IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 30, 2021, accepted January 5, 2022, date of publication January 12, 2022, date of current version January 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142536

Design and Synchronization of Chaos-Based True Random Number Generators and Its FPGA Implementation

T. L. LIAO[1], (Member, IEEE), P. Y. WAN[1], AND JUN-JUH YAN[2]
[1]Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan
[2]Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41107, Taiwan

Corresponding author: Jun-Juh Yan (jjyan@ncut.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST- 110-2221-E-167-030 and Grant MOST-110-2218-E-006 -014-MBK.

ABSTRACT In this paper, a new chaos-based true random number generator (TRNG) and a sliding mode controller (SMC) to synchronize the proposed TRNGs are proposed. Firstly, the continuous chaotic system is transformed into a discrete system that preserves the original continuous system's chaotic behavior and makes it easy to realize with the field-programmable gate array (FPGA) for synchronization control. Then, a discrete SMC is introduced to solve the synchronization problem of the master-slave discrete chaotic systems. Subsequently, a novel hybrid function integrated with the El-Gamal algorithm is proposed to complete the TRNG design. Finally, the FPGA-based realization of the synchronized master-slave TRNGs is implemented. The randomness quality of the proposed TRNGs has been ensured by using Shannon's entropy and histogram analysis. Furthermore, the National Institute of Standards and Technology (NIST) test suite has also been introduced to evaluate the proposed TRNGs and compare with the existing results in the literature.

INDEX TERMS True random number generators, chaotic systems, synchronization, sliding mode control, field-programmable gate array.

I. INTRODUCTION

In cryptographic applications, the design of random number generators is crucial. Random number generators can generally be divided into two categories, namely the true random number generator (TRNG) and the pseudo-random number generator. Generally, truly random numbers cannot be predicted and controlled in advance. The TRNG usually exists in the natural world, for example, electromagnetic noise, thermal noise signals, decay radiation of radioactive elements. However, additional high-cost hardware circuits are usually required to convert or extract the true random sequences. To obtain a low-cost random number, a method of manually generating random numbers is called a pseudo-random number generator. The advantages of pseudo-random number generators are simple and low cost.

Nevertheless, its randomness might not be satisfied and unsuitable for high-security requirements compared with true random numbers. Therefore, a new low-cost TRNG is worth studying. Since the state responses of chaotic systems are random dynamic behavior, sensitivity to initial values, broadband of white noise, these characteristics provide advantages for the design of TRNGs [1], [2]. However, when it is applied to communication security, simultaneously obtaining the same random numbers at both the transmitter and receiver is also a significant issue. Concerning the chaos-based TRNG design, the works [1], [2] successfully utilized chaotic systems' random property to propose a new high-speed FPGA-based chaotic TRNG. However, they had not considered the synchronization control of TRNGs. Therefore, in the communication encryption application, the same initial values must be given for the chaotic random number generators embedded in the transmitter (master) and receiver (slave), respectively. As well known, for chaotic systems, the butterfly effect is a sensitive dependence on initial conditions, wherein a small change in initial conditions can lead to vastly different outcomes [3]. Therefore, when the chaotic systems used in the TRNGs are disturbed, and

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

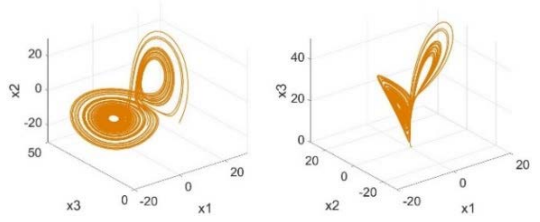

FIGURE 1. Strange attractors of continuous UCS (4).

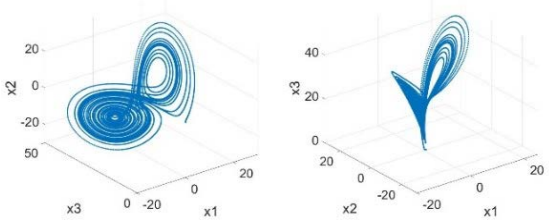

FIGURE 2. Strange attractors of discrete UCS (5).

the states become slightly different, the synchronization of random numbers will be destroyed due to the butterfly effect.

We will apply the control technology to synchronize master and slave random number generators to solve this problem, which effectively solves the above-mentioned problems. With the pioneering research of Pecora and Carroll [4], many effective control methods have been proposed for chaos synchronization, such as adaptive control, fuzzy control, sliding mode control [5] and H∞ control [6], etc. Because the sliding mode control method is insensitive to system parameters and external disturbances and has good robustness, this control approach will be applied in this paper. In [7]–[9], the randomness quality of their proposed chaos-based random numbers does not meet tests of the NIST-800-22 statistical standard, so they cannot be called true random number generators. In this paper, a novel chaos-based hybrid function integrated with the El-Gamal algorithm [10] is proposed to design low-cost TRNGs. The El-Gamal algorithm is a popular asymmetric encryption algorithm. In our design architecture, the El-Gamal algorithm acts as a hybrid function for upgrading the randomness of chaotic signals to the level of true random numbers. On the other hand, with the advancement of digital signal processing with very-large-scale integration (VLSI) technology, the computing ability and speed of the digital chips have significantly been promoted, and the price is reduced [11], [12]. FPGA is a kind of digital chip that can be programmed with the internal logic array to realize some specified functions. The advantages of FPGA are high flexibility, a short development cycle, and parallel computing function. The authors proposed novel design methods to successfully implement chaotic systems in the FPGA circuit implementation of [13], [14].

