points on $[x_k, x_{k+1}]$, respectively. u_k and x_k are

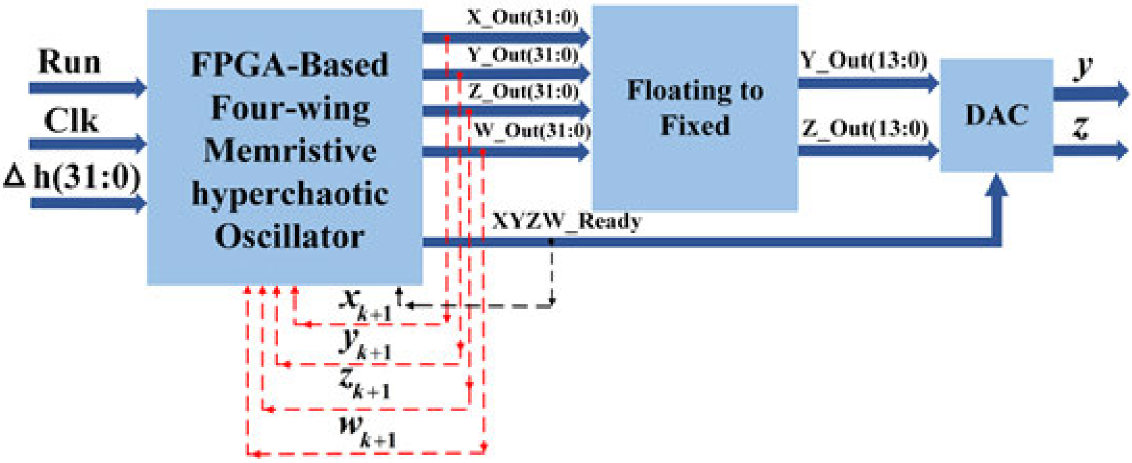


FIGURE 5. **Flow block diagram of the FPGA-based FWMHS.**

the values of time $t = t_k$. Similarly, u_{k+1} and x_{k+1} are the values of time $t = t_{k+1}$.

$$\begin{aligned}
K_1 &= \Delta h f(x_k, u_k)\\
K_2 &= \Delta h f(x_k + \frac{\Delta h}{2}, u_k + \frac{1}{2}K_1)\\
K_3 &= \Delta h f(x_k + \frac{\Delta h}{2}, u_k + \frac{1}{2}K_2)\\
K_4 &= \Delta h f(x_k + \Delta h, u_k + K_3)\\
u_{k+1} &= u_k + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)
\end{aligned} \tag{5}$$

Four equations of FWMHS are substituted into Eq. (5), and the four state variables (x, y, z, w) of FWMHS are solved respectively, as shown in Eq. (6). Where $K_{i1}, K_{i2}, K_{i3}, K_{i4}$ $(i = x, y, z, w)$ parameters in Eq. (6) representation the slope of RK4 method for the System (1). When $t = 0$, the initial values of x_k, y_k, z_k and w_k in numerical model are chosen as $x_0 = 0.1, y_0 = 0.1, z_0 = 0.1, w_0 = 0.1$.

$$\begin{aligned}
K_{x1} &= ax_k + by_k z_k\\
K_{x2} &= \Delta h(a(x_k + \frac{K_{x1}}{2}) + by_k z_k)\\
K_{x3} &= \Delta h(a(x_k + \frac{K_{x2}}{2}) + by_k z_k)\\
K_{x4} &= \Delta h(a(x_k + K_{x3}) + by_k z_k)\\
x_{k+1} &= x_k + \frac{1}{6}(K_{x1} + K_{x2} + K_{x3} + K_{x4})\\
K_{y1} &= cy_k + dx_{k+1}z_k - py_k(m + 3n{w_k}^2) - Q\\
K_{y2} &= \Delta h(c(y_k + \frac{K_{y1}}{2}) + dx_{k+1}z_k\\
&\quad - p(y_k + \frac{K_{y1}}{2})(m + 3nw_k^2) - Q)\\
K_{y3} &= \Delta h(c(y_k + \frac{K_{y2}}{2}) + dx_{k+1}z_k\\
&\quad - p(y_k + \frac{K_{y2}}{2})(m + 3nw_k^2) - Q)\\
K_{y4} &= \Delta h(c(y_k + K_{y3}) + dx_{k+1}z_k\\
&\quad - p(y_k + K_{y3})(m + 3nw_k^2) - Q)\\
y_{k+1} &= y_k + \frac{1}{6}(K_{y1} + K_{y2} + K_{y3} + K_{y4})\\
K_{z1} &= \Delta h(ez_k + fx_{k+1}y_{k+1} + gx_{k+1}w_k)\\
K_{z2} &= \Delta h(e(z_k + \frac{K_{z1}}{2}) + fx_{k+1}y_{k+1} + gx_{k+1}w_k)\\
K_{z3} &= \Delta h(e(z_k + \frac{K_{z2}}{2}) + fx_{k+1}y_{k+1} + gx_{k+1}w_k)\\
K_{z4} &= \Delta h(e(z_k + K_{z3}) + fx_{k+1}y_{k+1} + gx_{k+1}w_k)\\
z_{k+1} &= z_k + \frac{1}{6}(K_{z1} + 2K_{z2} + 2K_{z3} + K_{z4})\\
K_{w1} &= \Delta h(-y_k)\\
K_{w2} &= \Delta h(-y_k)\\
K_{w3} &= \Delta h(-y_k)\\
K_{w4} &= \Delta h(-y_k)\\
w_{k+1} &= w_k + \frac{1}{6}(K_{w1} + K_{w2} + K_{w3} + K_{w4})
\end{aligned} \tag{6}$$

B. FPGA IMPLEMENTATION

According to the high-precision 32-bit IEEE 754-1985 floating-point standard, FWMHS is modeled as a system based on FPGA by using RK4 algorithm in VHDL language. The IP core generator developed by Vivado 2018.3 design tool system is used to design multiplier, subtractor and adder for chaotic oscillator based on FPGA, which conforms to IEEE 754-1985 standard. The flow block diagram of chaotic signal generator based on FPGA implemented by RK4 algorithm is shown in Fig. 5. As shown in Fig. 5, the designed system has three inputs and two outputs ($y - z$ phase portrait). *Run* and *Clk* are the inputs of the system, and they are one bit signals, which are used for synchronization between the timing of all units and other units. Δh is an 32-bit input signal, which defines the step size. This signal is realized from outside, which makes the design more flexible.

The flow block diagram is composed of three units like FWMHS oscillator, floating to fixed and DAC unit. The oscillator unit has three input signals, which are 1-bit *Run*, *Clk* and 32bit Δh. At the outputs end of the

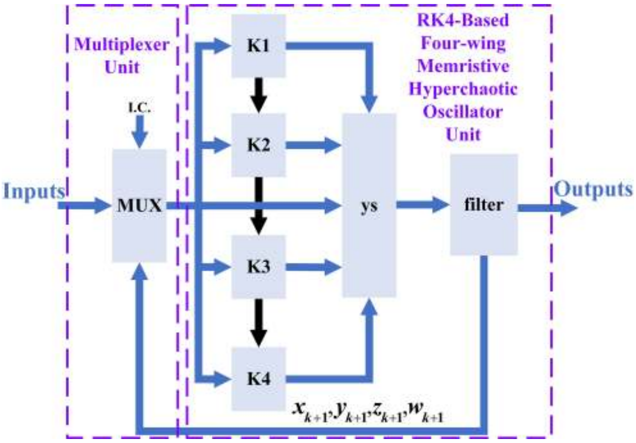

Multiplexer Unit
I.C.
Inputs
MUX
K1
K2
K3
K4
ys
filter
Outputs
RK4-Based Four-wing Memristive Hyperchaotic Oscillator Unit
$x_{k+1}, y_{k+1}, z_{k+1}, w_{k+1}$

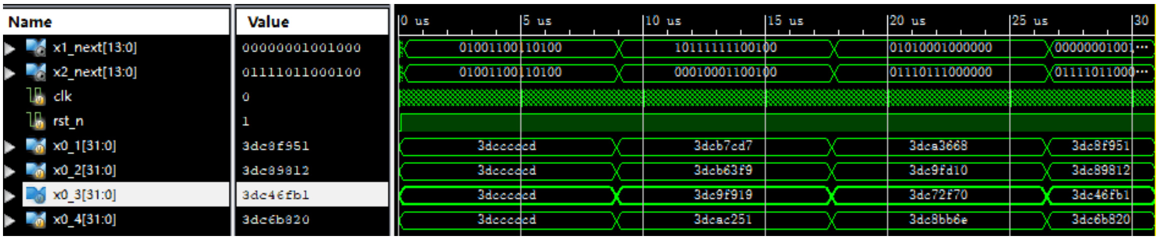
FIGURE 6. The block diagram of the FPGA-based chaotic oscillator unit.

