Arabian Journal for Science and Engineering (2019) 44:9427–9441
https://doi.org/10.1007/s13369-019-04027-x

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Implementation and Performance Analysis of True Random Number Generator on FPGA Environment by Using Non-periodic Chaotic Signals Obtained from Chaotic Maps

Ali Murat Garipcan[1] · **Ebubekir Erdem**[1]

Received: 4 March 2019 / Accepted: 2 July 2019 / Published online: 10 July 2019

Abstract
In this study, FPGA implementation of a hybrid random number generator (HRNG) based on digital design techniques is given. The ring oscillators (ROs) are used as the noise source of HRNG, and true randomness is obtained by sampling jitter signals forming on the oscillators. The statistical quality and reliability of random number generators that used jitter as source of true randomness alone are often cryptographically insufficient. For this reason, one-dimensional discrete-time chaotic maps such as quadratic map, logistic map and Bernoulli shift map are benefited in order for HRNG to meet these cryptographic requirements. In contrast to many studies in the literature, non-periodic signals derived from chaotic systems of a powerful source of entropy are used instead of periodic signals for the sampling of jitter signals in the system. Depending on the usage of chaotic systems, output bit rate and reliability of high generator model that does not need post-processing techniques and is easily applicable to digital devices are obtained. The hybrid system is tested in total six different scenarios for two separate ring oscillator (RO) architectures of 25 and 114 pieces consisting of three different chaotic maps and equal-length inverters. The statistical qualifications of the random numbers obtained from HRNG for each scenario are verified by NIST 800-22 tests. Also, for each scenario, the design parameters of the generator are examined and the hardware performances and non-periodicity analyses of the chaotic maps are performed. Based on the obtained results, it is demonstrated that the HRNG based on non-periodic sampling can be used for cryptographic purposes.

Keywords Hybrid random number generator · Non-periodic sampling · Jitter · Ring oscillator · Scale index · NIST 800-22

1 Introduction

The quality of the random numbers used in cryptography has significant importance for the reliability and power of the system in which they are used. Therefore, random numbers, unlike other areas of use, must meet some strict requirements about system security in cryptography. Also random numbers, unless they are obtained with the correct design techniques, entirely endanger the security of cryptographic applications [1, 2]. Concerning these requirements, besides having good statistical properties, basic characteristic competencies such as non-reproducibility and unpredictability are the most important features for random numbers required in cryptography. However, the obtaining of random numbers which can meet these competencies is a cryptographically significant and costly problem. To solve this problem, customized components/designs called 'Random Number Generators (RNG)' are needed for cryptographic applications.

Generators, of which general classification is presented in Fig. 1, are divided into two basic design classes, namely the pseudo-RNGs (PRNGs) and true RNGs (TRNGs), between themselves. Hybrid RNGs (HRNGs), in which these two design classes are used together, constitute another design class.

PRNGs that can be implemented as software and hardware have a deterministic structure. They generate random-looking number sequences, through expanding the seed value that is used as the entropy source within algorithmic structures. Due to their deterministic structures, the repetition of internal states after a sufficient number of iterations, in other

✉ Ebubekir Erdem
aberdem@firat.edu.tr

Ali Murat Garipcan
agaripcan6223@gmail.com

[1] Department of Computer Engineering, Firat University, Elazig, Turkey

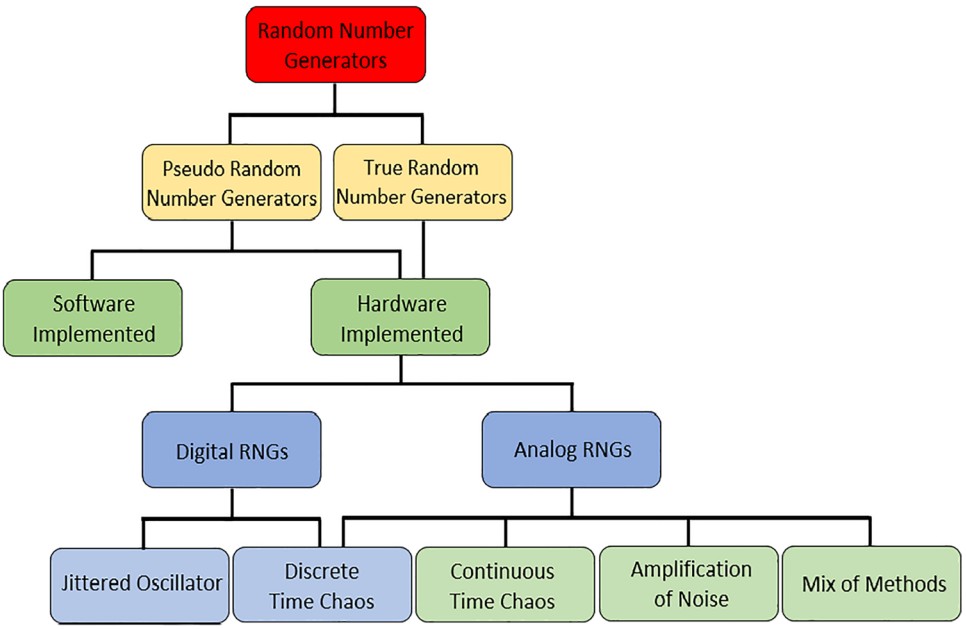


Fig. 1 Classification of random number generators

words, periodicity, is the most obvious shortcoming of this design class. Furthermore, the random numbers generated from the internal states of the generator or the seed value can be easily estimated [3]. Although random numbers, close to ideal statistical quality, can be obtained from fast and inexpensive solutions, the deterministic structures of PRNGs limit the use of them alone in cryptographic applications.

Unlike PRNGs, there is not any standard definition for TRNGs, and the design architecture of them is presented in Fig. 2. The process of obtaining design architecture and random numbers consists of three basic stages, namely noise source, sampler and post-processing. In the system, random numbers are obtained by sampling of noise source. Depending on the lack of entropy of the noise sources, random numbers are passed through the post-processing stage and their statistical properties are verified in terms of cryptography, outside the system. Noise source forms the heart of the designs, because TRNGs use the physical processes or states of the real quantum world phenomena explained by the uncertainty principle, which cannot be expressed with a deterministic model and therefore behaviours of which are impossible to predict, as a randomness input. This basic characteristic of noise source meets the basic competencies such as unpredictability and non-reproducibility of random numbers obtained from the system.

Although TRNGs are mostly slow, hardware-dependent and offer expensive solutions, the most important disadvantage of them is the cryptographic inadequacy of the statistical quality of the true random numbers they produce. Therefore, the output of the generator is not used directly in cryptography. Because the entropy of the noise sources used in true random number generators is generally low. At this point, post-processing techniques are needed in generator designs. In TRNGs, it is difficult to find powerful noise sources with high entropy, which enables defining the probabilistic properties of random number sequences. Besides, new designs, which are alternatives for current post-processing techniques, constitute another critical problem area, to increase entropy without reducing the output bit rate in the system.

In order to overcome these shortcomings, chaotic dynamics have been recently used in RNG designs as a direct source of entropy or to increase entropy in the system [4]. The basic characteristics of chaotic systems with nonlinear structure, like ergodicity and exponential sensitivity to initial and system parameters, have made chaotic systems one of the ideal solution tools for the design of any cryptographic primitive. Partial changes in the input parameters of chaotic systems, a deterministic and random-like process, lead to significant changes in the system output [5, 6]. Chaotic system outputs with irregular structure, like sampled from a true noise source, are unpredictable. These specific features, which lead to the formation of a natural relationship between chaotic systems and cryptography, can be used to improve the safety and statistical quality of the system in TRNG designs.

In this study, a HRNG design, in which random numbers with cryptographically required qualifications can be obtained, is presented. HRNG consists of two separate design classes hierarchically. The RO-based TRNG was used as the non-deterministic component of the hybrid system. The deterministic side of the system consists of three one-dimensional discrete-time, different chaotic maps including quadratic, logistic and Bernoulli shift maps. The HRNG was tested in six different scenarios for three different chaotic systems and 25 and 114 pieces fixed RO architecture, each

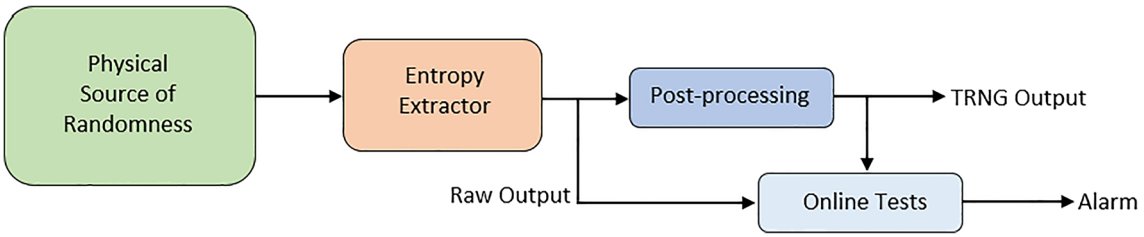


Fig. 2 General design architecture of a TRNG

consist of three inverters, and statistically successful results were achieved.