Nevertheless, there is no discussion about the design of real random numbers or the design of synchronization control for chaos-based random numbers in their research. In [15], the authors used a new 5D hyperchaotic system and implemented it on FPGA. However, it does not consider the synchronization design. In [16], the authors proposed a new type of chaotic neurons, used adaptive control to achieve synchronization, and added a combination of XOR and shift registers for post-processing to improve randomness. In this paper, following the concept of post-processing [16], a new hybrid function algorithm designed based on the El-Gamal algorithm for post-processing possesses the advantages of the El-Gamal algorithm, which increases the quality of random numbers but also improves the security of the cipher-text.

The rest of this paper is organized as follows. Section 2 describes the discretization of continuous chaotic systems In Section 3, based on the SMC, a controller is proposed to guarantee the stability of the error dynamics. In Section 4, a hybrid function based on the El-Gamal algorithm is designed for TRNGs. In Section 5, the implementation of the FPGA circuit for TRNGs is considered. Sections 6 and 7, respectively, show the chaotic behavior after the FPGA implementation and randomness test of TRNGs. Conclusions are provided in Section 8.

II. DISCRETIZATION OF CONTINUOUS CHAOTIC SYSTEMS

This paper aims to implement the synchronized chaos-based TRNGs. First, we need to discretize the continuous chaotic system into a corresponding discrete chaotic system that still preserves the random behavior like the original continuous system. We choose the continuous unified chaotic system (UCS) for our discussion for simplicity. However, the approach proposed can also be easily extended to other chaotic systems. The continuous UCS [17] is described as follows:

$$\begin{aligned}\dot{x}_1(t) &= (10 + 25w) \cdot [x_2(t) - x_1(t)] \\ \dot{x}_2(t) &= (28 - 35w)x_1(t) + (w - 1)x_2(t) - x_1(t)x_3(t) \\ \dot{x}_3(t) &= (-\frac{8}{3} - \frac{1}{3}w)x_3(t) + x_1(t)x_2(t)\end{aligned} \tag{1}$$

where $x_i(t), i = 1, 2, 3$ denotes system states and $w \in [0, 1]$ is the system parameter bridging the Lorentz attractor with the Chen attractor and the Lü attractor. For discretizing continuous chaotic systems, we consider a continuous system described by

$$\dot{X}(t) = AX(t) + Bg(X(t)) \tag{2}$$

where $X(t) \in R^n$ is the state vector, $g(X(t)) \in R^{n\times m}$ is a nonlinear vector. A and B are known constant matrices with appropriate dimensions. Then the corresponding discrete-time model of the system (2) can be obtained as:

$$X_d(k + 1)T = GX_d(kT) + Hg(X_d(kT)) \tag{3}$$

where T is the sampling time; $G = e^{AT}$, $H = [G - I_n]A^{-1}B$. Rearrange the system (1) into the form of equation (2), the

system is described as follows:

$$\underbrace{\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\end{bmatrix}}_{\dot{X}(t)} = \underbrace{\begin{bmatrix}10+25w & 10+25w & 0\\ 28-35w & w-1 & 0\\ 0 & 0 & -\frac{8}{3}-\frac{1}{3}w\end{bmatrix}}_{A} X(t) + \underbrace{\begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix}}_{B} \underbrace{\begin{bmatrix}-x_1(t)x_3(t)\\ x_1(t)x_2(t)\end{bmatrix}}_{g(X(t))} \quad (4)$$

According to (3), the discrete-time dynamics of UCS with $\omega = 0.0012$ and $T = 0.0021$ can be obtained as:

$$\begin{bmatrix}x_{d1}(k+1)T\\ x_{d2}(k+1)T\\ x_{d3}(k+1)T\end{bmatrix} = \begin{bmatrix}0.9798 & 0.0208 & 0\\ 0.058 & 0.9985 & 0\\ 0 & 0 & 0.9944\end{bmatrix}\begin{bmatrix}x_{d1}(k)T\\ x_{d2}(k)T\\ x_{d3}(k)T\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0.0021 & 0\\ 0 & 0.0021\end{bmatrix}\begin{bmatrix}-x_{d1}(kT)x_{d3}(kT)\\ x_{d1}(kT)x_{d2}(kT)\end{bmatrix} \quad (5)$$

To illustrate the consistency of the chaotic behavior of the discrete system (5) and the continuous system (4), we give the following simulation in Figures 1 and 2, respectively, for systems (4) and (5). The initial conditions are given as $x_1(0) = x_{d1}(0) = 0.5, x_2(0) = x_{d2}(0) = -0.3, x_3(0) = x_{d3}(0) = 0.4$. From Figures 1 and 2, it is observed that the discretized system (5) still preserves the chaotic behavior like the original continuous system (4).