FWMHS oscillator based on RK4 are four 32-bit output signals (X_Out, Y_Out, Z_Out and W_Out) at floating-point standards, and $XYZW\_Ready$ signals provide clock and enable signals for DAC unit. These signals are equivalent to x, y, z and w variables of FWMHS. When the output of the chaotic oscillator produces the first value, the value of $XYZW\_Ready$ is "1", and then the input of the multiplexer unit (MUX) receives the output signals from the oscillator. The inputs of floating to fixed-point unit are four 32-bit output signals of the oscillator unit, which converts the outputs of the previous unit into 14-bit unsigned fixed-point. The DAC unit converts the digital signals generated by the FWMHS into analog signals and outputs them to the oscilloscope. In practical experiments, we choose Y_Out and Z_Out to output to the dual-channel DAC module, and then output to the oscilloscope.

Fig. 6 shows the block diagram of the FWMHS oscillator using RK4 algorithm. x_0, y_0, z_0 and w_0 signals are the initial conditions ($I.C.$) for the system to start running. In the design, they are defined as 32-bit symbolic floating-point numbers, which are determined internally by the user. The purpose of MUX is to select the external initial conditions at the start or the internal values provided by RK4-based FWMHS oscillator unit in successive steps [51]. In the successive steps after the start of operation, the signals ($x_{k+1}, y_{k+1}, z_{k+1}$ and w_{k+1}) generated by the oscillator unit are used as the feedback inputs of the MUX, as the input signals (x_k, y_k, z_k and w_k) of the next step. The oscillator unit consists of six modules: $K1, K2, K3, K4, ys$ and filter. $K1, k2, K3$ and $K4$ modules are used to calculate the values of $K_{i1}, K_{i2}, K_{i3}, K_{i4}$ $(i = x, y, z, w)$, and ys module is used to calculate the values of $x_{k+1}, y_{k+1}, z_{k+1}$ and w_{k+1}. When ys does not produce the final required calculation results, the filter unit will prevent unnecessary values from reaching the output.

C. FPGA TEST RESULTS

The FWMHS based on RK4 is synthesized on Xilinx ZYNQ-XC7Z020 chip. The use of the chip source and the clock speed of the FWMHS are calculated. Using Vivado 2018.3 design tool, the data processing duration of the FWMHS designed in this paper is determined. The X_Out, Y_Out, Z_Out and W_Out signals are equivalent to the x, y, z and w signals in the FWMHS. The chaotic oscillator is discretized on the FPGA using Vivado 2018.3 design tool. Although 32-bit floating-point standard is adopted in the system design to detect the time series values of these signals more easily, Vivado simulation results are displayed in hexadecimal digital format. The results of the Xilinx Isim simulator for FWMHS are shown in Fig. 7 when $\Delta h = 0.001$. The system runs in pipeline mode and produces x, y, z and w signals after every 320 clock cycles. The chip statistics of chaotic oscillator implemented on FPGA are shown in Table 1. The minimum working period of FWMHS signal generator based on FPGA is 7.405 ns. Finally, the Y_Out and Z_Out signals obtained from the RK4-based FPGA design of FWMHS are recorded in a file in the form of 32-bit floating-point hexadecimal number during the test step, which is given in Table 2. The $y - z$ phase portrait of the output signals are obtained



Name	Value
x1_next[13:0]	00000001001000
x2_next[13:0]	01111011000100
clk	0
rst_n	1
x0_1[31:0]	3dc8f951
x0_2[31:0]	3dc89812
x0_3[31:0]	3dc46fb1
x0_4[31:0]	3dc6b820

FIGURE 7. Xilinx Isim Simulation results of the FWMHS based on FPGA.

TABLE 1. The Xilinx ZYNQ-XC7Z020 chip statistics of FPGA-based FWMHS.

FPGA chip	Slice register number	LUTs number	Bonded IOBs number	Max. clock frequency (MHz)
ZYNQ-XC7Z020	26,426	22,556	34	135.043
Utilization (%)	24	42	27	—

TABLE 2. The conversion result from 32-bit floating-point number output by FPGA to decimal number.

FPGA output signals in 32-bit floating-point number with hexadecimal format		Decimal number values	
Y_Out	Z_Out	$y\,(t)$	$z\,(t)$
3dcb63f9	3dc9f91a	0.0993117729756227	0.0986196568394701
3dc9fd10	3dc72f72	0.0986272076742696	0.0972584656747548
3dc89812	3dc46fb5	0.0979462974053773	0.0959161856107186
3dc734fd	3dc1b9c1	0.0972690350292466	0.0945925779289057
3dc5d3d1	3dbf0d78	0.0965954129754891	0.0932874060940825
3dc4748c	3dbc6ab9	0.0959254232608698	0.0920004357597591
3dc3172e	3db9d167	0.0952590575065645	0.0907314347727196
3dc1bbb5	3db74162	0.0952590575065645	0.0894801731765933
3dc06221	3db4ba8a	0.0939371624852399	0.0882464232144955
3dbf0a6e	3db23cc3	0.0932816146300924	0.0870299593307697
3dbdb49e	3dafc7ef	0.0926296535896916	0.0858305581718593
3dbc60ad	3dad5bee	0.0919812692468322	0.0846479985863375

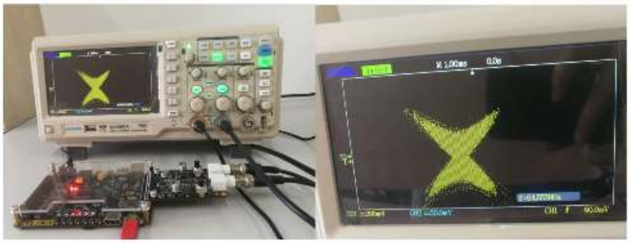


FIGURE 8. Implementation platform and exemplification $y - z$ phase portrait generated by the FPGA implementation of the FWMHS.

using the data set generated in decimal format by FWMHS based on FPGA given in Table 2. A picture of the $y - z$ phase portrait obtained from the hardware implementation of the RK4-based FWMHS on FPGA is shown in Fig. 8. The results show that the phase portrait obtained by the model based on MATLAB and FPGA have good consistency.

IV. DESIGN AND FPGA IMPLEMENTATION OF THE PRNG

The proposed PRNG incorporates four units: entropy source, sampling, quantization (comparator) and post-processing, the architecture is shown in Fig. 9, where a dual entropy sources is consisting of the FWMHS and Bernoulli map. Due to the difference of random numbers produced by chaotic oscillator and Bernoulli map in each iteration, two clock signals $Clk1$ and $Clk2$ with different sampling frequencies are designed in the implementation of FPGA. When sampling the four 32-bit floating-point signals (X_Out, Y_Out, Z_Out and W_Out) generated by the chaotic oscillator, we select the bits between 0 and 21 for sampling and output to the next step. In the quantization unit, two, three and all (four) 22-bit signals are simultaneously pre-processed by XOR function so that the chaotic oscillator generates 242-bit random numbers in each iteration period. The Bernoulli map is sampled and compared according to the sampling frequency of $Clk2$ and the threshold value T_h, and the output bit stream is sent to the post-processing unit. Then, the two bitstreams generated by the dual entropy sources are post-processed. The post-processor is used to improve the randomness of bit sequences generated by PRNG. In this design, XOR function is used as post-processing to reduce the utilization rate of the resources of the FPGA. The PRNG design steps are shown in **Algorithm 1** as pseudocode. When PRNG is implemented on FPGA, we take the period of $Clk1$ as 484 clock cycles, and the period of $Clk2$ as 2 clock cycles. For each iteration,