1.1 Contributions and Motivations

This study is directly related to the studies given in [7, 8], and the main points contributed can be summarized as follows:

The TRNG model using ROs was first proposed by Sunar and Stinson [9]. Another well-known oscillator-based architecture has been proposed by Wold and Tan [8]. In the system, which stands out by its simplicity, 25 ROs with independent sampling units, each consisting of three inverters, were used. The TRNG component of the hybrid system was constructed on this architecture. The shortcomings of the proposed TRNG architecture highlighted in [10, 11] are among the contribution points of this study in terms of system security. In both studies, it was emphasized that the number of ROs in the system was reduced uncontrollably according to the Sunar and Stinson [9] model and this caused entropy loss in the system. It was stated that the pseudo-randomness formed connected to the loss of entropy in the system was masked by the XOR (exclusive OR) process. It was expressed that in the system in which post-processing techniques are not being used, pseudo-randomness could be guessed or the system could be manipulated from outside. At this point, a hybrid generator model resistant to environmental changes and tampering was developed by increasing the entropy of the system with non-periodic sampling inputs obtained from the chaotic maps/systems.

In the literature, HRNG-based chaotic sampling has been proposed by Tuncer in [7]. However, the limiting effect of the output bit rate of the generator connected to the hardware complexity of the selected chaotic system has been ignored. In addition, due to the complexity of the chaotic system used, design details such as area/source and energy consumption of the proposed model as well as non-periodic analysis of the chaotic system are missing. This limits the use of the generator for cryptographic purposes. Within the scope of the study, these shortcomings were eliminated by the use of chaotic systems with less structural complexity than according to [7]. In the hybrid system, three different chaotic systems were used to examine the effect of chaos-related complexity on the generator's design parameters and positive results were obtained. In this aspect, the applicability of non-periodic signals obtained from different chaotic dynamics was demonstrated for sampling purposes. In another aspect, the study provided the possibility to see the random quality and hardware costs of the chaotic systems, which were analysed for non-periodicity in real-time cryptographic applications. In the output bit rate point, which is accepted to be an important performance parameter for RNG designs, more successful results in comparison with the [7] were obtained for all scenarios. Furthermore, according to [7], a more functional hybrid generator model was obtained in terms of applicability, output bit rate and energy consumption with the preference of chaotic maps with less complexity.

The rest of the study is organized as follows: In Sect. 2, information on the literature review is listed. Information about RO architecture and jitter formation and the theoretical information about the used chaotic maps are presented in Sects. 3 and 4, respectively. The hardware implementation of the HRNG, design parameters of which are given, on FPGA environment is presented in separate Sect. 5. In Sect. 6, the results obtained from the HRNG are analysed statistically and the results are interpreted in terms of the performance of the system. Finally, Sect. 7 concludes the paper briefly.

2 Related Works

Systems with chaotic behaviour are hypersensitive to initial and system parameters. Minor changes in system parameters or initial conditions cause significant changes in system orbits. The fact that this situation can theoretically meet the confusion and diffusion properties that modern cryptographic systems should have led to the frequent use of chaos in cryptography [4, 6, 11]. In the literature, there are many studies [12–16] carried out by benefiting from this natural relation between chaos and cryptography.

In the literature, random numbers can be generated as software and hardware with different approaches based on continuous- and discrete-time chaotic dynamics. In the literature proposed in this direction, there are generator designs [17, 18], especially based on discrete-time chaotic dynam-

ics. Unlike software applications, effective results can be obtained in terms of efficiency, speed and system security in hardware-based designs and in chaotic systems which require high processing power [6, 7]. Furthermore, because of the implementation difficulty related to the process complexity of continuous-time chaotic systems, discrete-time structures are more preferred for hardware-based designs [19, 20]. It is observed in the literature that one-dimensional logistic maps that are easy for hardware implementation due to the simple mathematical definition often appear to be preferred [18, 21–23]. Besides, it is observed that one-dimensional multimodal chaotic maps [24, 25] with better randomness qualities and structures consisting of a combination of multiple chaotic maps [26] are used for cryptographic purposes.

Jitter on FPGA environments which is frequently preferred in terms of system security in TRNG designs often uses the true randomness source. Jitter's basic preference is that it can be easily obtained on ROs which have simple design structure and are easily applicable on integrated devices such as FPGA. Another reason for preferring the jitter is that with high-frequency signals formed on ROs can be achieved high output bit rate TRNG designs. Some studies that used ROs in the literature are as follows:

The use of the jitter obtained from ROs as a noise source in TRNG designs has been proposed by Sunar and Stinson [9]. The design with the given mathematical model consists of 114 free oscillating ROs, each consisting of 13 inverters. In the system, the high-oscillating RO outputs, coupled with the XOR process, were sampled with the help of a D-type flip-flop. In the system, to eliminate the statistical weaknesses of the sampled pure random numbers, resilient functions were used as the post-processing technique. Successful implementation of the proposed model was performed in [27] with fewer numbers of oscillators and inverters. In another study in which referenced Sunar and Stinson, Wold and Tan [8] have proposed a TRNG model using 25 ROs, each consisting of three inverters. The output bit rate of the model, which passed statistical tests without post-processing techniques, has been high. Other TRNG designs using ROs have also been proposed by Kollhenberger and Gaj [28], Golic [29], Dichtl and Golic [30] and Tuncer [31].

Some of the chaos-based HRNG designs in the literature are as follows: Tuncer achieved true random numbers with non-periodic sampling input within a RO-based hybrid architecture in [7]. Sinusoidal iterator was used as the sampling input for 25-, 10-, 5-piece RO architectures, respectively, each consisting of three inverters, and successful results were obtained. Avaroğlu et al. [32], in a true hardware-based TRNG in which they used ROs as the physical noise source, proposed a new chaos-based approach as an alternative to the existing post-processing techniques. The proposed new post-processing technique is logistic map based. The hybrid system was tested for different ROs to demonstrate the applicability of the chaotic system as post-processing, and successful results were obtained. The cryptographic suitability of random numbers obtained from the hybrid generator was proved by statistical tests.

In another study, Avaroğlu et al. [33] implemented the signals they obtained from ROs (5, 3) as an additional input to a PRNG in which chaotic attractors in different modes (2 + 2, 2 + 4, 5 + 4) were used. The PRNG, for which unsuccessful results were obtained for the (2 + 4 and 4 + 5) modes alone, successfully passed the statistical tests by achieving a hybrid structure. Arslan Tuncer [34] for two separate PUF circuits in the system, 64 ROs, each consisting of 13 inverters, was used. Random numbers obtained from the RO-PUF implemented on two different FPGA cores against the same query were passed through the post-processing technique, and successful results were obtained. In [19], the query input of the PUF circuit with 128 ROs, each consisting of three inverters, was obtained from the logistic map with chaotic behaviour. Avaroglu et al. [35] obtained a hybrid architecture using the random numbers obtained from the Sprott 94 chaotic attractor in the AES block encryption standard. For randomly selecting 128-bit initial inputs, the encrypted data obtained from AES were subjected to XOR processing with 128-bit random data obtained from the chaotic attractor to obtain the output of the hybrid system. The 128-bit output value obtained for each iteration is also given as the input to the block encryption standard for the next iteration. The hybrid system, whose internal state values were updated with chaotic system, passed the statistical tests successfully.

Özkaynak proposed a hybrid model in [4] by incorporating true random outputs obtained from chaos into a hash-based cryptographic function. Hybrid architecture, in which chaotic systems are used as entropy source, passed the statistical tests successfully. Allhadawi et al. [36] proposed a new hybrid architecture consisting of a 31-bit and 33-bit combination of LFSRs and a discrete-time chaotic map. In the hybrid system, random numbers are produced in two stages. In the first stage, two separate LFSR structures are directly linked to the chaotic map by the XOR process. In the first stage, the random numbers obtained from the interaction of two separate LFSRs and the chaotic system are obtained in the second stage by passing the selection criteria associated with the chaotic system. The verification of the hybrid system was accomplished by the test packs of NIST, TestU01 and DIEHARD. Two separate hybrid architectures based on another chaotic entropy based on LFSR were proposed by Jiteurtragool and Masayoshi in [37]. In a different study, Merah et al. [38] used the continuous-time Chua circuit designed on FPGA as an entropy source. They generated the true random numbers obtained from the chaotic oscillator as additional inputs for the PRNG design and obtained a cryptographically secure HRNG architecture.

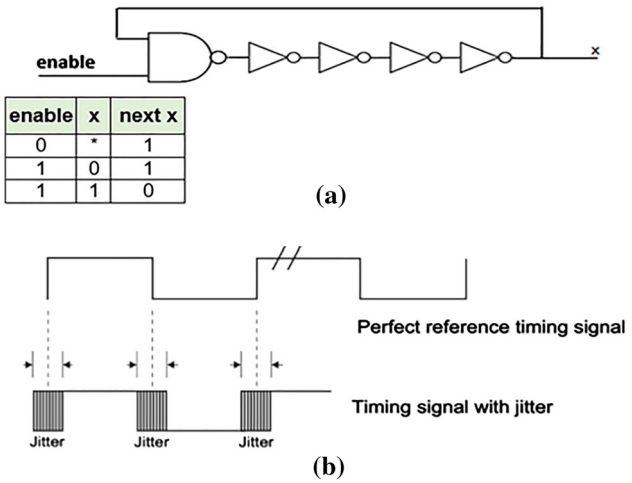


Fig. 3 **a** RO architecture, **b** formation of jitter signals

3 Generic Architecture of Proposed System

RO, open schematic structure of which is given in Fig. 3a, is a combinational loop consisting of a single number of inverters (delay elements). In the literature, ROs are often preferred as an entropy source in the design cycles of TRNGs because of their simple and easily applicable structures [32, 39]. In the significant majority of designs, the phase jitter, also called jitter, is used as an entropy source on square-wave signals at the oscillator outputs. The main reason for this case is directly related to being the true randomness of the jitter. Because the usage of entropy sources whose behavior cannot be expressed with the deterministic model meets the true randomness the most distinctive feature of a TRNG design class.