III. SYNCHRONIZATION CONTROL OF DISCRETE CHAOTIC SYSTEMS

To discuss the synchronization of TRNGs, the following master and slave systems are described, respectively, as Master discrete UCS:

$$\begin{aligned}x_1(k+1) &= 0.9798x_1(k) + 0.0208x_2(k)\\ x_2(k+1) &= 0.058x_1(k) + 0.9985x_2(k) - 0.0021x_1(k)x_3(k)\\ x_3(k+1) &= 0.9944x_3(k) + 0.0021x_1(k)x_2(k)\end{aligned} \quad (6)$$

Slave discrete UCS:

$$\begin{aligned}y_1(k+1) &= 0.9798y_1(k) + 0.0208y_2(k)\\ y_2(k+1) &= 0.058y_1(k) + 0.9985y_2(k)\\ &\quad -0.0021y_1(k)y_3(k) + u(k)\\ y_3(k+1) &= 0.9944y_3(k) + 0.0021y_1(k)y_2(k)\end{aligned} \quad (7)$$

where $x_i, i = 1, 2, 3$ and $y_i, i = 1, 2, 3.$ are state variables of the master and slave systems, respectively. The control input $u(k) \in R$ introduced into (7) will be designed later to guarantee the synchronization of both the master and slave UCSs. The definition of $e_i(k) = y_i(k) - x_i(k), i = 1, 2, 3$, the dynamics of synchronization error between the master and slave systems given in (6) and (7) can be described by the following equation:

$$\begin{aligned}e_1(k+1) &= 0.9798e_1(k) + 0.0208e_2(k)\\ e_2(k+1) &= 0.058e_1(k) + 0.9985e_2(k)\\ &\quad -0.0021(y_1(k)y_3(k) + x_1(k)x_3(k)) + u(k)\\ e_3(k+1) &= 0.9944e_3(k) + 0.0021(y_1(k)y_2(k) - x_1(k)x_2(k))\end{aligned} \quad (8)$$

To complete the SMC design for synchronization, we first select a switching function as follows:

$$s(k) = e_2(k) + ce_1(k) \quad (9)$$

where c is a design parameter quickly decided later. Assume the system is in the sliding manifold, i.e., $s(k) = 0$. Then we have:

$$e_2(k) = -ce_1(k) \quad (10)$$

Therefore, when the system enters the sliding mode, i.e., $s(k) = e_2(k) + ce_1(k) = 0$, from (8) and (10), we have:

$$e_1(k+1) = (0.9798 - 0.0208\ c)e_1(k) \quad (11)$$

If c is selected satisfying $|0.9798 - 0.0208\ c| < 1$, by (11), $e_1(k)$ will converge to zero, and because $e_2(k+1) = -ce_1(k)$ in (10), then $e_2(k)$ will also converge to zero. We can observe that the dynamics of $e_3(k)$ will degenerate to $e_3(k+1) = 0.9944e_3(k)$ and then converge to zero. So, the system can reach complete chaos synchronization in the sliding manifold. After guaranteeing the synchronization in the sliding manifold, we still need to design an appropriate SMC to ensure the fact of $s(k) = 0$.

Theorem 1: If the controller $u(k)$ in (7) is appropriately designed as

$$u(k) = -f(k) + \alpha s(k), \quad (12)$$

then the sliding motion (i.e., $\lim_{k\to\infty} s(k) = 0$) is guaranteed. where $|\alpha| < 1$ and

$$f(k) = -(0.058e_1(k) + 0.9985e_2(k) - 0.0021(y_1(k)y_3(k) + x_1(k)x_3(k)) + 0.9798ce_1(k) + 0.0208ce_2) \quad (13)$$

Proof: From (8) and (9), we have

$$s(k+1) = e_2(k+1) + ce_1(k+1) = f(k) + u(k) \quad (14)$$

Substituting (12) into (14), we can obtain $s(k+1) = \alpha s(k)$. Since α is selected satisfying $|\alpha| < 1$, $\lim_{k\to\infty} s(k) = 0$ can be ensured.

Consequently, numerical simulations are performed to verify the synchronization of SMC design. For simulation, the initial conditions of master and slave systems are selected as $x_1(0) = 0.2, x_2(0) = -0.3, x_3(0) = 0.1$, and $y_1(0) = -0.8, y_2(0) = 1.2, y_3(0) = -1$ and the parameters in (9) and (12) are given as $\alpha = 0.1, c = 47$ satisfying, respectively, $|\alpha| < 1$ and $|0.9798 - 0.0208c| < 1$. Figure 3 shows the dynamic error responses between the master andslave

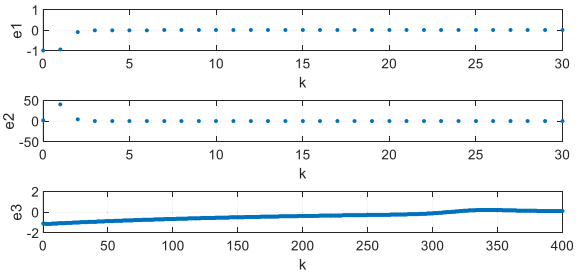

FIGURE 3. The error responses of the controlled master-slave discrete UCSs.