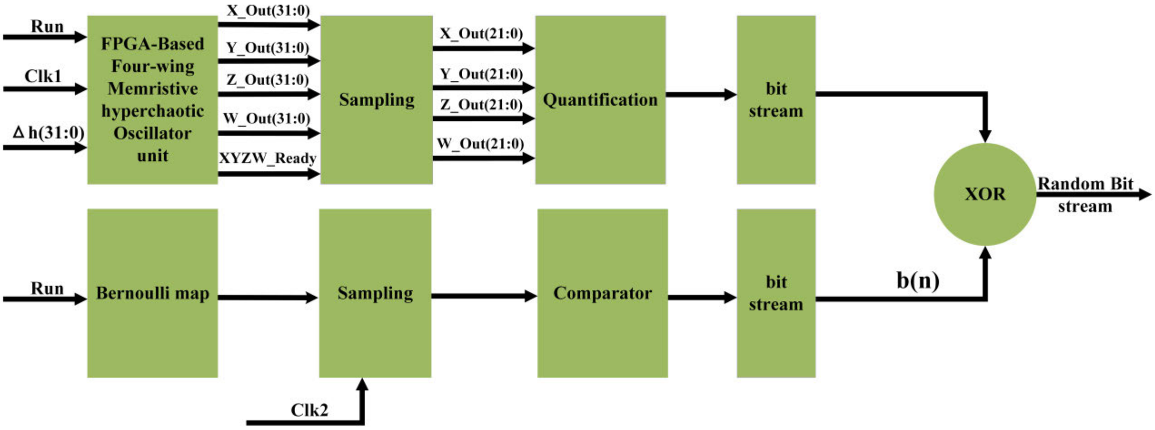


FIGURE 9. The architecture of the proposed PRNG based on FWMHS and Bernoulli map.

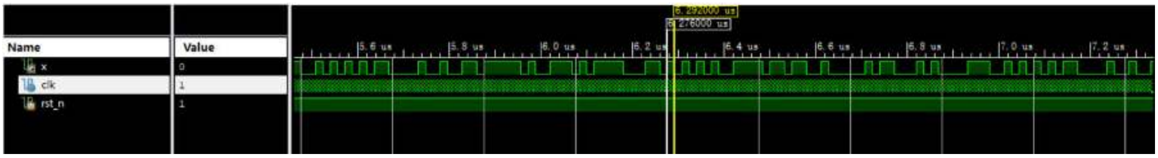


FIGURE 10. Simulation results of FPGA-based PRNG unit.

TABLE 3. The Xilinx ZYNQ-XC7Z020 chip statistics of the PRNG unit.

FPGA chip	Slice register number	LUTs number	Bonded IOBs number	Max. clock frequency (MHz)
ZYNQ-XC7Z020	27,371	24,836	3	135.04
Utilization (%)	25.72	46.68	2.4	—

Algorithm 1 PRNG Design Algorithm Pseudocode

```
Input: Parameter input and initial conditions of chaotic systems and Bernoulli map, $\Delta h$, $T_h$, $Clk1$, $Clk2$
Output: 1000 Mbit data
1. Repeat:
2. Obtain the $X\_Out(31:0)$, $Y\_Out(31:0)$, $Z\_Out(31:0)$, $W\_Out(31:0)$ according to (6);
3. Choose the last 22 decimal places from the $X\_Out(31:0)$, $Y\_Out(31:0)$, $Z\_Out(31:0)$, $W\_Out(31:0)$;
4. Obtain bit stream of chaotic system according to XOR of $xy$, $zw$, $xz$, $yw$, $xw$, $yz$, $xyz$, $xyw$, $yzw$, $xzw$ and $xyzw$;
5. Obtain $b(n)$ according to (3) and (4);
6. Obtain bit stream according to XOR of bit stream of chaotic system and $b(n)$;
7. $t = t + 1$;
8. Until: least 1000 Mbit data
9. Return: 1000 Mbit data
```

the FWMHS generates 242 bits of random number while Bernoulli map generates 1 bits of random number.

An example of simulation results obtained using Vivado 2018.3 FPGA is shown in Fig. 10. After synthesizing Verilog code, the FPGA editor gives a maximum operating frequency of 135.04 MHz. As can be seen in Fig. 10, the realized bit output rate is 62.5 Mbps. The statistics of the design of PRNG based on FPGA chips obtained from Vivado 2018.3 are given in Table 3. The experimental setup and the visualization on the oscilloscope of a typical output of the proposed PRNG is shown in Fig. 11.

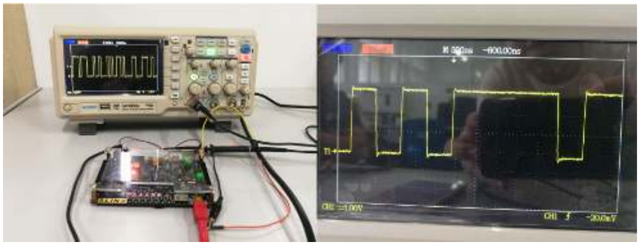


FIGURE 11. The implementation platform and RN waveform generated by the FPGA implementation of the proposed PRNG.

RN performance evaluation is an important aspect of RNGs, because it is necessary to test and evaluate the randomness and statistical characteristics of pseudo-random generators to determine whether they can be used in cryptographic applications. Statistical tests of RNs use samples to infer the overall situation. Generally, in order to get the most accurate overall situation through sample analysis, it is always possible to sample the population to be tested several times. At present, some commonly used test standards are used to verify the randomness of bit streams, such as the Federal Information Processing Standard (FIPS 140.1), the National Institute of Standards and Technology (NIST 800.22), the ENT test suite and the DIEHARD test suite. NIST 800.22 can handle large-size bit streams which contains 15 test methods: frequency test, run test, serial test, etc. The significance level in NIST 800.22 is preset as $\alpha = 0.01$ and the sample size of the binary sequence in each test should not be less than α^{-1}. Each sub-test result is represented by a $p - value$. The test results can be explained in three ways: $p - value$, pass rate and $p - value_T$. The interpretation of $p - value$ is to test whether the generated $p - value$ is in the range of [0.01, 1]. Pass-through rate represents the proportion of the sequence passing the $p - value$ test in all test sequences. $p - value_T$ is the statistics of $p - value$ distribution. If $p - value_T \geq 0.0001$, the sequences can be considered to be uniformly distributed. NIST 800.22 standard has a lot of test times and is complex, so we choose NIST 800.22 statistical test suite to test the bit stream generated by the proposed PRNG. The bit stream obtained from chaotic PRNG based on FPGA is tested by NIST using 1000 sample sequences of 1Mbit. As shown in Table 4, all the statistical tests of PRNG based on FPGA prove that the designed PRNG can be used in real stochastic systems. The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately 0.98.

The ENT test set consists of five statistical tests to evaluate the PRNG which applies various tests to byte sequences stored in files and reports the results of these tests. For ENT test suite, if the results of a numerical sequence deviate too far from the ideal values listed in Table 5, it will fail.

TABLE 4. **NIST test results for FPGA-based PRNG.**

NIST Statistical test	$p - value_T$	Proportion	Result
Frequency (monobit) test	0.870856	0.986	Successful
Frequency test within a block	0.713641	0.986	Successful
Runs test	0.444691	0.983	Successful
Test for the longest run of ones in a block	0.903338	0.988	Successful
Binary matrix rank test	0.266235	0.991	Successful
Discrete Fourier transform test	0.161703	0.984	Successful
Non-overlapping template matching test	0.543623	0.989331	Successful
Overlapping template matching test	0.544254	0.987	Successful
Maurer's universal statistical test	0.647530	0.985	Successful
Linear complexity test	0.228367	0.984	Successful
Serial test 1	0.469232	0.986	Successful
Serial test 2	0.236810	0.991	Successful
Approximate entropy test	0.476911	0.986	Successful
Cumulative sums test	0.314544/0.886162	0.984/0.987	Successful
Random-excursions test	0.729608375	0.991653	Successful
Random-excursions variant test	0.3062010556	0.992580	Successful

TABLE 5. **ENT test results for FPGA-based PRNG.**

Test name	Test output	Ideal value	Result
Entropy	7.999979	8.0	Success
Chi-Square	56.78%	50%	Success
Arithmetic Mean	127.5240	127.5	Success
Monte Carlo Value for Pi	3.143354182	3.141592653	Success
Serial Correlation Coefficient	-0.000129	0	Success

TABLE 6. **AIS.31 test results for FPGA-based PRNG.**

Test name	Acceptance range	Value	Result
T0 (Disjointness)	—	Pass	Pass
T1 (Monobit)	9654 < value < 10,346	10158	Pass
T2 (Poker)	1.03 < value < 57.4	20.1408	Pass
T3 (Run)	runs 1:2267-2733	2467	Pass
T3 (Run)	runs 2:1079-1421	1270	Pass
T3 (Run)	runs 3:502-748	628	Pass
T3 (Run)	runs 4:233-402	300	Pass
T3 (Run)	runs 5:90-223	155	Pass
T3 (Run)	runs 6+:90-223	160	Pass
T4 (Longrun)	Long Run=34	Pass	Pass
T5 (Autocorrelation)	2326 < value < 2674	2511	Pass

Table 5 shows the results of randomness test, and the proposed PRNs can successfully pass all tests.