The jitter can be observed theoretically as the random variations of the normal signal in the forward or backward direction in the frequency (time) domain due to the electronic or thermal noise, as illustrated in Fig. 3b. These random variations cause small temporal changes in the rising and falling edge positions of the ideal clock signals, periods (cycles) of which are actually constant, as depicted in Fig. 3b. These variations, also named the Gaussian jitter, are an undesirable feature in electronic systems, and they occur entirely randomly, depending on the production and operating conditions of the logic circuit elements.

The general design architecture of HRNG given in Fig. 4 consists of two distinct hierarchical components: deterministic and non-deterministic. In the system, non-deterministic and deterministic components are represented by RO-based TRNG (fast oscillator) and discrete-time chaotic systems (slow oscillator), respectively. Bold lines represent buses, and thin lines represent bit-level inputs and outputs in Fig. 4.

The main characteristic behaviour of HRNG based on the relationship between hierarchical components in Fig. 4 can be summarized as follows:

Any f_i signal in the system is a square-wave signal with period T. In the system, n and τ are the number of inverters and the delay time on a single inverter, respectively. T is calculated as $T = 2(n \times \tau)$ and $f(t)$ *is as* $f(t) = f(t+T)$ for any t time. However, the period of signals represented by the $(f_1, f_2 \ldots f_r)$ in the system is irregular at run time and is not in the ideal square-wave format. Because, although they are identical, the delay time formed on the inverters representing the delay chain of each oscillator is variable for each iteration. This unstable delay time, represented by the $\check{T}$ in the system, causes a periodic disorder on the output signal (f). Therefore, the actual period of the output signal is $T = T + \check{T}$ and $(\check{T})$ value is in the range $(-T/2, T/2)$. The amount of this periodic irregularity $(\check{T})$, expressed by the Gaussian distribution, which is the actual random occurring depending on the production or working conditions, cannot be estimated for any sampling time. The amount of periodic irregularity is an important metric for TRNG designs.

In the system, f_c represents the non-periodic sampling input obtained from the chaotic maps modelling. Detailed information on the modelling of chaotic systems is given in Sect. 4. The true random output of hybrid system is obtained by sampling the oscillator outputs (f_b) combined with the XOR process. For a one-bit sampling signal obtained from chaotic maps, the sampling of the oscillators, the combining operation and obtaining one-bit true random output operations occur as synchronously in the system. D-type flip-flops are used for sampling in the system.

4 Chaotic Maps

Although they seem to be complicated, the systems exhibiting chaotic behaviours consist of equations of a nonlinear structure, which can be expressed usually with simple mathematical definitions. Despite their deterministic structures, the outputs of the chaotic systems have natural random appearance depending on their exponential sensitivities to the initial and system parameters. Therefore, chaotic components are used instead of the noise signal that is used as an entropy source in the design of many RNGs. The use of chaos in RNG designs removes most of the time the need for complicated and difficult operations required to obtain and process noise signals. However, the random number sequences obtained with chaotic systems are deterministic and reproducible, even though they seem to be unpredictable and non-periodic [4, 6, 18, 25]. This case leads to a debate on whether any cryptographic primitive can be constructed on the randomness of chaotic systems alone. For this reason, the studies [4, 7, 18, 32] in which chaotic systems were used as additional inputs in PRNG and TRNG designs have been recently encountered in the literature. This study, which overlaps with reference studies regarding the use of chaos, will also provide an

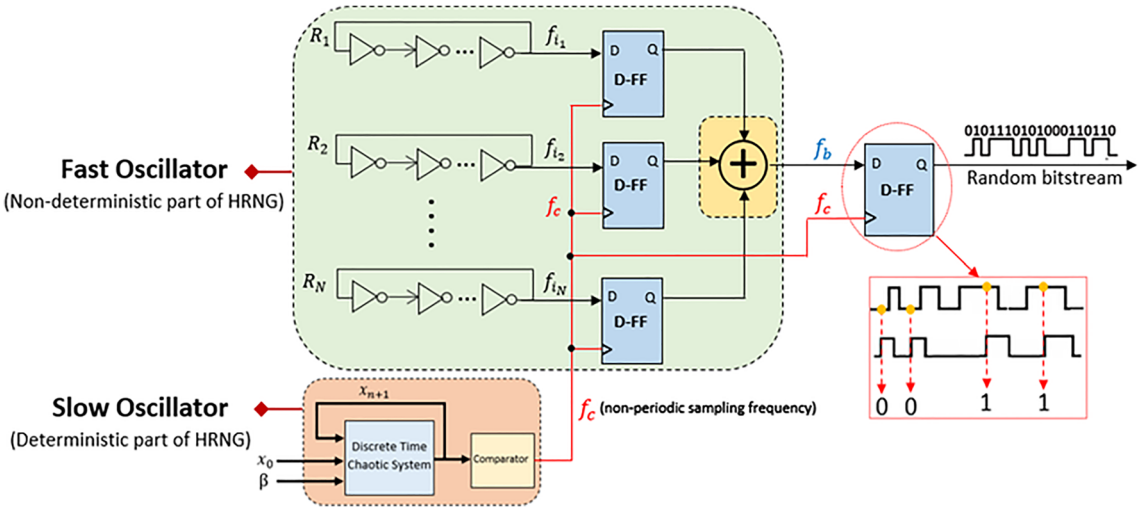


Fig. 4 Generic architecture of proposed HRNG

opportunity to see the performance of chaotic systems for hardware-based real-time implementations.

Discrete-time chaotic systems are generally formed by the time-dependent iteration of a simple nonlinear equation and equations with feedback characteristics [6]. The common feature of chaotic maps used in the study is being discrete time. This case facilitates the hardware implementation of chosen chaotic systems with regard to process complexity compared to discrete-time systems with analogous or more complex structure. Another reason for the choice of selected chaotic systems is the elapsed time for getting a one-bit chaotic signal for sampling from the circuits created corresponding to these systems. This time is equal for all three systems and corresponds to a minimum of 13 clock signals. The output bit rate is a significant performance parameter for RNG designs. The chaotic signals used for sampling high-oscillating RO outputs in the hybrid system are intended for minimizing the effect of limiting the output bit rate of the hybrid system. This ratio is 1/13 of the operating frequency applied to the input of chaotic circuits. The theoretical details of the selected chaotic systems are as in Table 1.

5 Hardware Implementation of HRNG

In this section, detailed information about the implementation on the FPGA of the scenarios of the HRNG generated according to the number of chaotic maps was given. For the implementation, the Altera EPC4GX150 FPGA board was used. HRNG scenarios were created by using schematic and VHDL dataflow design techniques on Quartus II platform. HRNG, which is generic design architecture, is given in Fig. 4 for two separate RO architectures ((114,3)–(25,3)), and three different chaotic systems tested a total of six separate scenarios and successful results were obtained.

The design details and logical work of the RO core given in Fig. 5A in the system briefly are as follows: for each scenario, there is a 2×1 mux (multiplexer) connected to the common select pin (sel) at the RO inputs. The data0, data1 and sel inputs of mux are the enable, feedback and select pins of the RO in Fig. 3a, respectively. The select pin which was obtained from the physical environment is the control pin of the RO, and at the start position, sel is at logical 0 level. The output pin of mux is connected to the input of the block structure representing the inverters of the oscillator. In order to obtain the jitter signal at the oscillator output, the enable data0 input must be at the logical 1 level during the system's operation period. In the system, data0 input is active for sel = '0' status and the oscillator output consisting of an odd number of inverters is always at the logical 0 level. For sel = '1' state, the oscillator's feedback input which oscillates both logical levels is active. In order to complete the combinational cycle of the RO, the select pin must remain in this position for the duration of the system operation. According to the RO scenarios, the core oscillator structure of the desired number can be connected in parallel.