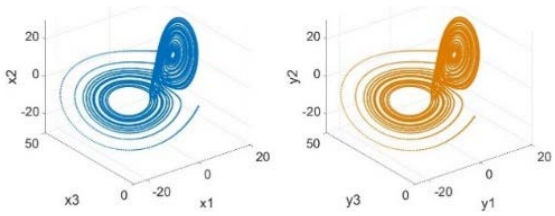

FIGURE 4. Strange attractors of master-slave discrete UCSs.

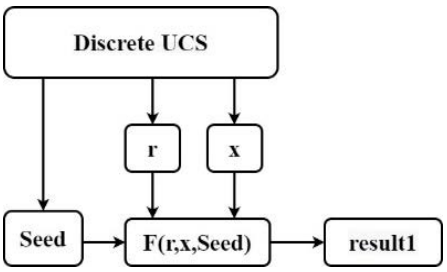

FIGURE 5. The structure of the El-Gamal-based hybrid function.

discrete UCSs, and Figure 4 shows the master (left) and the slave (right) strange attractors of discrete UCSs. From Figures 3 and 4, we can observe that the state response of the slave can be synchronized with the master, and the dynamic error will also converge to zero as expected.

IV. DESIGN OF TRUE RANDOM NUMBER GENERATORS

After solving the synchronization problem of master-slave chaotic systems, we introduce a novel TRNG design that integrates synchronized random-like chaos signals with the El-Gamal algorithm. The El-Gamal encryption algorithm has high security [10], which means that the El-Gamal encryption algorithm can convert plaintext into complex random ciphertext. So, to improve the randomness of the original discrete UCS, an El-Gamal-based hybrid function is firstly proposed, as shown in Figure 5.

The El-Gamal-based hybrid function F(r, x, Seed) in Figure 5 is described as follows:

step.1 Randomly select a prime number p, $p \in Z_p^*$.
step.2 Select a prime number g, $g \in Z_p^*$.
step.3 Convert chaos state to the IEEE754 format (see Figure 6 below), and use bits 16-23 of state x_1 in IEEE754 format as x, $x \in Z_{P-1}^*$, $0 \le x \le p-1$.

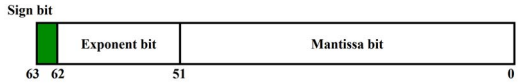

FIGURE 6. Double-precision floating-point numbers stored in IEEE754 format.

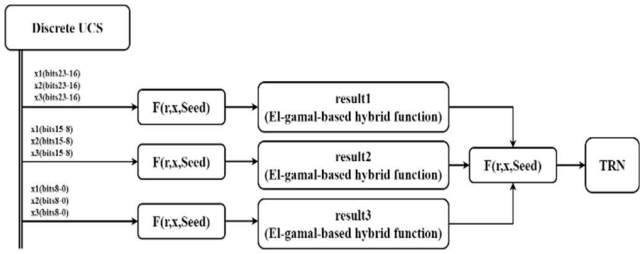

FIGURE 7. The structure of the TRN generator.

step.4 Calculation $y = g^x \bmod p$.
step.5 Use bits 16-23 of state x_2 in IEEE754 format as a random positive integer r, $r \in Z_{P-1}^*$.
step.6 Select the x_3 byte (bits 23-16) as the Seed.
step.7 Obtain the random value, result1$= (Seed \cdot y^r) \bmod p$.

After introducing the El-Gamal-based hybrid function, the algorithm is proposed in Figure 7 to obtain the TRN sequence.

In Figure 7, results $i, i = 1, 2, 3$ are obtained from the El-Gamal-based hybrid function in Figure 5 with the chaotic signals x_i. Furthermore, the result i, $i = 1, 2, 3$ is then mapped again by the El-Gamal-based hybrid function, and the design of the TRN generator is completed.

A. SECURITY ANALYSIS OF TRN

The National Institute of Standards and Technology (NIST) test [18] suite evaluates the randomness of the numbers obtained from the FPGA output. First, we set the length of the sequence by 10^5 bits or 10^6 bits, the number of subsequences is selected by 10 or 100. When the outcome value ≥ 0.01 passes the test, it confirms that the proposed TRN sequence is a true random number.

From Table 1, we can observe that the original states of the UCS have not passed all test items and only passed the Non-Overlapping-Template test. However, after the modulation of the El-Gamal-based hybrid function, the randomness has been improved a lot, but it still has not reached the standard of the NIST test, and the modulated TRNG passed all the test items. Also, we compare our results with the published papers [19], [20]. We perform the NIST test according to the conditions set by each paper. In the paper [19], the tested random number is 7000bits, and the bitstream length is set to 2, and the comparison results are given in Table 2. For the paper [20], the tested random number is 10000bits, and the bitstream length is set to 100, and the comparison results are given in Table 3. According to the comparison results, under the individual test conditions, we can see that our results might not be the best in all test items. This is because each random number generator has its advantages.