In AIS.31 test, RNG is divided into two levels: P1 and P2, in which P2 requires higher requirements and is downward compatible with P1. According to the requirements of AIS.31 standard, the internal random sequence must be able to pass the P1 class test, which refers to the internal RNs passing six tests. The test methods applied to P1 include 6 tests: T0 to T5. For tests from T1 to T5 (excluding T0), the length of the sample sequence is set to 20000 bits and 257 rounds are conducted respectively. If internal RNs pass the T0 to T5 test, it is determined that internal RNs belong to P1. The results of internal RNs passing P1 class test are listed in Table 6.

In Table 7, the proposed PRNG is compared with RNGs in recent literature. It can be seen that in the proposed PRNG, the output speed can reach 62.5 Mbits/s when the maximum clock of RNG implemented by FPGA is 135.04 MHz using FWMHS and Bernoulli map as dual entropy sources. Compared with the methods proposed in the table, the proposed PRNG runs faster and can be applied to various embedded systems based on chaos, including cryptography and secure communication.

V. SECURITY ANALYSIS

A. DYNAMICAL DEGRADATION

In real life, there are always some deviations between the chaotic system we study and the real chaotic system. No matter how high-precision computer is used, it can not accurately simulate every state of the chaotic system. For chaotic sequences, the chaotic systems used are all realized with limited precision. Limited by the limited precision, the chaotic sequence will show the chaotic degradation behavior, which makes the periodicity and pseudorandomness of the sequence worse, so it is not suitable for security encryption and this phenomenon is called the limited precision effect. So the researchers naturally think of improving the realization accuracy of chaotic system to avoid its dynamical degradation. In 1988, Martelli *et al.* [95] discovered the relationship between the average cycle length T of the periodic orbit of the system and the realization accuracy L, that is $T \sim \varepsilon^{-d/2}$, where $\varepsilon = 2^{-L}$, d is the correlation dimension of the chaotic attractor. It can be seen that the average length of system orbit period will increase exponentially with the increase of calculation accuracy. But there are a lot of orbits with cycle length less than the average cycle length in the system at the same time, so improving the accuracy can only improve the dynamical degradation phenomenon, which can not be completely eradicated. Although it can't solve the problem of limited precision and get the ideal pseudo chaos sequence at once, it can be used to improve the security of the algorithm in practical application.

B. KEY SPACE

In order to protect the confidentiality of information against cryptanalysis, it is very important to choose the size of key space. The larger the key space, the higher the

TABLE 7. Comparison of the proposed design with others in the literature.

Refs.	Types	Entropy source	FPGA Chip	Output bit rate (Mbit/s)	Max. Clock frequency (MHz)	Post processing	Test suite
[20]	Pseudo	Bernoulli map	Spartan 3E	7.380	36.90	XOR	NIST 800.22
[20]	Pseudo	Tent map	Spartan 3E	6.652	33.26	XOR	NIST 800.22
[20]	Pseudo	Zigzag map	Spartan 3E	6.266	31.33	XOR	NIST 800.22
[53]	True	Bernoulli map	Spartan XC3S1600E Anadigm AN231E04	1.5	—	—	NIST 800.22
[54]	True	Chaotic System	Virtex-6	58.7	293	XOR	NIST 800.22 FIPS 140.1
[55]	True	memristive canonical Chua's oscillator and logistic map	Kintex-7	0.125	59.492	XOR	NIST 800.22
This work	Pseudo	four-wing memristive hyperchaotic system and Bernoulli map	ZYNQ-XC7Z020	62.5	135.04	XOR	NIST 800.22 ENT AIS.31

encryption intensity, and the more suitable for information encryption. Otherwise, too small key space is vulnerable to exhaustive attack, so the key password is deciphered. In this paper, the PRNG is constructed by using continuous high-dimensional chaotic system and discrete mapping to increase the required key space. The high-dimensional chaotic system has many parameters, is sensitive to the boundary conditions and the initial value of the system, and has a large relative key space, which will have a better application prospect in information encryption than the low-dimensional chaotic system.

In most PRNGs using chaotic mapping in continuous space, the key space depends on the precision of floating-point number. According to IEEE floating point standard [96], the key consists of 15 16-bit single precision floating-point numbers of FWMHS initial conditions $\{x(0), y(0), z(0), w(0)\}$ and system parameters $\{a, b, c, d, e, f, g, m, n, p, Q\}$, and 2 16-bit single precision floating-point numbers of Bernoulli map initial condition $x_{n+1}(0)$ and parameter B. In this method, the keys ares 240 bits. That is to say, the key spaces of this method are 2^{240}, which is much larger than 2^{128}, so it can effectively resist exhaustive attack.

C. KEY SENSITIVITY

Chaos is very sensitive to the initial conditions, so the PRNG based on chaos should also be sensitive to the key to produce good random number. In this test, we give an original key as a benchmark key to generate a pseudo-random sequence of 1000 bits. We change the initial condition $x(0)$ and the value of parameter a slightly by 10^{-8}.

(1) $x(0) = 0.1$: a sequence S_1 with 1000 bits is generated, then a new sequence S_2 by slight modification of the initial condition $x'(0) = 0.1 \times 10^{-8}$ is generated;

(2) $a = 0.25$: a sequence S_3 with 1000 bits is generated, then a new sequence S_4 by slight modification of the system parameter $a' = 0.25 \times 10^{-8}$ is generated.

In this test, the bit change rate can be used to measure its sensitivity to the key [97]–[99], that is, when the key changes slightly, the number of bits in the sequence generated by the PRNG is different. The ideal bit change rate is 50% and the closer the simulation result is to 50%, the better the sensitivity of PRNG to initial value is. If the length of two pseudo-random sequences S_1 and S_2 with different initial values is n, then the corresponding bit change rate is defined as:

$$P = \frac{\sum_{t=1}^{n} (S_{1t} - S_{2t})}{n} \times 100\% \tag{7}$$

where S_{1t} and S_{2t} are the t-bit values of chaotic pseudorandom S_1 and S_2 respectively. The change of bit change rate P with initial value and parameter change $\Delta i (i = x, a)$ of the system is shown in Table 8. It can be seen that when the initial value and parameters of the system are only small changes of 10^{-8}, the bit change rate of the pseudo-random sequence is very close to the ideal 50%, which shows that the PRNG is very sensitive to the initial values and parameters of the chaotic system and mapping.

TABLE 8. Initial value sensitivity analysis of pseudorandom sequence.

—	Δx	Δa	P
S_2	10^{-8}	0	49.99%
S_4	0	10^{-8}	49.99%

Fig. 12 (a) is the time domain waveforms of x when the initial conditions are $x(0) = 0.1$ and $x(0) = 0.10000000001$. Fig. 12(b) shows the time domain waveforms of x of system parameter $a = 0.25$ and $a = 0.25000001$. It can be seen that with the increase of iteration times, the sensitivity of initial value and parameter becomes more and more obvious.