For modelling the chaotic maps given in Figs. 6, 7 and 8, the ready ip-core modules, presented by Altera and defined on IEEE 754 floating-point numbers, were used. For each chaotic system that was modelled, a 200 MHz input operating frequency obtained from PLL (phase-locked loop) was used. The common working principle of chaotic maps that are included into the system by being transformed into the block circuit element presented in Fig. 5B is summarized as follows: in the modelled chaotic systems, $x_{(n)}$ values are fed back to the system via a mux for iteration $x_{(n+1)}$. The 32-bit $x_{(n)}$ values that are fed back to the system are simultaneously

Table 1 Theoretical details of selected chaotic maps

Chaotic map	Equation	Control par.	Initial values	Output range
Quadratic map	$x_{n+1} = r - x_n^2)$	$r \in [0.074]$	$r = 1.69 - x_0 = 0.196$	$(-2, 2)$
Logistic map	$x_{n+1} = r x_n(1 - x_n)$	$r \in [3.57, 3, 9]$	$r = 3.9 - x_0 = 0.631$	$(0, 1)$
Bernoulli shift map	$x_{n+1} = \begin{cases} bx_n - a, & x_n \geq 0 \\ bx_n + a, & x_n < 0 \end{cases}$	$a = 1$ $b \in [1.4, 2.0)$	$b = 1.95 - x_0 = 0.578$	$[-1, 1]$

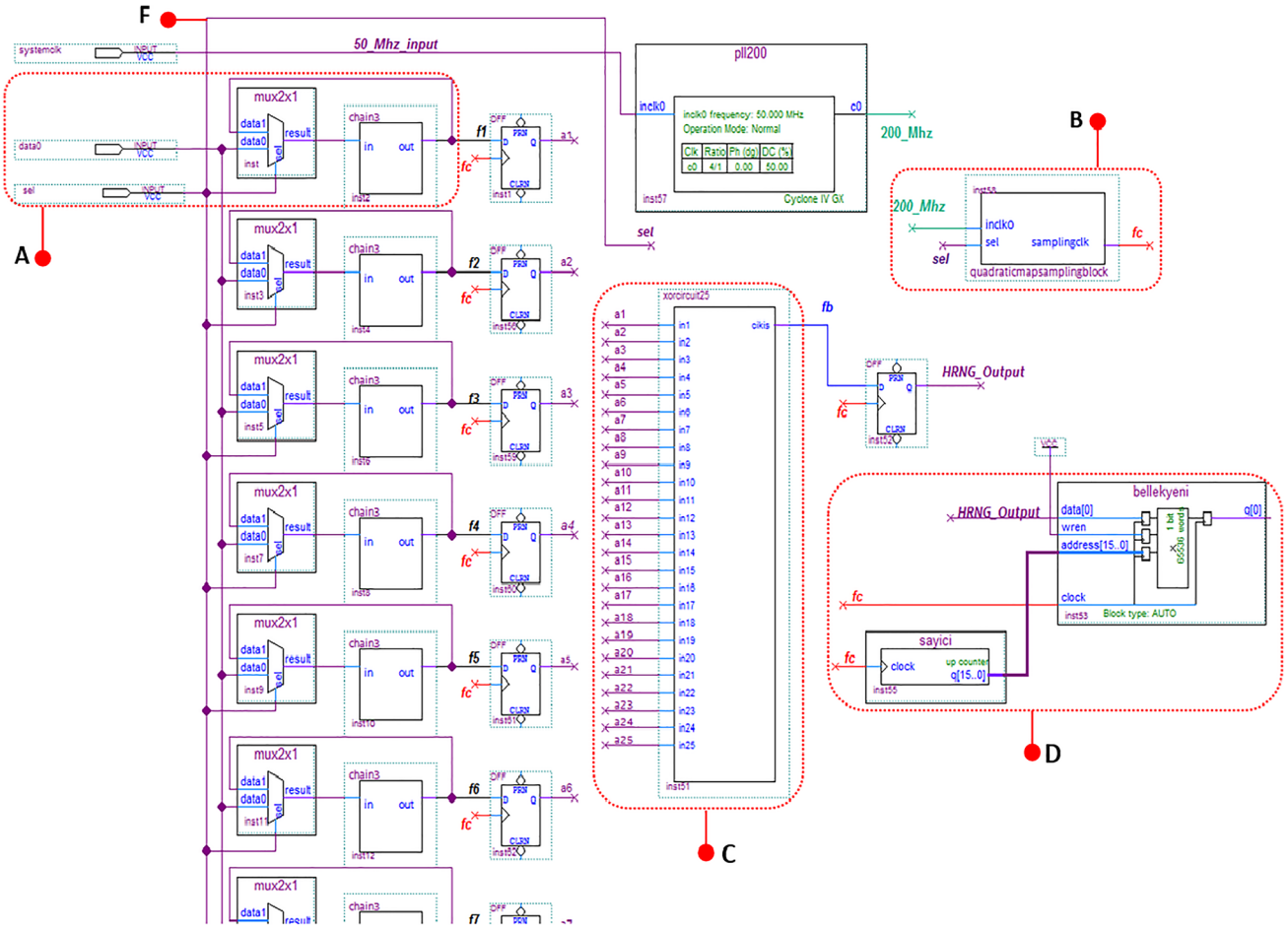


Fig. 5 Hybrid RNG's hardware design for quadratic map and (25,3) RO scenario

applied to the input of a simple comparator circuit, and in return to these values, chaotic signals are obtained at the bit level. Differently, in the first iteration, the select pin (sel) of the mux must remain at logic '0' level for a very short time (12 clock signal) in order to make the application of the $x_{(0)}$ seed value to the system possible. Thus, the $x_{(0)}$ seed value connected to mux's data0 input is transferred to the output, and the chaotic system becomes active.

For each scenario in the hybrid system, the parallel-connected oscillator and the chaotic systems were connected to a common select pin in Fig. 5E for synchronization. Thus, the oscillators and the chaotic system will operate synchronously by being triggered together. Immediately thereafter, by pulling the select pin to logic '1' level, the oscillators and the chaotic system will continue to work together in the feedback position. The design details of the chaotic systems modelled are as follows, respectively.

5.1 Quadratic Map

In the system, for the quadratic map, one mux (lmp_mux), three constant (lmp_constant), one multiplication (altfp_mult0), one subtraction (altfp_sub0) and one comparator (altfp_compare0) modules were used. The values of $x_{(n)}$ given as input to the system in each iteration were obtained by feeding back to the system by passing through

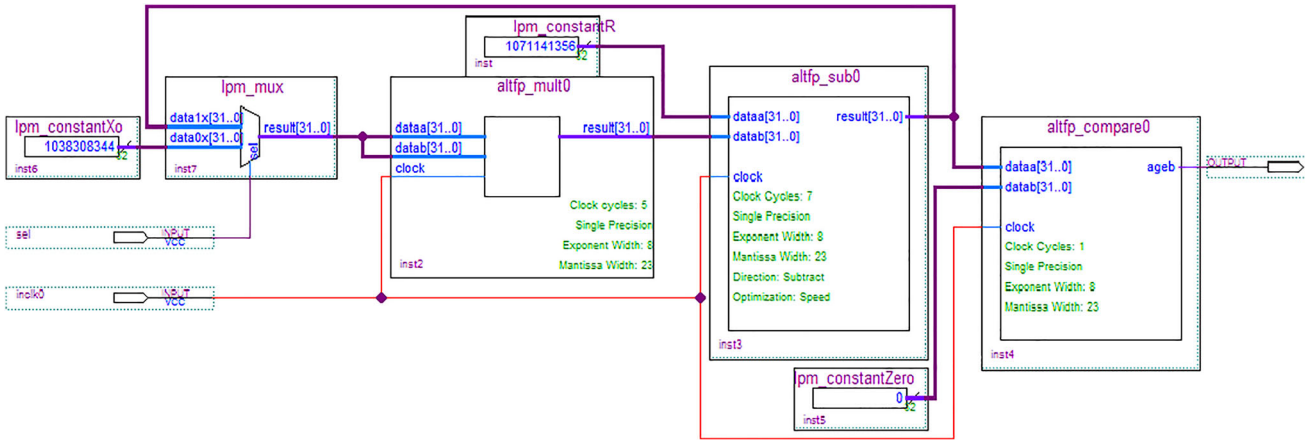


Fig. 6 Hardware modelling of the quadratic map

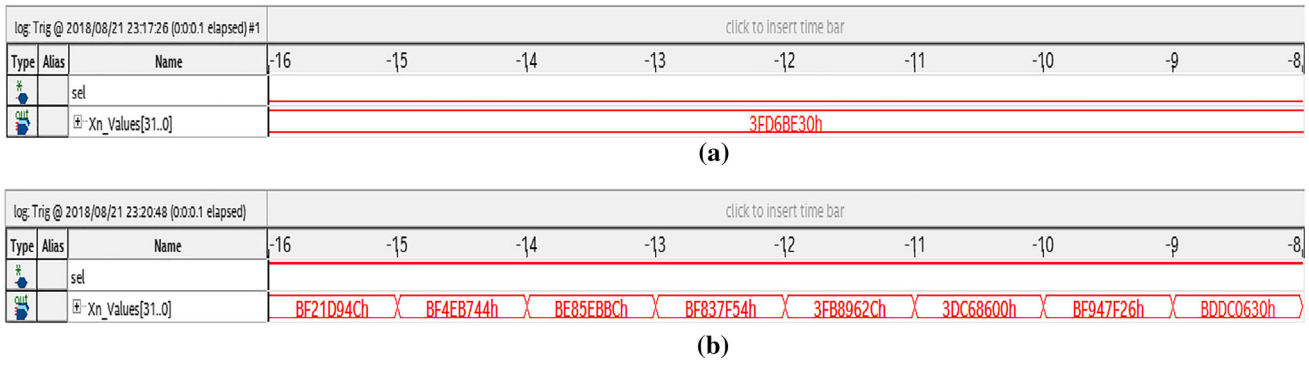


Fig. 7 **a** Initial state of the chaotic system, **b** random numbers generated from the chaotic system