TABLE 1. NIST test.

Statistical test	P-value		
	UCS	El-gamal-based hybrid function	The proposed TRN
Frequency	0	0.534146	0.534146
BlockFrequency	0	0.911413	0.739918
CumulativeSums	0	0.008879	0.991468
Runs	0	0.911413	0.350485
LongestRun	0	0.534146	0.739918
Rank	0	0.350485	0.350485
FFT	0	0.004301	0.739918
NonOverlappingTemplate	0.534146	0.911413	0.991468
OverlappingTemplate	0	0.739918	0.122325
Universal	0	0.719747	0.55442
ApproximateEntropy	0	0.004301	0.739918
RandomExcursions	0	0.611108	0.911413
RandomExcursionsVariant	0	0.611108	0.987896
Serial	0	0.534146	0.978072
LinearComplexity	0.816537	0.455937	0.978072
	1.350683	7.842461	10.709922

TABLE 2. NIST Test comparison with [19].

Statistical Test	Binary and Ternary True Random Number Generators Based on Spin Orbit Torque [19]		The proposed TRN	
	P-VALUE	PROPORTION	P-VALUE	PROPORTION
Frequency	0.709991	2/2	0.94283	2/2
BlockFrequency	0.806341	2/2	0.300746	2/2
CumulativeSums	0.876773	2/2	0.511043	2/2
Runs	0.733546	2/2	0.621362	2/2
LongestRun	0.443576	2/2	0.331159	2/2
Rank	0.333851	2/2	0.802001	-
FFT	0.036257	2/2	0.323576	2/2
NonOverlappingTemplate	-	-	-	-
OverlappingTemplate	0.830234	2/2	0.644216	2/2
Universal	-	-	-	-
ApproximateEntropy	-	-	-	-
RandomExcursions	-	-	-	-
RandomExcursionsVariant	-	-	-	-
Serial	0.659356	4/4	0.856336	2/2
LinearComplexity	0.695952	2/2	0.843663	2/2
	6.125877		6.176932	

TABLE 3. NIST Test comparison with [20].

Statistical Test	A Compact L-V TRNG [20]		The proposed TRN	
	P-VALUE	PROPORTION	P-VALUE	PROPORTION
Frequency	0.202268	96/100	0.719747	100/100
BlockFrequency	0.213309	100/100	0.595549	99/100
CumulativeSums	0.428568	96/100	0.897763	100/100
Runs	0.171867	99/100	0.096578	100/100
LongestRun	0.437274	100/100	0.334538	99/100
Rank	-	-	-	-
FFT	0.474986	98/100	0.334538	98/100
NonOverlappingTemplate	-	-	-	-
OverlappingTemplate	0.055361	99/100	0.983453	99/100
Universal	-	-	-	-
ApproximateEntropy	-	-	-	-
RandomExcursions	-	-	-	-
RandomExcursionsVariant	-	-	-	-
Serial	0.494555	100/100	0.678686	100/100
LinearComplexity	0.249284	97/100	0.978072	99/100
	2.727472		5.618924	

However, judging from the sum of the outcomes in all test items, the TRN we proposed is better than others, and it can also show that our proposed TRN has better randomness characteristics.

V. FPGA CIRCUIT IMPLEMENTATION

In this part, the above-mentioned TRNGs design is then implemented on FPGA, and IP core (intellectual property core) is designed through Vivado HLS (high-level synthesis). The production unit has used units such as subtractor, adder, and multiplier. PYNQ (Python Productivity for Zynq) is a development version based on the ZYNQ architecture and supporting Python. The development version used in this paper is PYNQ-Z2. The special feature of PYNQ is the built-in interactive logic that encapsulates the ARM processor and FPGA. Users can directly edit programs through Jupyter interactive notebook and Python API, and programmable logic circuits are introduced as hardware libraries or provide hardware acceleration functions for the system.

We utilize FPGA to implement the proposed TRNG and their synchronization design, as shown in Figure 8. The master-slave discrete UCSs are synchronized by the synchronization controller presented in Section 3. Hence, we can simultaneously obtain the master-slave synchronized TRNs. Figure 9 shows the wiring diagram, it contains two (master and slave) discrete UCS intellectual property (IP) cores, eight mixed functions IP cores, and GPIO IP core. FPGA chip statistics have been obtained and given in Table 4. Table 5 shows the latency and throughput of the proposed TRNG.

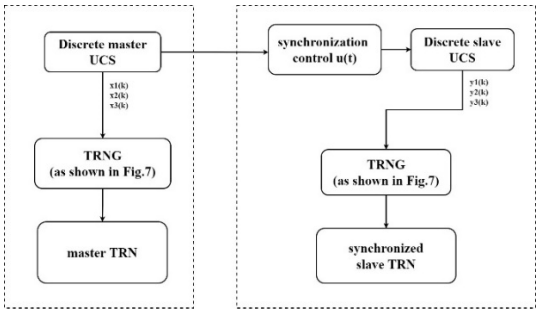

FIGURE 8. The structure of chaos-based synchronized TRNs.