D. CORRELATION

Correlation is an important method to measure the randomness of two sequences generated by adjacent keys. In order to further verify the sensitivity of the design method to the initial key, the correlation between the two sequences generated by the similar key is observed. The correlation coefficient measures the statistical relationship between the two pseudo-random sequences, and then uses the correlation coefficient to carry on the simulation test. If the generated sequence is random, its autocorrelation graph is δ function, and the autocorrelation graph should be all zero. For two

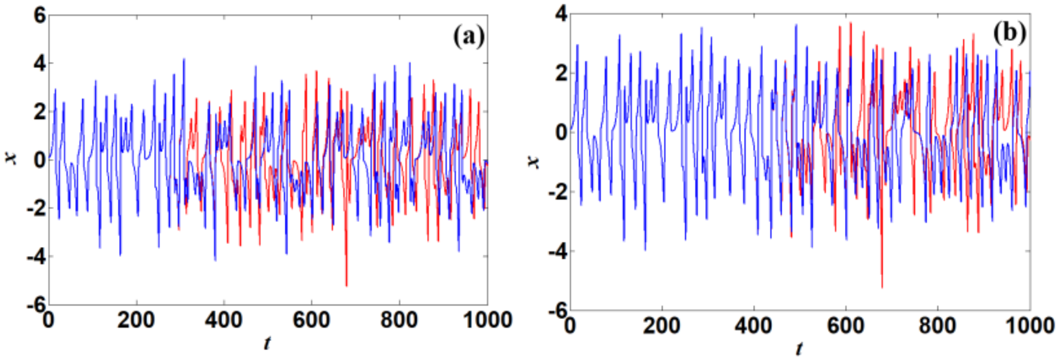


FIGURE 12. The time domain waveforms of x when: (a) initial conditions are $x(0) = 0.1$ (blue) and $x(0) = 0.10000000001$ (red), (b) system parameter $a = 0.25$ (blue) and $a = 0.25000001$ (red).

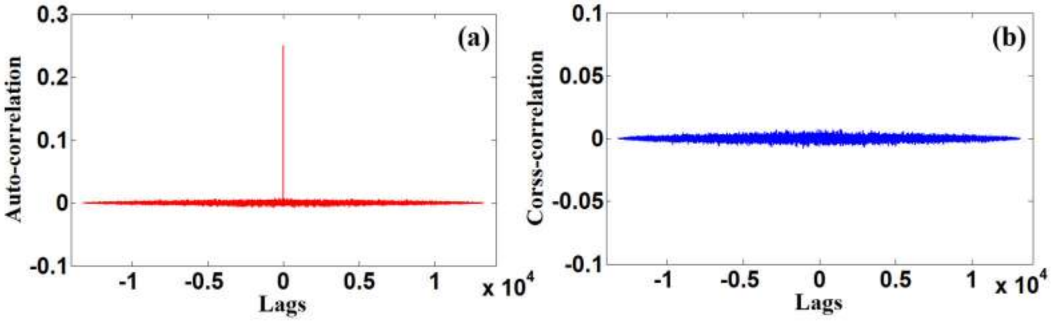


FIGURE 13. Correlation analysis of 10 sequences for PRNG-FWMHS: (a) Auto-correlation, (b) Cross-correlation.

sequences $X = \{x_0, x_1, \cdots, x_{n-1}\}$ and $Y = \{y_0, y_1, \cdots, y_{n-1}\}$, the correlation between the two sequences can be expressed as follows:

$$C_{XY} = \frac{\sum_{i=0}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=0}^{N}(y_i - \bar{y})^2}} \tag{8}$$

where $\bar{x} = \sum_{i=0}^{N} \frac{x_i}{N}, \bar{y} = \sum_{i=0}^{N} \frac{y_i}{N}$, N is the length of the sequence. $C_{XY} \in (-1, 1)$, Here we calculated that $C_{XY} = 1.98 \times 10^{-4}$, then we can assume that there is no correlation between X and Y, therefore, the sensitivity of chaotic system to small changes of parameters is high.

Fig. 13 shows the correlation between the pseudo-random sequence generated by the original key and the 10 pseudo-random sequences generated by the randomly selected 10 keys. The uniform results of 10 experiments close to 0 verify that there is no correlation between the pseudo-random sequences generated by this method.

E. INFORMATION ENTROPY

Information entropy determines the unpredictability of some messages and it is a measure of the uncertainty of the random bit stream generated by our proposed structure. When the random bit stream is uniformly distributed, the entropy is the largest. It is very important in safety analysis, high entropy means a robust pseudorandom generator, while low entropy means a weak pseudorandom generator with certain predictability. The entropy $H(x)$ of a sequence x can be estimated using the following expression

$$H(x) = -\sum_{i=1}^{2^N} p(x_i)\log_2 p(x_i) \tag{9}$$

where N is the number of bits of each element of sequence x, 2^N is all possible symbols in the sequence, x_i is 2^N different symbols in sequence x, $p(x_i)$ represents the probability of distribution of element x_i in sequence x. If sequence x has 2^N possible elements, the entropy should be $H(x) = N$. Fig. 14 shows the entropy of 51 random sequences, and the average entropy of 51 random sequences is 0.9998. Therefore, the sequence is unpredictable.

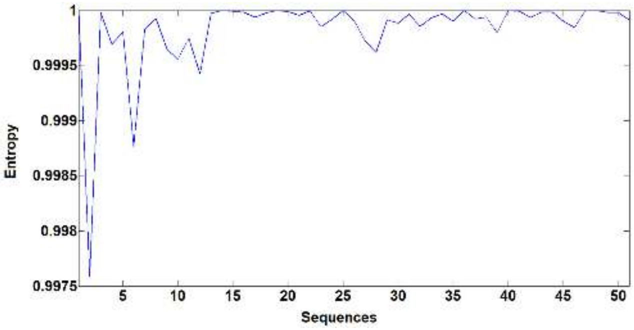


FIGURE 14. Information entropy of 51 sequences for PRNG.

VI. CONCLUSION

In this article, first of some dynamic behaviours of a FWMHS with no-equilibrium points and Bernoulli map were basically analyzed. Then, the FWMHS has been numerically modeled by VHDL using RK4 algorithm, one of the most popular numerical differential equation decryption methods in literature. Additionally, oscilloscope screen image obtained from FPGA has confirmed the results obtained from numerical model. Finally, a new PRNG based on the combination of a dual entropy sources architecture with FWMHS and Bernoulli map was discussed. PRNG has been implemented on FPGA and proved to be able to generate random bit streams with an output bit rate of 62.5 Mbit/s. Statistical tests and security analysis show that the binary sequences have good pseudorandom characteristics and it has been proved that the design can be used in cryptographic applications.

REFERENCES

[1] K. Xie, X. Ning, X. Wang, S. He, Z. Ning, X. Liu, J. Wen, and Z. Qin, ''An efficient privacy-preserving compressive data gathering scheme in WSNs,'' *Inf. Sci.*, vol. 390, no. 2, pp. 702–715, 2016.

[2] K. Gu, W. Jia, G. Wang, and S. Wen, ''Efficient and secure attribute-based signature for monotone predicates,'' *Acta Inf.*, vol. 54, no. 5, pp. 521–541, Aug. 2017.

[3] M. Long, F. Peng, and H.-Y. Li, ''Separable reversible data hiding and encryption for HEVC video,'' *J. Real-Time Image Process.*, vol. 14, no. 1, pp. 171–182, 2018.

[4] L. Xiang, X. Shen, J. Qin, and W. Hao, ''Discrete multi-graph hashing for large-scale visual search,'' *Neural Process. Lett.*, vol. 49, no. 3, pp. 1055–1069, Jul. 2018.

[5] K. Gu, N. Wu, and B. Yin, ''Secure data sequence query framework based on multiple fogs,'' *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2019.2943524.

[6] S. He, W. Zeng, and K. Xie, ''PPNC: Privacy preserving scheme for random linear network coding in smart grid,'' *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 3, pp. 1510–1533, 2017.

[7] Z. Xia, Z. Fang, and F. Zou, ''Research on defensive strategy of real-time price attack based on multiperson zero-determinant,'' *Secur. Commun. Netw.*, vol. 2019, Jul. 2019, Art. no. 6956072, doi: 10.1155/2019/6956072.

[8] K. Gu, N. Wu, B. Yin, and W. J. Jia, ''Secure data query framework for cloud and fog computing,'' *IEEE Trans. Netw. Service Manage.*, to be published, doi: 10.1109/TNSM.2019.2941869.

[9] M. Long, F. Peng, and Y. Zhu, ''Identifying natural images and computer generated graphics based on binary similarity measures of PRNU,'' *Multimedia Tools Appl.*, vol. 78, no. 1, pp. 489–506, Jan. 2017.

[10] K. Gu, X. Dong, and L. Wang, ''Efficient traceable ring signature scheme without pairings,'' *Adv. Math. Commun.*, to be published, doi: 10.3934/amc.2020016.