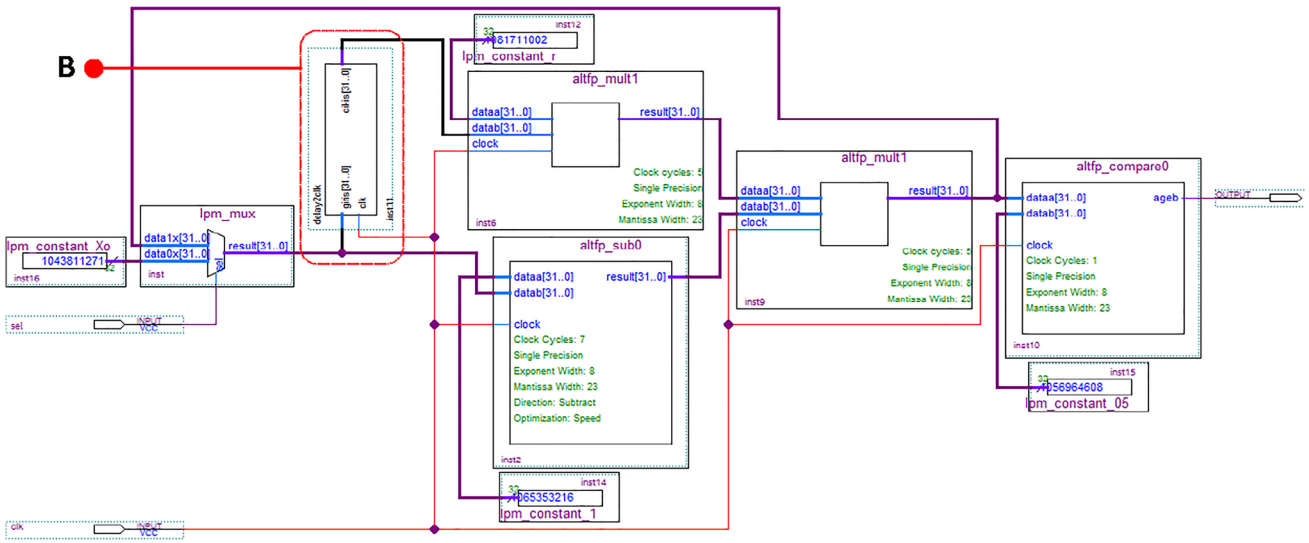


Fig. 8 Hardware modelling of the logistic map

the multiplication and subtraction operations for the next iteration. The $x_{(n)}$ values in the feedback position were also applied to the input of a comparison circuit. In the system, an $x_{(n)}$ value is calculated with a total delay of 12 clock signals, after five clock multiplication and seven clock subtraction operations, respectively. Therefore, the change interval of the chaotic signal applied to the input of the comparator circuit corresponds to standard 12 clock signals for each iteration. This calculation time is also the same for other chaotic systems.

At the output of the comparator circuit, the bit-level chaotic sampling signal generated correspondingly to the current $x_{(n)}$ value occurs with one clock delay. In the hybrid system, the time required to generate a chaotic signal applied to the clock input of flip-flops for sampling is a total of 13 clock signals. This interval is the same for the selected chaotic maps. The system parameter r and the initial value $x_{(0)}$ for the quadratic map given in Fig. 6 are selected as 1.69 $(3FD851EC)_H$, 0.111 $(3DE353F8)_H$, respectively. The output value $x_{(1)}$ that is created correspondingly to these values by the chaotic system for the first iteration and applied to the input of the comparator circuit is 1.6776 $(3FD6BE30)_H$. Some of the other chaotic output values generated by the system are as presented in Fig. 7b.

5.2 Logistic Map

In the system, for the logistic map, one mux (lmp_mux), four constant (lmp_constant), two multiplication (altfp_mult0, altfp_mult1), one subtraction (altfp_sub0) and one comparator (altfp_compare0) modules were used. The $x_{(n)}$ values, given as input to the system in each iteration, are first passed through multiplication and subtraction operations simultaneously for the next iteration. The obtained result is transferred to the simple comparator circuit at the same time while being fed back to the system by being passed through the multiplication operation again. In the system, the delay time needed to generate an $x_{(n)}$ value is a total of 12 clock signals because the multiplication (altfp_mult0) and subtraction (altfp_sub0) modules are independent of each other and they work simultaneously. The delay time for the chaotic signal generated, correspondingly to the value of 32 bits $x_{(n)}$, which is transferred to the input of the comparator circuit (altfp_compare0), is again one clock. Therefore, the time needed to generate a chaotic signal in the hybrid system for sampling is a total of 13 clock signals.

The response times of the multiplication (altfp_mult0) and subtraction (altfp_sub0) circuits, which operate independently of each other in the system, are five and seven clock signals, respectively. Therefore, in order to avoid the synchronization problem that may occur in the system, a delay circuit was used as shown in Fig. 8B. The delay circuit will transfer the values of $x_{(n)}$ applied to its input to the output of the multiplication circuit with a delay of two clock signals. Thus, the synchronization problem was eliminated by synchronizing the response time of the multiplication circuit (altfp_mult0) with the response time of the subtraction circuit (altfp_sub0).

5.3 Bernoulli Shift Map

In the system, for the Bernoulli shift map, two multiplexer (lmp_mux, lmp_mux1), four constant (lmp_constant), one multiplication (altfp_mult0), one subtraction (altfp_sub), one addition (altfp_add) and two comparator (altfp_compare0, altfp_compare1) modules were used. The Bernoulli map given in Table 1 consists of two separate sets of equations depending on the state of $x_{(n)}$ values. Therefore, in contrast to the other chaotic maps modelled, it was required to use an additional comparator circuit (altfp_compare0) in the chaotic system for Bernoulli. The main purpose of the comparator circuit applied to the input of the system as in Fig. 9A is to determine which equation set will be used to calculate $x_{(n)}$ values in the system.

In the system in the first step, $x_{(n)}$ values are multiplied with the system parameter b. Then, addition or subtraction operations are performed according to the state of the signal obtained from the comparator circuit applied to the system input. For each iteration, to activate only one of the addition or subtraction circuits, the clock enable inputs (clk_en) of the addition (altfp_add0) and subtraction (altpf_sub0) circuits are used. The output of the comparator circuit at the input of the system was applied directly to the clock enable input of the subtraction circuit. However, it was applied to the clock enable (clk_en) input of the addition circuit (altfp_add0) by taking its' 'not' as in Fig. 9B. Thus, for any iteration in the system, power consumption and complexity were reduced due to the redundant operation of both circuits.

In any iteration, only one of the equations given for Bernoulli in Table 1 is used depending on the state of the $x_{(n)}$ values in the system. This distinction in the hybrid system is important for the operation of the system because the values of $x_{(n)}$ obtained from the true equality must be fed back to the system and chaotic signals must be obtained corresponding to these values. For this, a 2×1 extra mux (lmp_mux1) was added to the output of the system as in Fig. 9C. The data0 and data1 inputs of the 2×1 mux are connected to the outputs of the addition and subtraction circuits, respectively. The time required to generate an $x_{(n)}$ value in the system is 12 clock signals. Therefore, the change of the $x_{(n)}$ inputs applied to the input of the mux is periodic 12 clock signals. In order for the mux to operate synchronously with the variations connected to its own inputs, the select pin (sel) was obtained from the delay circuit as shown in Fig. 9D.

The output of the comparator circuit at the system input and the operating frequency of the system were applied to the input of the delay circuit created by data flow design techniques. The logic signal that is taken from the output of the comparator circuit (lmp_compare0) by the delay circuit is given to the select pin of the mux, after a time of 1/12 of the operating frequency. Thus, the $x_{(n)}$ values obtained at the output of the mux are transferred to the system and the final comparator circuit correctly for iteration $x_{(n+1)}$. The time needed to obtain a chaotic signal in correspondence with the $x_{(n)}$ values transferred to the comparator circuit is 13 clock signals in total.