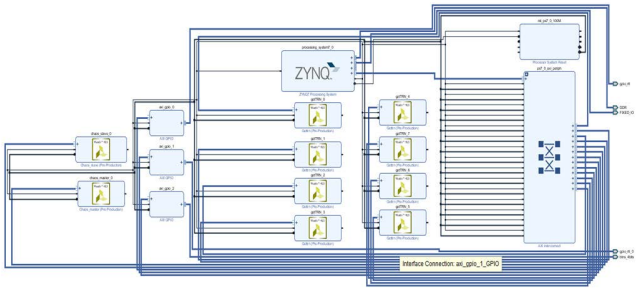

FIGURE 9. Wiring diagram.

TABLE 4. Hardware resources of the implementation of Fig. 8.

Resource	Utilization	Available	Utilization%
LUT	12383	53200	23.28%
LUTRAM	131	17400	0.75%
FF	13483	106400	12.67%
DSP	145	220	65.91%

TABLE 5. Latency and throughput of the TRNG.

TRNG	latency	throughput
	2.97ms	4.895Mbits

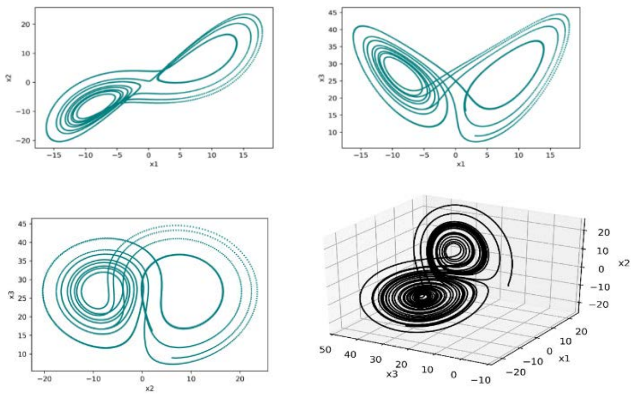

FIGURE 10. Strange attractor of UCSs. (FPGA output).

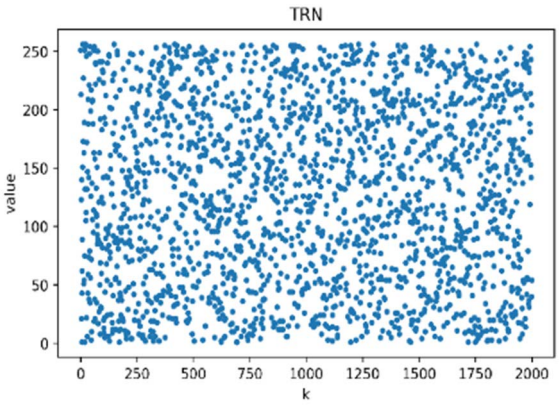

FIGURE 11. TRN display.

VI. CHAOTIC BEHAVIOR WITH THE FPGA IMPLEMENTATION

Next, we use FPGA to implement the above-designed TRNGs. The initial conditions are set to $x_1(0) = 3, x_2(0) = 6, x_3(0) = 9$, for realization. The responses of the state variables with FPGA implementation are recorded and shown in Figure 10. By observing Figure 10, it reveals that there are strange attractors peculiar to the chaotic system through the FPGA implementation, which shows that the chaotic behavior is still preserved after the FPGA implementation.

In Figure 11, with the initial conditions $x_1(0) = 0.2$, $x_2(0) = -0.3, x_3(0) = 0.1$, we show two thousand TRNs states to show the randomness of obtained TRNs. In Figure 11, k is the sampling interval. Since the TRN data is 8bit, the state range will be between 0 and 255.

Next, to show the synchronization of the master and slave TRNGs in FPGA, we display 100 data of our master and slave TRNGs. The initial conditions are selected as

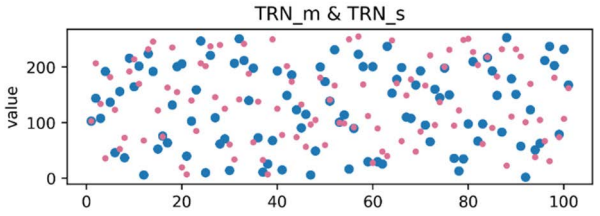

FIGURE 12. TRNG signal display (synchronization controller is disabled.)

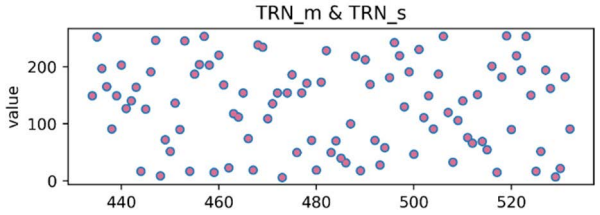

FIGURE 13. TRNG signal display (synchronization controller is enabled.)

TABLE 6. Information entropy results of the chaotic systems.