[11] G. Iovane, A. Amorosia, E. Benedetto, and G. Lamponi, ''An information fusion approach based on prime numbers coming from RSA algorithm and fractals for secure coding,'' *J. Discrete Math. Sci. Cryptogr.*, vol. 18, no. 5, pp. 455–479, 2015.

[12] G. Horng, ''Accelerating DSA signature generation,'' *Cryptologia*, vol. 39, no. 2, pp. 121–125, 2015.

[13] M. Long and Y. Chen, ''Average throughput and BER analysis for energy harvesting communications,'' *IET Commun.*, vol. 13, no. 3, pp. 289–296, 2019.

[14] F. Peng, X. Zhang, and Z. X. Lin, ''A tunable selective encryption scheme for H.265/HEVC based on chroma IPM and coefficient scrambling,'' *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2019.2924910.

[15] M. A. Murillo-Escobar, C. Cruz-Hernández, L. Cardoza-Avendaño, and R. Méndez-Ramírez, ''A novel pseudorandom number generator based on pseudorandomly enhanced logistic map,'' *Nonlinear Dyn.*, vol. 87, no. 1, pp. 407–425, Jan. 2017.

[16] G. D. P. Stanchieri, A. De Marcellis, and E. Palange, and M. Faccio, ''A true random number generator architecture based on a reduced number of FPGA primitives,'' *AEU-Int. J. Electron. Commun.*, vol. 105, pp. 15–23, Jun. 2019.

[17] R. S. Hasan, S. K. Tawfeeq, N. Q. Mohammed, and A. I. Khaleel, ''A true random number generator based on the photon arrival time registered in a coincidence window between two single-photon counting modules,'' *Chin. J. Phys.*, vol. 56, no. 1, pp. 385–391, Feb. 2018.

[18] F. Yu, L. Li, Q. Tang, S. Cai, Y. Song, and Q. Xu, ''A survey on true random number generators based on chaos,'' *Discrete Dyn. Nature Soc.*, vol. 2019, Dec. 2019, Art. no. 2545123.

[19] M. Park, J. C. Rodgers, and D. P. Lathrop, ''True random number generation using CMOS Boolean chaotic oscillator,'' *Microelectron. J.*, vol. 46, no. 12, pp. 1364–1370, 2015.

[20] L. G. de la Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas-López, ''Hardware implementation of pseudo-random number generators based on chaotic maps,'' *Nonlinear Dyn.*, vol. 90, no. 3, pp. 1661–1670, Nov. 2017.

[21] F. Yu, L. Gao, K. Gu, B. Yin, Q. Wan, and Z. Zhou, ''A fully qualified four-wing four-dimensional autonomous chaotic system and its synchronization,'' *Optik*, vol. 131, pp. 79–88, Feb. 2017.

[22] X. Zhang and C. Wang, ''A novel multi-attractor period multi-scroll chaotic integrated circuit based on CMOS wide adjustable CCCII,'' *IEEE Access*, vol. 7, no. 1, pp. 16336–16350, 2019.

[23] F. Yu, P. Li, K. Gu, and B. Yin, ''Research progress of multi-scroll chaotic oscillators based on current-mode devices,'' *Optik*, vol. 127, no. 3, pp. 5486–5490, 2016.

[24] J. Jin, ''Programmable multi-direction fully integrated chaotic oscillator,'' *Microelectron. J.*, vol. 75, pp. 27–34, May 2018.

[25] X. Zhang, C. Wang, W. Yao, and H. Lin, ''Chaotic system with bondorbital attractors,'' *Nonlinear Dyn.*, vol. 97, no. 4, pp. 2159–2174, 2019.

[26] J. Jin and L. V. Zhao, ''Low voltage low power fully integrated chaos generator,'' *J. Circuits Syst. Comput.*, vol. 27, no. 10, p. 1850155, Sep. 2018.

[27] X. Zhang and C. Wang, ''Multiscroll hyperchaotic system with hidden attractors and its circuit implementation,'' *Int. J. Bifurcation Chaos*, vol. 29, no. 9, p. 1950117, 2019.

[28] Y. Y. Huang, Y. H. Wang, and H. G. Chen, ''Shape synchronization control for three-dimensional chaotic systems,'' *Chaos Solitons Fractals*, vol. 87, pp. 136–145, Jun. 2016.

[29] L. Zhou, F. Tan, F. Yu, and W. Liu, ''Cluster synchronization of two-layer nonlinearly coupled multiplex networks with multi-links and time-delays,'' *Neurocomputing*, vol. 359, pp. 264–275, Sep. 2019.

[30] Y.-Y. Huang, Y.-H. Wang, and Y. Zhang, ''Shape synchronization of drive-response for a class of two-dimensional chaotic systems via continuous controllers,'' *Nonlinear Dyn.*, vol. 78, no. 4, pp. 2331–2340, 2014.

[31] F. Yu and Y. Song, ''Complete switched generalized function projective synchronization of a class of hyperchaotic systems with unknown parameters and disturbance inputs,'' *J. Dyn. Syst., Meas., Control-Trans.*, vol. 136, no. 1, p. 014505, 2014.

[32] F. Yu, C. Wang, Q. Wan, and Y. Hu, ''Complete switched modified function projective synchronization of a five-term chaotic system with uncertain parameters and disturbances,'' *Pramana-J. Phys.*, vol. 80, no. 2, pp. 223–235, 2013.

[33] F. Yu and C. Wang, ''Secure communication based on a four-wing chaotic system subject to disturbance inputs,'' *Optik*, vol. 125, no. 20, pp. 5920–5925, 2014.

[34] L. Zhou, C. Wang, H. He, and Y. Lin, ''Time-controllable combinatorial inner synchronization and outer synchronization of anti-star networks and its application in secure communication,'' *Commun. Nonlinear Sci. Numer. Simul.*, vol. 22, nos. 1–3, pp. 623–640, 2015.

[35] F. Yu, L. Liu, B. He, Y. Huang, C. Shi, S. Cai, Y. Song, S. Du, and Q. Wan, ''Analysis and FPGA realization of a novel 5D hyperchaotic four-wing memristive system, active control synchronization and secure communication application,'' *Complexity*, vol. 2019, Nov. 2019, Art. no. 4047957, doi: 10.1155/2019/4047957.

[36] G. Xu, J. Xu, C. Xiu, F. Liu, and Y. Zang, ''Secure communication based on the synchronous control of hysteretic chaotic neuron,'' *Neurocomputing*, vol. 227, pp. 108–112, Mar. 2017.

[37] L. Zhou and F. Tan, ''A chaotic secure communication scheme based on synchronization of double-layered and multiple complex networks,'' *Nonlinear Dyn.*, vol. 96, no. 2, pp. 869–883, 2019.

[38] L. Zhou, F. Tan, and F. Yu, ''A robust synchronization-based chaotic secure communication scheme with double-layered and multiple hybrid networks,'' *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2019.2927495.

[39] M. I. Mihailescu, ''New enrollment scheme for biometric template using hash chaos-based cryptography,'' *Procedia Eng.*, vol. 69, pp. 1459–1468, Mar. 2014.

[40] V. Petrauskiene, R. Palivonaite, A. Aleksa, and M. Ragulskis, ''Dynamic visual cryptography based on chaotic oscillations,'' *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 1, pp. 112–120, 2014.

[41] S. Behnia, A. Akhshani, S. Ahadpour, A. Akhavan, and H. Mahmodi, ''Cryptography based on chaotic random maps with position dependent weighting probabilities,'' *Chaos, Solitons Fractals*, vol. 40, no. 1, pp. 362–369, 2009.

[42] S. Deng, D. Xiao, Y. Li, and W. Peng ''A novel combined cryptographic and hash algorithm based on chaotic control character,'' *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, no. 11, pp. 3889–3900, 2009.

[43] C. Wang, L. Xiong, J. Sun, and W. Yao, ''Memristor-based neural networks with weight simultaneous perturbation training,'' *Nonlinear Dyn.*, vol. 95, no. 4, pp. 2893–2906, 2019.

[44] F. Yu, L. Liu, L. Xiao, K. Li, and S. Cai ''A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function,'' *Neurocomputing*, vol. 350, pp. 108–116, Jul. 2019.

[45] L. Zhou, C. Wang, S. Du, and L. Zhou, ''Cluster synchronization on multiple nonlinearly coupled dynamical subnetworks of complex networks with nonidentical nodes,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 570–583, Mar. 2017.