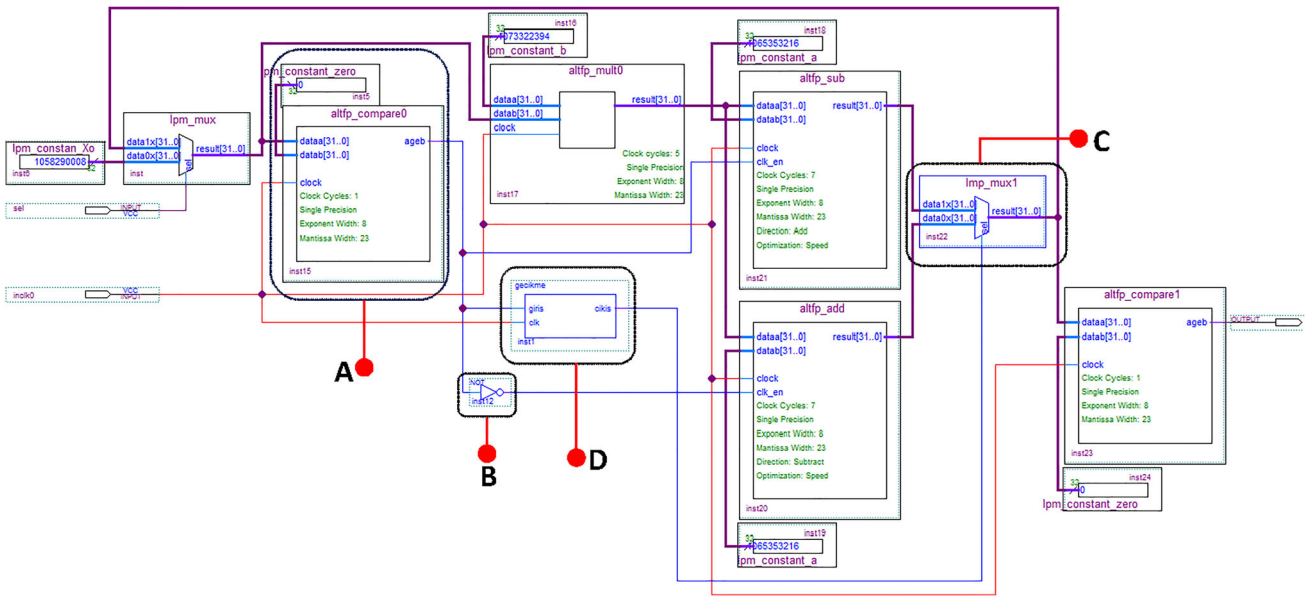


Fig. 9 Hardware modelling of the Bernoulli shift map

The $x_{(n)}$ values representing the outputs of the chaotic maps given in Table 1 are in the 32-bit floating-point number format in the system. These numbers were compared with the threshold value by implementing to the input of the comparator circuit finally. Thus, in correspondence with these numbers, non-periodic signals were generated at the bit level to be used for sampling. Equalities in Eq. 1 were used to obtain the chaotic sampling signals. The threshold value parameter q in Eq. 1 was selected as 0 for the Bernoulli and quadratic map and as 0.5 for the logistic map. For test purposes, the random numbers obtained from the system were stored in the 16-bit counter-supported memory unit, consisting of 1-bit width and 65,536-bit depth, as in Fig. 5D. A real-time implementation of the hybrid system according to the given information is presented in Fig. 10:

$$b_n = \begin{cases} 1, & x_{n+1} \geq q \\ 0, & x_{n+1} < q \end{cases} \tag{1}$$

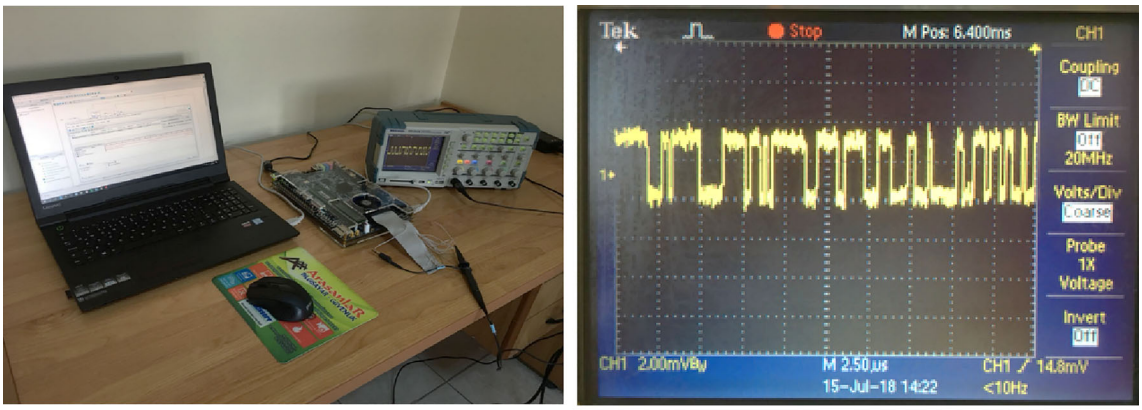
6 Experimental Results

6.1 Findings

One of the most critical steps in any RNG design is the verification process of the cryptographic competencies of the generator. The main focus point of this process is the statistical verification of the randomness quality of the numbers obtained from generators, which is of great importance for cryptography. Various statistical test tools developed for this purpose are available. These tools are useful for detecting the samples that are not random by examining the probabilistic distributions of numeric sequences subject to testing. However, there is not any finite set of test tools, which can fully verify the cryptographic competencies of any RNG. For this reason, it is possible to obtain different results for the same random number subject to the test from different test groups [1, 4, 18].

NIST 800-22 test package, which has a more powerful and more compelling structure in terms of reliability level than other test groups, was used in the study. The NIST test suite consists of 15 separate subtest criteria in itself to evaluate the statistical characteristics of number sequences that have a sufficient length. For each test criterion, α and p value parameters of random numbers included in the NIST 800-22 software suite are considered. The p value parameter, which is regarded as the success criterion, varies in the range of [0–1]. It is assumed that for the value '1' of the p value parameter, the number sequence is perfectly random and that for the value '0' it is not random. For the cryptographic implementations, α parameter corresponding to the typical significance level is in the range of [0.001–0.01]. For the numbers subjected to the test for each test criterion, it is a necessary condition that the p value parameter is greater than the parameter α [34, 40].

In the NIST 800-22 statistical test package (suit), the minimum length of the random number sequence tested for each test criterion is variable. For some test criteria, the p value parameter cannot be measured in cases where the sequence of random numbers is not long enough. For this reason, in order to obtain healthy results from the test suite, the size of the random number sequences subject to testing should be constant and long enough. N is the constant size of the sequence of random numbers obtained from the generator and must be in the value range of minimum and maximum $10^3 < N < 10^7$.



(a) **(b)**

Fig. 10 **a** Hybrid system set-up, **b** real-time generated bits

Table 2 Hybrid RNG's NIST 800-22 statistical test results

Test criteria	Quadratic map		Logistic map		Bernoulli map	
	(25,3)	(114,3)	(25,3)	(114,3)	(25,3)	(114,3)
The frequency (monobit) test	0.845	0.542	0.381	0.169	0.235	0.448
Frequency test within a block	0.087	0.929	0.478	0.770	0.859	0.0936
The run test	0.616	0.840	0.213	0.412	0.911	0.234
Test for the longest run of ones in a block	0.723	0.155	0.600	0.820	0.211	0.480
The binary matrix rank	0.849	0.868	0.460	0.641	0.816	0.285
The discrete Fourier transform (spectral) test	0.322	0.959	0.874	0.532	0.555	0.528
Non-overlapping template matching	0.649	0.691	0.719	0.805	0.426	0.401
The overlapping template matching test	0.554	0.086	0.040	0.174	0.326	0.090
Maurer' s 'Universal Statistical' test	0.595	0.811	0.218	0.738	0.868	0.826
The linear complexity test	0.813	0.918	0.378	0.731	0.551	0.619
The serial test	0.878	0.334	0.297	0.560	0.892	0.062
	0.410	0.545	0.337	0.749	0.676	0.079
The approximate entropy test	0.782	0.375	0.372	0.362	0.246	0.673
The cumulative sums test	0.484	0.436	0.698	0.245	0.370	0.156

The NIST 800-22 statistical test results, conducted for the verification of the HRNG, hardware design of which is given for the different scenarios within the scope of the study, are presented in Table 2. For each scenario, the length of the random number sequence being tested is equal to the memory depth.

The basic idea in the implemented hybrid system is based on the non-periodicity of random numbers obtained from discrete-time chaotic maps used as the sampling component. Therefore, in the hybrid system, the scale index technique proposed by Benitez [41] was used in order to determine the degree of non-periodicity of the signals obtained from the chaotic maps. The test technique that can be applied to continuous- and discrete-time chaotic components is based on the continuous wavelet transform (CWT) and wavelet multiresolution analysis (MRA). In the scale index method, especially the ratio of the scalogram value in the absolute measure to the scalogram value in the least important measure is calculated. This ratio specifies the scale index, in other words, the degree of non-periodicity, and takes definitely a positive value if the signal is non-periodic [32, 41].