TRNG	Number of '0'	Number of '1'	P(0)	P(1)	H(shannon)
	34860	35140	0.498	0.502	0.99999

$x_1(0) = 0.6, x_2(0) = 0.8, x_3(0) = 0.2, y_1(0) = 0.3, y_2(0) = 0.4 y_3(0) = 0.1, \alpha = 0.1, c = 47$. In Figure 12, the synchronization controller is not enabled, and the output signals of the master-slave TRNs cannot be synchronized. After allowing the synchronization to a controller, the master-slave TRNs can be synchronized as expected, as shown in Figure 13.

VII. INFORMATION ENTROPY AND HISTOGRAM ANALYSIS

In the section, we give information entropy and histogram analysis. Shannon's entropy formula [21] is given as follows:

$$H_{shannon} = -\sum_{i=1}^{n} P(TRN) \log_2 P(TRN) = -P(0) \cdot \log_2 \{P(0)\} - P(1) \cdot \log_2 \{P(1)\} \quad (15)$$

where $P(TRN)$ is the probability of appearance of the data proposed TRNG, and it may either be 0 or 1. We generate 70,000 bits for the proposed TRNG, where the probability of $P(0)$ is 0.498%, the probability of $P(1)$ is 0.502%, and the deviation between the two is 0.004% Substituting equation (15), we get:

$$H_{shannon} = -0.498 \cdot \log_2 0.498 - 0.502 \cdot \log_2 0.502 = 0.99999 \quad (16)$$

The results given in Table 6 reveal that the information entropy $H_{shannon}$ obtained from the proposed TRNG is very close to the ideal value. It shows that the bit-level random numbers obtained from the proposed TRNG have a uniform distribution, making it difficult to predict.

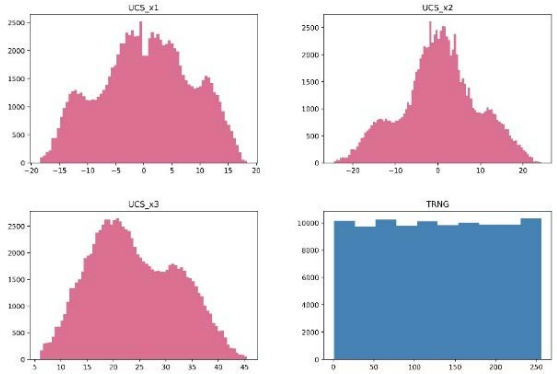

FIGURE 14. Histogram analysis.

A. HISTOGRAM ANALYSIS [22]

In this analysis, the initial value of UCS is set to $x_1(0) = 0.6, x_2(0) = 0.8, x_3(0) = 0.2$, and a total of 100,000 data are generated. It can be seen from Figure 7 that the proposed TRNG can get 8 bits at one time, so here will be presented as integers from 0 to 255. The three states of the UCS chaotic system and the proposed TRNG histogram analysis are shown in Figure 14.

From the histogram analysis in Figure 14, the uniformity of the original chaos states has been improved very well by the proposed TRNG design.

VIII. CONCLUSION

In this paper, a novel chaos-based hybrid function integrated with the El-Gamal algorithm has been proposed to design TRNG. Moreover, a discrete sliding mode control (SMC) method has been designed to guarantee the synchronization of the master-slave discrete unified chaotic systems. Furthermore, the random bit sequences produced by TRNG have been evaluated by NIST-800-22 statistical standards and compared with the existing results in the literature to demonstrate their validity and correctness. Finally, the proposed chaos-based TRNGs and their synchronization has been implemented in FPGA. The randomness quality of the proposed TRNGs has been ensured by using Shannon's entropy and histogram analysis.