[46] L. Zhou, C. Wang, and L. Zhou, ''Cluster synchronization on multiple sub-networks of complex networks with nonidentical nodes via pinning control,'' *Nonlinear Dyn.*, vol. 83, nos. 1–2, pp. 1079–1100, 2016.

[47] W. Yao, C. Wang, J. Cao, Y. Sun, and C. Zhou, ''Hybrid multisynchronization of coupled multistable memristive neural networks with time delays,'' *Neurocomputing*, vol. 363, pp. 281–294, Oct. 2019.

[48] Q. Yin and C. Wang, ''A new chaotic image encryption scheme using breadth-first search and dynamic diffusion,'' *Int. J. Bifurcation Chaos*, vol. 28, no. 4, p. 1850047, May 2018.

[49] Y. Lin, C. H. Wang, and H. Xu, ''Grid multi-scroll chaotic attractors in hybrid image encryption algorithm based on current conveyor,'' *Acta Phys. Sinica*, vol. 61, no. 24, pp. 514–518, 2012.

[50] W. Feng, Y.-G. He, H.-M. Li, and C.-L. Li, ''Cryptanalysis of the integrated chaotic systems based image encryption algorithm,'' *Optik*, vol. 186, pp. 449–457, Jun. 2019.

[51] Y. Li, C. Wang, and H. Chen, ''A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation,'' *Opt. Lasers Eng.*, vol. 90, pp. 238–246, Mar. 2017.

[52] G. Cheng, C. Wang, and H. Chen, ''A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture,'' *Int. J. Bifurcation Chaos*, vol. 29, no. 9, 2019, Art. no. 1950115.

[53] I. Cicek, A. E. Pusane, and G. Dundar, ''A new dual entropy core true random number generator,'' *Analog Integr. Circuits Signal Process.*, vol. 81, no. 1, pp. 61–70, 2014.

[54] . Koyuncu and A. T. Özcerit, ''The design and realization of a new high speed FPGA-based chaotic true random number generator,'' *Comput., Elect. Eng.*, vol. 58, pp. 203–214, Feb. 2017.

[55] B. Karakaya, A. Gülten, and M. Frasca, ''A true random bit generator based on a memristive chaotic circuit: Analysis, design and FPGA implementation,'' *Chaos, Solitons Fractals*, vol. 119, pp. 143–149, Feb. 2019.

[56] Z. Li, P. Li, Y. Mao, and W. A. Halang, ''Chaos-based pseudo-random number generators and chip implementation,'' *IFAC Proc.*, vol. 38, no. 1, pp. 1090–1094, 2005.

[57] C.-Y. Li, Y.-H. Chen, T.-Y. Chang, L.-Y. Deng, and K. To, ''Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG,'' *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 385–389, Feb. 2012.

[58] F. Yu, Q. Tang, W. Wang, and H. Wu, ''A 2.7 GHz low-phase-noise LC-QVCO using the gate-modulated coupling technique,'' *Wireless Pers. Commun.*, vol. 86, no. 2, pp. 671–681, 2016.

[59] F. Yu, ''A low-voltage and low-power 3-GHz CMOS LC VCO for S-band wireless applications,'' *Wireless Pers. Commun.*, vol. 78, no. 2, pp. 905–914, 2014.

[60] J. Jin and M. Tan, ''Low power quadrature voltage controlled oscillator,'' *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 29, no. 12, p. e21952, 2019, doi: 10.1002/mmce.21952.

[61] Q. Wan, J. Dong, H. Zhou, and F. Yu, ''A very low power quadrature VCO with modified current-reuse and back-gate coupling topology,'' *J. Circuits, Syst. Comput.*, vol. 26, no. 1, p. 1750184, 2017.

[62] S. Cai, W. Wang, F. Yu, and B. He, ''Single event transient propagation probabilities analysis for nanometer CMOS circuits,'' *J. Electron. Test.-Theory Appl.*, vol. 35, no. 2, pp. 163–172, 2019.

[63] F. Yu, L. Gao, and L. Liu, ''A 1 V, 0.53 ns, 59 μW current comparator using standard 0.18 μm CMOS technology,'' *Wireless Pers. Commun.*, pp. 1–9, 2019, doi: 10.1007/s11277-019-06888-9.

[64] M. Bakiri, C. Guyeux, J.-F. Couchot, and A. K. Oudjida, ''Survey on hardware implementation of random number generators on FPGA: Theory and experimental analyses,'' *Comput. Sci. Rev.*, vol. 27, pp. 135–153, Feb. 2018.

[65] A. A. Rezk, A. H. Madian, A. G. Radwan, and A. M. Soliman, ''Reconfigurable chaotic pseudo random number generator based on FPGA,'' *AEU-Int. J. Electron. Commun.*, vol. 98, pp. 174–180, Jan. 2019.

[66] M. O. Meranza-Castillón, M. A. Murillo-Escobar, R. M. López-Gutiérrez, and C. Cruz-Hernández, ''Pseudorandom number generator based on enhanced Hénon map and its implementation,'' *AEU-Int. J. Electron. Commun.*, vol. 107, pp. 239–251, Jul. 2019.

[67] J. Jin and L. Cui, ''Fully integrated memristor and its application on the scroll-controllable hyperchaotic system,'' *Complexity*, vol. 2019, Jan. 2019, Art. no. 4106398, doi: 10.1155/2019/4106398.

[68] F. Yu, C. Wang, and H. He, ''Grid multiscroll hyperchaotic attractors based on Colpitts oscillator mode with controllable grid gradient and scroll numbers,'' *J. Appl. Res. Technol.*, vol. 11, no. 3, pp. 371–380, 2013.

[69] Z. Wan, C. Wang, X. Luo, Y. Lin, and T. Huang, ''Generating variable number of wings from a novel four-dimensional hyperchaotic system with one equilibrium,'' *Optik*, vol. 125, no. 3, pp. 1371–1376, 2014.

[70] F. Yu, C. H. Wang, Y. Hu, and J. W. Yin, ''Antisynchronization of a novel hyperchaotic system with parameter mismatch and external disturbances,'' *Pramana-J. Phys.*, vol. 79, no. 1, pp. 81–93, 2012.

[71] Y. Liu and X. Tong, ''Hyperchaotic system-based pseudorandom number generator,'' *IET Inf. Secur.*, vol. 10, no. 6, pp. 433–441, 2016.

[72] M. García-Martínez, L. J. Ontañón-García, E. Campos-Cantón, and S. Čelikovský, ''Hyperchaotic encryption based on multi-scroll piecewise linear systems,'' *Appl. Math. Comput.*, vol. 270, pp. 413–424, Nov. 2015.

[73] L. O. Chua, ''Memristor-the missing circuit element,'' *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.

[74] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, ''The missing memristor found,'' *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[75] Q. Zhao, C. Wang, and X. Zhang, ''A universal emulator for memristor, memcapacitor, and meminductor and its chaotic circuit,'' *Chaos*, vol. 29, no. 1, p. 013141, 2019.

[76] L. Zhou, C. Wang, X. Zhang, and W. Yao, ''Various attractors, coexisting attractors and antimonotonicity in a simple fourth-order memristive twin-T oscillator,'' *Int. J. Bifurcation Chaos*, vol. 28, no. 4, p. 1850050, 2018.

[77] Q. Xu, Z. Song, H. Bao, M. Chen, and B. Bao, ''Two-neuron-based non-autonomous memristive Hopfield neural network: Numerical analyses and hardware experiments,'' *AEU-Int. J. Electron. Commun.*, vol. 96, pp. 66–74, Nov. 2018.

[78] L. Zhou, C. Wang, X. Zhang, and W. Zhang, ''Coexisting attractors and antimonotonicity in a simple fourth-order memristive twin-T oscillator,'' *Int. J. Bifurcation Chaos*, vol. 28, no. 4, p. 1850050, 2018.

[79] Q. Xu, Y. Lin, B. Bao, and M. Chen, ''Multiple attractors in a non-ideal active voltage-controlled memristor based Chua's circuit,'' *Chaos, Solitons Fractals*, vol. 83, pp. 186–200, Feb. 2016.

[80] C. Wang, X. Liu, and H. Xia, ''Multi-piecewise quadratic nonlinearity memristor and its $2N$-scroll and $2N+1$-scroll chaotic attractors system,'' *Chaos*, vol. 27, no. 3, p. 033114, 2017.