The scale index technique can be briefly described as follows in several steps [32, 41]:

- Being u time and s scale values, the continuous wavelet transform (CWT) of f is like in Eq. 2. The scalogram value of f is calculated as in Eq. 3:

$$Wf(u,s) := f, \Psi_{u,s} = \int_{-\infty}^{+\infty} f(t)\Psi^*_{u,s}(t)\mathrm{d}t, \tag{2}$$

$$S(s) := \left\| wf_{(u,s)} \right\| = \left(\int_{-\infty}^{+\infty} |Wf(u,s)|^2 \mathrm{d}u \right)^{1/2} \tag{3}$$

Table 3 Non-periodicity degree of chaotic components

	Quadratic map	Logistic map	Bernoulli shift map
Result	0.9647	0.9183	0.9361

- $S(s)$ is the energy of f calculated by continuous wavelet transform for the scale value s. The inner scalogram value of f for a scale of s value is calculated as in Eq. 4:

$$S^{\text{inner}}(s) := \left\| wf_{(u,s)} \right\|_{J(s)} = \left(\int_{c(s)}^{d(s)} |Wf(s,u)|^2 \mathrm{d}u \right)^{1/2}. \tag{4}$$

- The equality of $J(s) = [c(s), d(s)] \subseteq I$ expresses the maximum subrange in I. I must be long enough, for $J(s)$ not to be empty or too small. The length of $J(s)$ depends on the scale value s, so its inner scalogram values cannot be compared for different scale values. For this reason, the inner scalogram was normalized as in Eq. 5:

$$\bar{S}^{\text{inner}}(s) = \frac{S^{\text{inner}}(s)}{(d(s) - c(s))^{1/2}}. \tag{5}$$

- If the scalogram $S(s)$ is not too small for the $s > s_{\max}$ inequality, it is assumed that the signal is non-periodic for the range value of $[s_0, s_1]$. For the range value of $[s_0, s_1]$, the scale index value of f (i_{scale}) is calculated as in Eq. 6. In the inequality of $0 \leq i_{\text{scale}} \leq 1$, for i_{scale} value that is 0 or near 0, the chaotic system is periodic, while it is non-periodic for values 1 and near 1:

$$i_{\text{scale}} := \frac{S(s_{\min})}{S(s_{\max})}. \tag{6}$$

The non-periodicity degree of random numbers/signals obtained from the chaotic maps was measured by using the scale index technique, and the results are presented in Table 3. The NIST 800-22 test results of the random numbers obtained from the hybrid system are given in Table 2. When the test results are examined, it is observed that successful results are obtained for all the scenarios of the HRNG. This situation confirms the statistical properties of the hybrid system in terms of cryptography. The other hardware design parameters for (114,3) RO scenario of the hybrid system are presented in Table 4.

6.2 Evaluations

For the HRNG, the evaluation results obtained from the experimental set-up mechanism given in Fig. 10 are given in Tables 2, 3 and 4. For the TRNG or PRNG designs depending on the use of low-noise sources (jitter, metastability, etc.)

Table 4 Resource usage and power consumption of the designs for (114,3) oscillator architecture

Area/resource usage	Quadratic map (114,3)	Logistic map (114,3)	Bernoulli map (114,3)
Programmable logic element number	2011 (%1)	2158 (%1)	2771 (%1)
Flip-flop/register number	1376	1532	1815
Total pin	3	3	3
Embedded multiplier	6	14	9
Power consumption (s)			
Core dynamic thermal power cons.	1.25 mW	1.31 mW	11.75 mW
Core static thermal power cons.	123.76 mW	125.77 mW	125.87 mW
Input/output thermal power cons.	11.73 mW	11.73 mW	11.70 mW
Total thermal power cons.	136.74 mW	138.81 mW	149.31 mW

with entropy, the statistical requirement/competence must be provided cryptographically. When the results given in Table 2 are examined, it can be seen that this basic competence was provided in six separate scenarios for HRNG. This situation indicates that HRNG can be used in cryptographic applications.

In TRNG designs, in order to reduce the potential weaknesses depending on the statistical deficiency, post-processing techniques are generally used. However, the use of chaos, which is a powerful entropy source in the hybrid system, has eliminated the necessity of post-processing. The statistical results in Table 2 were obtained without using the post-processing technique in the hybrid system. This fact confirms the appropriateness of non-periodic sampling method based on chaos.

Although many of the chaotic systems are deterministic, these do not contain randomness/uncertainty and their outputs are predictable. Because after a certain iteration number, deterministic systems containing pseudo-randomness repeat themselves. This is not a desirable feature in cryptography where randomness is also considered a measure of unpredictability. However, unpredictable determinism is the most important feature that distinguishes chaotic systems from linear (deterministic) systems. In other words, in spite of simple deterministic structures, chaotic systems can perform randomly similar oscillations in an infinite number of non-

Fig. 11 Time-dependent simulation of the random numbers generated in the system

Table 5 Comparison of the HRNG with other RO-based RNGs known in the literature

References	Bit rate (Mbps)	Number of ROs	Number of inverters in each RO	Operating frequency (MHz)
[7]	4.77	5, 10, 25	3	200
[8]	100	25	3	100
[9]	2.5	114	13	40
[27]	2.0	110	3	40
[32]	20.45	5, 10, 25	3	450
[42]	2.17	128	3	50
HRNG	15.4	25, 114	3	200

repeating orbits for a specific interval value in the case of chaos.

The test results given in Table 3 show that the chaotic system trajectories used as the sampling component of HRNG are non-periodic. This also confirms that the systems for which mathematical definitions are given in Table 1 for input parameters are in case of chaos. Therefore, the test technique indicates that the source of randomness, which cannot be verified by the statistical testing tools of sampling inputs, is cryptographically reliable. It also shows us that the sampling inputs obtained from the chaotic system trajectories making non-periodic oscillations are unpredictable for the long period due to the behaviour of chaotic systems.

In the HRNG design, chaotic systems with the least hardware complexity were used to obtain non-periodic sampling signals. This case has provided that a one-bit non-periodic random sampling signal for ROs is obtained at a shorter cycle time than modelled chaotic systems. In response to the 200 MHz clock signal applied to the input of chaotic systems, this cycle is a total of 43 clock pulses for sinusoidal iterator in [7], whereas this is only 13 clock pulses for each chaotic system modelled within the scope of the study. In other words, a true random number is generated in every 0.065 μs from the system. According to the simulation results obtained by real-time operation of HRNG, the change of bit-level random numbers is shown in Fig. 11.

The high-oscillating RO outputs in the hybrid system have been reduced to the production rate of non-periodic signals obtained from chaotic systems. The choice of chaotic systems with little hardware complexity has reduced the limiting effect of sampling time in the hybrid system on the bit production rate of the generator. The non-periodic signals obtained with a shorter cycle time ensured obtaining much more successful results compared to the [7] in the bit generation speed rate that is one of the important evaluation criteria for cryptographic RNGs. In terms of average bit production rate, HRNG has high performance compared to other known oscillator-based studies in the literature. The comparison results are given in Table 5.

Output bit rates for the [8, 9, 27, 32, 42] given in Table 5, periodic signal is used for sampling in the system. When the data in Table 5 are analysed, the output bit rate for [32], in which the non-periodic signals are used as the post-processing technique contrary to HRNG, is 20.45 Mbps. This situation results from the high frequency (450 MHz) of the clock signal applied to the input of the chaotic system modelled. The HRNG is consistent with [7] at the point of usage of non-periodic signals. For this reason, the frequency of the clock signal used for chaotic systems is set to 200 MHz for the contribution of the chaotic systems used in the proposed method and the accuracy of the comparison criterion. In terms of hardware, the operating frequency and the calculation time required for any iteration step are decisive in terms of the output bit rate of a chaotic system. Therefore, for the HRNG with lower calculation time, it is possible to obtain better results with respect to output bit rate than [32] at higher operating frequencies.

The significant shortcomings in terms of system security of TRNG proposed by Wold and Tan [8] were clearly pointed out in [10, 11]. In particular, according to Sunar and Stinson [9], it was emphasized that the number of oscillators in the system was reduced uncontrollably and this situation caused serious entropy loss in the system. It was stated that no post-processing techniques were used to mask the loss of entropy, making the system cryptographically unsafe. The same situation is true to the study in [7, 32], in which the (10,3)

and (5,3) RO scenarios were used for obtaining a simpler architecture. However, the use of chaos-based non-periodic post-processing inputs in [32] masked the loss of entropy in the system. In [10], it was stated that the deterministic randomness, mostly caused by the sequential behaviour of the system, played an important role on the statistical success of TRNG. As a result of the deterministic randomness occurring, it is clearly stated that system outputs can be estimated or the system can be easily manipulated from outside. Within the scope of the study, the HRNG architecture obtained by the inclusion of chaos, which is a cryptographically powerful entropy source for the same TRNG architecture, is cryptographically more reliable than the [7, 8].

Chaotic maps based on simple mathematical definitions were preferred as deterministic component of HRNG. Thus, the resource demand and energy consumption of the hybrid system, which has reduced complexity due to chaos, have also been reduced. When the data in Table 3 were examined, the results obtained from the hybrid system for the (114,3) RO scenario in which the Bernoulli shift map is applied are worse according to the other scenarios in terms of hardware. The main reason for this case is that although the Bernoulli shift map is one-dimensional, it is due to its bimodal structure consisting of two separate equations, unlike the other chaotic systems in Table 1. Although the calculation time of the equation sets is equal, extra hardware circuit elements (mux, delay circuit, comparator, etc.) are used in the system in order to feed back the outputs of the correct equation set. This case has increased the HRNG's hardware resource demand and associated energy consumption. However, the hardware resource demand for HRNG's worst-case scenario is 58% less than [7] for the (25.3) oscillator scenario in which the sinusoidal iterator was used. This case facilitates the applicability of HRNG, which reduced resource demand and simplified design architecture, on partial devices. Furthermore, it was emphasized that the main disadvantage of the proposed hybrid system in [7] is the power consumption due to the chaotic component, but no statistical data sharing was made at this point. For HRNG, we believe that along with reduced hardware complexity, better results will be obtained at the point of energy consumption as well as functionality compared to [7].

7 Conclusions

The HRNG, of which hardware implementation is given in the study, meets the cryptographic qualifications for six scenarios in accordance with the results obtained. It was seen that for important evaluation criteria such as non-periodicity of chaotic systems, safety, energy consumption, applicability and bit production rate provide the design objectives of HRNG. In terms of the evaluation criteria shown above, for the scenarios in which the quadratic map is used, the results obtained from the hybrid system are better than the other scenarios. Furthermore, the use of chaotic components made the hybrid system more robust to aggressive tampering and environmental changes.