REFERENCES

[1] İ. Koyuncu and A. T. Özcerit, "The design and realization of a new high speed FPGA-based chaotic true random number generator," *Comput. Electr. Eng.*, vol. 58, pp. 203–214, Feb. 2017.
[2] İ. Koyuncu, M. Tuna, İ. Pehlivan, C. B. Fidan, and M. Alçin, "Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator," *Anal. Integr. Circuits Signal Process.*, vol. 102, no. 2, pp. 445–456, Feb. 2020.
[3] R. C. Hilborn, "Sea gulls, butterflies, and grasshoppers: A brief history of the butterfly effect in nonlinear dynamics," *Amer. J. Phys.*, vol. 72, no. 4, pp. 425–427, 2004.
[4] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, p. 821, 1990.
[5] H.-T. Yau, C.-L. Kuo, and J.-J. Yan, "Fuzzy sliding mode control for a class of chaos synchronization with uncertainties," *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 7, no. 3, pp. 333–338, 2006.
[6] S. M. Lee, D. H. Ji, J. H. Park, and S. C. Won, "H_∞ synchronization of chaotic systems via dynamic feedback approach," *Phys. Lett. A*, vol. 372, no. 29, pp. 4905–4912, 2008.
[7] M. Jalilian, A. Ahmadi, and M. Ahmadi, "Hardware implementation of a chaotic pseudo random number generator based on 3D chaotic system without equilibrium," in *Proc. 25th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2018, pp. 741–744.
[8] H. A. Abdullah and H. N. Abdullah, "A new chaotic map for secure transmission," *Telkomnika*, vol. 16, no. 3, pp. 1135–1142, 2018.
[9] H. A. Abdullah, H. N. Abdullah, and W. A. Mahmoud Al-Jawher, "A hybrid chaotic map for communication security applications," *Int. J. Commun. Syst.*, vol. 33, no. 4, p. e4236, Mar. 2020.
[10] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[11] D. T. Nguyen, T. N. Nguyen, H. Kim, and H. J. Lee, "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1861–1873, Aug. 2019.
[12] M. F. Tolba, A. M. AbdelAty, N. S. Soliman, L. A. Said, A. H. Madian, A. T. Azar, and A. G. Radwan, "FPGA implementation of two fractional order chaotic systems," *AEU-Int. J. Electron. Commun.*, vol. 78, pp. 162–172, Aug. 2017.
[13] O. Guillén-Fernández, M. F. Moreno-López, and E. Tlelo-Cuautle, "Issues on applying one-and multi-step numerical methods to chaotic oscillators for FPGA implementation," *Mathematics*, vol. 9, no. 2, p. 151, 2021.
[14] A. M. Garipcan and E. Erdem, "Implementation and performance analysis of true random number generator on FPGA environment by using non-periodic chaotic signals obtained from chaotic maps," *Arabian J. Sci. Eng.*, vol. 44, no. 11, pp. 9427–9441, Nov. 2019.
[15] S. Vaidyanathan, A. Sambas, B. Abd-El-Atty, A. A. Abd El-Latif, E. Tlelo-Cuautle, O. Guillén-Fernández, and M. A. H. Ibrahim, "A 5-D multi-stable hyperchaotic two-disk dynamo system with no equilibrium point: Circuit design, FPGA realization and applications to TRNGs and image encryption," *IEEE Access*, vol. 9, pp. 144555–144573, 2021.
[16] A. M. González-Zapata, E. Tlelo-Cuautle, I. Cruz-Vega, and W. D. León-Salas, "Synchronization of chaotic artificial neurons and its application to secure image transmission under MQTT for IoT protocol," *Nonlinear Dyn.*, vol. 104, pp. 4581–4600, May 2021.
[17] J.-J. Yan, C.-Y. Chen, and J. S.-H. Tsai, "Hybrid chaos control of continuous unified chaotic systems using discrete rippling sliding mode control," *Nonlinear Anal.: Hybrid Syst.*, vol. 22, pp. 276–283, Nov. 2016.
[18] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication, Gaithersburg, MD, USA, Tech. Rep. 800-22, May 2002.
[19] H. Chen, S. Zhang, N. Xu, M. Song, X. Li, R. Li, Y. Zeng, J. Hong, and L. You, "Binary and ternary true random number generators based on spin orbit torque," in *IEDM Tech. Dig.*, Dec. 2018, p. 36.
[20] A. T. Erozan, G. Y. Wang, R. Bishnoi, J. Aghassi-Hagmann, and M. B. Tahoori, "A compact low-voltage true random number generator based on inkjet printing technology," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 6, pp. 1485–1495, Jun. 2020.
[21] R. Sivaraman, S. Rajagopalan, A. Sridevi, J. B. B. Rayappan, M. P. V. Annamalai, and A. Rengarajan, "Metastability-induced TRNG architecture on FPGA," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 44, no. 1, pp. 47–57, Mar. 2020.
[22] A. M. Garipcan and E. Erdem, "A TRNG using chaotic entropy pool as a post-processing technique: Analysis, design and FPGA implementation," *Anal. Integr. Circuits Signal Process.*, vol. 103, no. 3, pp. 391–410, Jun. 2020.

T. L. LIAO (Member, IEEE) received the B.S. degree (Hons.) from the Department of Engineering Science, National Cheng Kung University (NCKU), Taiwan, in 1982, and the M.S. and Ph.D. degrees (Hons.) from the Department of Electrical Engineering, National Taiwan University, Taiwan, in 1984 and 1991, respectively. He joined the Department of Engineering Science, NCKU, as an Associate Professor, in 1991, and was promoted to a Full Professor, in 1999. His research interests include control theory, multi-agent systems, signal and image processing, chaos communication systems, and cryptosystems.

P. Y. WAN received the B.S. degree in mechatronics engineering from the National Kaohsiung University of Technology, Taiwan, in 2016, and the M.S. degree in computer and communication from the Shu-Te University of Technology, Taiwan, in 2018. He is currently pursuing the Ph.D. degree with the Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan. His main research interests include chaotic systems and nonlinear control.

JUN-JUH YAN received the B.S. degree in electrical engineering from the National Cheng Kung University, Taiwan, in 1987, the M.S. degree in electrical engineering from the National Central University, Taiwan, in 1992, and the Ph.D. degree in electrical engineering from the National Cheng Kung University, in 1998. He is currently a Professor with the Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. His main research interests include multi-robot dynamic systems, chaotic systems, neural networks, variable-structure control systems, and adaptive control.