[81] L. Zhou, C. Wang, and L. Zhou, ''A novel no-equilibrium hyperchaotic multi-wing system via introducing memristor,'' *Int. J. Circuit Theory Appl.*, vol. 46, no. 1, pp. 84–98, Jan. 2018.

[82] L. Zhou, C. Wang, and L. Zhou, ''Generating four-wing hyperchaotic attractor and two-wing, three-wing, and four-wing chaotic attractors in 4D memristive system,'' *Int. J. Bifurcation Chaos*, vol. 27, no. 2, p. 1750027, Feb. 2017.

[83] F. Yu, Z. Zhang, L. Liu, H. Shen, Y. Huang, C. Shi, S. Cai, Y. Song, S. Du, and Q. Xu, ''Secure communication scheme based on a new 5D multistable four-wing memristive hyperchaotic system with disturbance inputs,'' *Complexity*, vol. 2019, Dec. 2019, Art. no. 5859273.

[84] C. Wang, H. Xia, and L. Zhou, ''A memristive hyperchaotic multiscroll jerk system with controllable scroll numbers,'' *Int. J. Bifurcation Chaos*, vol. 27, no. 6, 2017, Art. no. 1750091.

[85] L. Zhou, C. H. Wang, and L. Zhou, ''Generating hyperchaotic multi-wing attractor in a 4D memristive circuit,'' *Nonlinear Dyn.*, vol. 85, no. 4, pp. 2653–2663, Sep. 2016.

[86] B. A. Mezatio, M. T. Motchongom, B. R. W. Tekam, R. Kengne, R. Tchitnga, and A. Fomethe, ''A novel memristive 6D hyperchaotic autonomous system with hidden extreme multistability,'' *Chaos, Solitons Fractals*, vol. 120, pp. 100–115, May 2019.

[87] R.-E. Karamani, V. Ntinas, and I. Vourkas, "1-D memristor-based cellular automaton for pseudo-random number generation," in *Proc. 27th Int. Symp. Power Timing Modeling Optim. Simulation (PATMOS)*, Thessaloniki, Greece, Sep. 2017, pp. 1–6.

[88] K. Rajagopal, A. Bayani, and A. J. M. Khalaf, H. Namazi, S. Jafari, and V.-T. Pham, "A no-equilibrium memristive system with four-wing hyperchaotic attractor," *AEU-Int. J. Electron. Commun.*, vol. 95, pp. 207–215, Oct. 2018.

[89] J. L. Zhang, W. Z. Wang, X. W. Wang, and Z.-H. Xia, "Enhancing security of FPGA-based embedded systems with combinational logic binding," *J. Comput. Sci. Technol.*, vol. 32, no. 2, pp. 329–339, 2017.

[90] C. A. Lua, S. Di Gennaro, A. N. Guzman, S. Ortega-Cisneros, and J. R. Domínguez, "Digital Implementation via FPGA of controllers for active control of ground vehicles," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2253–2264, Apr. 2019.

[91] Q. Tang, K. Yang, D. Zhou, Y. Luo, and F. Yu, "A real-time dynamic pricing algorithm for smart grid with unstable energy providers and malicious users," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 554–562, Aug. 2016.

[92] L. D. Medus, T. Iakymchuk, J. V. Frances-Villora, M. Bataller-Mompeán, and A. Rosado-Muñoz, "A novel systolic parallel hardware architecture for the FPGA acceleration of feedforward neural networks," *IEEE Access*, vol. 7, pp. 76084–76103, 2019.

[93] M. Meribout, I. M. Saied, and E. A. Hosani, "A new FPGA-based terahertz imaging device for multiphase flow metering," *IEEE Trans. THz Sci. Technol.*, vol. 8, no. 4, pp. 418–426, Jul. 2018.

[94] M. Tuna, M. Alçin, I. Koyuncu, C. B. Fidan, and I. Pehlivan, "High speed FPGA-based chaotic oscillator design," *Microprocess. Microsyst.*, vol. 66, pp. 72–80, Apr. 2019.

[95] M. Martelli, M. Dang, and T. Seph, "Defining chaos," *Math. Mag.*, vol. 71, no. 2, pp. 112–122, 1998.

[96] *IEEE Standard for Floating-Point Arithmetic*, IEEE Standards 754-2008, IEEE Computer Society, 2008, pp. 1–58.

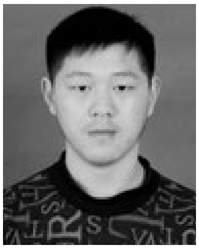
[97] S. He, K. Sun, and H. Wang, "Complexity analysis and DSP implementation of the fractional-order Lorenz hyperchaotic system," *Entropy*, vol. 17, no. 12, pp. 8299–8311, Dec. 2015.

[98] S. Ke-Hui, H. Shao-Bo, H. Yi, and Y. Lin-Zi, "Complexity analysis of chaotic pseudo-random sequences based on spectral entropy algorithm," *Acta Phys. Sinica*, vol. 62, no. 1, Jan. 2013, Art. no. 010501.

[99] C. Chen, K. Sun, and Y. Peng, "A novel control method to counteract the dynamical degradation of a digital chaotic sequence," *Eur. Phys. J. Plus*, vol. 134, p. 31, Jan. 2019.

FEI YU received the B.E. degree from Anhui Normal University, in 2007, and the M.E. and Ph.D. degrees from the College of Information Science and Engineering, Hunan University, Changsha, China, in 2010 and 2013, respectively. He is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha. His current research interests include IC design, chaotic system and nonlinear circuit, and chaos synchronization and its application.

LIXIANG LI was born in Linquan, Anhui, China, in 1995. She received the B.E. degree from the Chengnan College, Changsha University of Science and Technology, Changsha, China, in 2017, where she is currently pursuing the M.E. degree in computer and communication engineering. Her current research interest includes random number generators based on chaos.

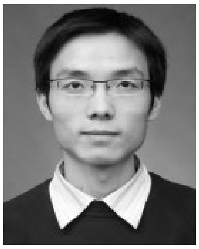


BINYONG HE was born in Hunan, China, in 1996. He received the B.Sc. degree from the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China, in 2018, where he is currently pursuing the degree. His current research interest includes IC design and test and reliability evaluation.

LI LIU received the B.Sc. degree from the School of Electronic Information, Hunan University of Information Technology, Changsha, China, in 2018. She is currently pursuing the degree with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha. Her current research interest includes nonlinear dynamics and chaos.

SHUAI QIAN was born in Hunan, China, in 1995. She received the B.Sc. degree from the School of Electronic Information Science and Technology, Xinzhou Teaching University, Xinzhou, China, in 2017. She is currently pursuing the degree with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China. Her current research interest includes random number generators based on chaos.

YUANYUAN HUANG received the B.E. degree from the University of South China, Hengyang, China, in 2002, the M.S. degree from Hunan Normal University, Changsha, China, in 2005, and the Ph.D. degree from the Guangdong University of Technology, Guangzhou, China, in 2015. Since 2005, she has been with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha. Her current research interest includes control and synchronization for complex dynamical networks and chaotic systems.



SHUO CAI received the B.E. degree from Zhejiang University, in 2004, and the M.E. and Ph.D. degrees from the College of Information Science and Engineering, Hunan University, Changsha, China, in 2007 and 2015, respectively. He is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha. His current research interests include reliability evaluation and fault-tolerant computing.

YUN SONG received the B.E. degree from Xiangtan University, in 1997, the M.E. degree from the Changsha University of Science and Technology, Changsha, China, in 2008, and the Ph.D. degree from the College of Information Science and Engineering, Hunan University, Changsha, in 2017. He is currently an Associate Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology. His current research interests include video/image process, compressed sensing, and nonlinear circuits design.

QIANG TANG received the B.E., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2005, 2007, and 2010, respectively, all in control science and engineering. He is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China. His current research interests include wireless networks and smart grid.

QIUZHEN WAN is currently a Lecturer with the College of Information Science and Engineering, Hunan Normal University, Changsha, China. His current research interests include nonlinear circuits design and RF front-end design for wireless communications.

JIE JIN received the B.Sc. degree from Shenzhen University and the M.S. degree and the Ph.D. degree in computer science and technology from Hunan University, in 2010 and 2015, respectively. His current research interests include nonlinear circuits and IC circuits design.