The number of programmable logic elements needed to model chaotic maps in the hybrid system could be seen as a disadvantage. However, proposed HRNG is easy to control with a single push button on the FPGA. Therefore, it can be stopped and re-operated in terms of energy saving for applications whose power consumption is a problem. Chaotic systems can be used to increase the entropy of the system within the design cycles of TRNGs. We believe that chaotic systems with low complexity will contribute to achieving more successful results for real-time applications.

References

1. Özkaynak, F.: Kriptolojik Rasgele Sayı Üreteçleri. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 8-2, 2015 (2016) (**in Turkish**)
2. Koç, Ç.: Cryptographic engineering. Springer, Berlin (2009)
3. Palacio-Luengas, L.; Pichardo-Méndez, J.L.; Diaz-Méndez, J.A.; et al.: PRNG based on skew tent map. Arab. J. Sci. Eng. (2018). https://doi.org/10.1007/s13369-018-3688-y
4. Ozkaynak, F.: Cryptographically secure random number generator with chaotic additional input. Nonlinear Dyn. **78**(3), 2015–2020 (2014). https://doi.org/10.1007/s11071-014-1591-y
5. Özkaynak, F.; Özer, A.B.; Yavuz, S.: Kaos Tabanlı Yeni Bir Blok Şifreleme Algoritması. BİLDİRİLER KİTABI 108 (2011) (**in Turkish**)
6. Tuna, M.: Kaotik sistemler ve FPGA tabanlı kaotik osilatörlerin gerçek rasgele sayı üretimindeki (GRSÜ) önemi üzerine bir araştırma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi* (2018) (**in Turkish**)
7. Tuncer, T.; Avaroğlu, E.; Türk, M.; Özer, A.B.: Implementation of non-periodic sampling true random number generator on FPGA. J. Microelectron. Electron. Compon. Mater. **44**, 296–302 (2014)
8. Wold, K.; Tan, C.H.: Analysis and enhancement of random number generator in FPGA based on oscillator ring. In: International Conference on Reconfigurable Computing and FPGAs, pp 385–390 (2008)
9. Sunar, B.; Martin, W.J.; Stison, D.R.: A provably secure true random number generator with built-in tolerance to active attacks. IEEE Trans. Comput. **56**(1), 109–119 (2007)
10. Bochard, N.; Bernard, F.; Fischer, V.; Valtchanov, B.: True-randomness and pseudo-randomness in ring oscillator-based true random number generators. Int. J. Reconfig. Comput. **2010**, 879281 (2010)
11. Fischer, V.: A closer look at security in random number generators design. In: International Workshop on Constructive Side-Channel Analysis And Secure Design. Springer, Heidelberg, pp 167–182 (2012)
12. Rodríguez-Orozco, E.; García-Guerrero, E.; Inzunza-Gonzalez, E.; López-Bonilla, O.; Flores-Vergara, A.; Cárdenas-Valdez, J.; Tlelo-Cuautle, E.: FPGA-based chaotic cryptosystem by using voice recognition as access key. Electronics **7**(12), 414 (2018)
13. Özkaynak, F.; Özer, A.B.; Yavuz, S.: Cryptanalysis of a novel image encryption scheme based on improved hyperchaotic sequences. Opt. Commun. **285**(24), 4946–4948 (2012)

14. Özkaynak, F.; Özer, A.B.: A method for designing strong S-boxes based on chaotic Lorenz system. Phys. Lett. A **374**(36), 3733–3738 (2010)
15. Lambić, D.: A novel method of S-box design based on discrete chaotic map. Nonlinear Dyn. **87**(4), 2407–2413 (2017)
16. Sbiaa, F.; Kotel, S.; Zeghid, M.; Tourki, R.; Machhout, M.; Baganne, A.: High-level implementation of a chaotic and AES based crypto-system. J. Circuits Syst. Comput. **26**, 1750122 (2017)
17. Stoyanov, B.; Kordov, K.: Novel secure pseudo-random number generation scheme based on two tinkerbell maps. Adv. Stud. Theory Phys. **9**, 411–421 (2015)
18. de la Fraga, L.G.; Torres-Pérez, E.; Tlelo-Cuautle, E.; Mancillas-López, C.: Hardware implementation of pseudo-random number generators based on chaotic maps. Nonlinear Dyn. **90**(3), 1661–1670 (2017)
19. Dabal, P.; Pelka, R.: A chaos-based pseudo-random bit generator implemented in FPGA device. In: IEEE 14th International Symposium on Design and Diagnostics Of Electronic Circuits And Systems (DDECS) (2011)
20. Valtierra, J.L.; Tlelo-Cuautle, E.; Rodríguez-Vázquez, Á.: A switched-capacitor skew-tent map implementation for random number generation. Int. J. Circuit Theory Appl. **45**(2), 305–315 (2017)
21. Cicek, I.; Pusane, A.E.; Dundar, G.: A novel design method for discrete time chaos based true random number generators. Integr. VLSI J. **47**(1), 38–47 (2014)
22. Khanzadi, H.; Eshghi, M.; Borujeni, S.E.: Design and FPGA implementation of a Pseudo random bit generator using Chaotic maps. IETE J. Res. **59**(1), 63–73 (2013)
23. Wang, Y.; Liu, Z.; Ma, J.; He, H.: A pseudo random number generator based on piecewise logistic map. Nonlinear Dyn. **83**(4), 2373–2391 (2016)
24. Sahari, M.L.; Boukemara, I.: A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption. Nonlinear Dyn. **94**(1), 723–744 (2018)
25. García-Martínez, M.; Campos-Cantón, E.: Pseudo-random bit generator based on multi-modal maps. Nonlinear Dyn. **82**(4), 2119–2131 (2015)
26. François, M.; Grosges, T.; Barchiesi, D.; Erra, R.: Pseudorandom number generator based on mixing of three chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **19**(4), 887–895 (2014)
27. Schellekens, D.; Preneel, B.; Verbauwhede, I.: FPGA vendor agnostic true random number generator. In: Proceedings of 16th International Conference on Field Programmable Logic and Applications- FPL (2006)
28. Kohlbrenner, P.; Gaj, K.: An embedded true random number generator for FPGAs. In: Proceedings on ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays (FPGA 2004). ACM, pp 71–78 (2004)
29. Golić, J.D.: New methods for digital generation and postprocessing of random data. IEEE Trans. Comput. **55**(10), 1217–1229 (2006)
30. Dichtl, M.; Golić, J.D.: High-speed true random number generation with logic gates only. In: Proceedings on Cryptographic Hardware and Embedded Systems—CHES 2007, LNCS 4727. Springer, Berlin, pp 45–62 (2007)
31. Tuncer, T.: Implementation of duplicate TRNG on FPGA by using two different randomness source. Elektronika ir Elektrotechnika **21**(4), 35–39 (2015)
32. Avaroğlu, E.; Tuncer, T.; Özer, A.B.; Ergen, B.; Türk, M.: A novel chaos-based post-processing for TRNG. Nonlinear Dyn. **81**, 189–199 (2015)
33. Avaroğlu, E.; Tuncer, T.; Özer, A.B.; Türk, M.: A new method for hybrid pseudo random number generator. J. Microelectron. Electron. Compon. Mater. **4**(4), 303–311 (2014)
34. Tuncer, S.A.: Real-time random number generation with RO-based double PUF. Informacije MIDEM **48**(2), 121–128 (2018)
35. Avaroğlu, E.; Koyuncu, İ.; Özer, A.B.; Türk, M.: Hybrid pseudo-random number generator for cryptographic systems. Nonlinear Dyn. **82**, 239–248 (2015)
36. Alhadawi, H.S.; Zolkipli, M.F.; Ismail, S.M.; Lambić, D.: Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map. Cryptologia (2019). https://doi.org/10.1080/01611194.2018.1548390
37. Jiteurtragool, N.; Masayoshi, T.: Hybrid random number generator based on chaotic oscillator. Presented at the Management and Innovation Technology International Conference (MITicon) (2016)
38. Merah, L.; Ali-Pacha, A.; Said, N.H.; Mamat, M.: Pseudo random number generator based on the chaotic system of Chua's circuit, and its real time FPGA implementation. Appl. Math. Sci. **7**(55), 2719–2734 (2013)
39. Garipcan, A.M.; Erdem, E.: Hardware design and analysis of ring oscillator based noise source for true random number generators. Presented at the International artificial intelligence and data processing symposium (IDAP'18), Malatya, Turkey (2018)
40. Garipcan, A.M.; Erdem, E.: Donanım Tabanlı Trivium Akış Şifreleme Algoritmasının FPGA Ortamında Gerçekleştirilmesi", Fırat Üni. Müh. Bil. Dergisi, **29**(2), 119–130 (2017) (**in Turkish**)
41. Benìtez, R.; Bolos, V.J.; Ramìrez, M.E.: A wavelet-based tool for studying non-periodicity. Comput. Math. Appl. **60**, 634 (2010)
42. Tuncer, T.: The implementation of chaos-based PUF designs in field programmable gate array. Nonlinear Dyn. **86**(2), 975–986 (2016)