IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 6, 2021, accepted July 15, 2021, date of publication July 26, 2021, date of current version August 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099534

A Fully Digital True Random Number Generator With Entropy Source Based in Frequency Collapse

RONALDO SERRANO[1], (Student Member, IEEE), CKRISTIAN DURAN[1], (Graduate Student Member, IEEE), TRONG-THUC HOANG[1,2], (Graduate Student Member, IEEE), MARCO SARMIENTO[1], (Student Member, IEEE), KHAI-DUY NGUYEN[1], (Student Member, IEEE), AKIRA TSUKAMOTO[2], KUNIYASU SUZAKI[2,3], (Member, IEEE), AND CONG-KHA PHAM[1], (Member, IEEE)

[1]Department of Computer and Network Engineering, The University of Electro-Communications (UEC), Tokyo 182-8585, Japan
[2]National Institute of Advanced Industrial Science and Technology (AIST), Tokyo 135-0064, Japan
[3]Technology Research Association of Secure IoT Edge Application Based on RISC-V Open Architecture (TRASIO), Tokyo 101-0022, Japan

Corresponding author: Ronaldo Serrano (ronaldo@vlsilab.ee.uec.ac.jp)

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Grant JPNP16007.

ABSTRACT All cryptography systems have a True Random Number Generator (TRNG). In the process of validating, these systems are necessary for prototyping in Field Programmable Gate Array (FPGA). However, TRNG uses an entropy source based on non-deterministic effects challenging to replicate in FPGA. This work shows the problems and solutions to implement an entropy source based on frequency collapse in multimodal Ring Oscillators (RO). The entropy source implemented in FPGA pass all SP800-90B tests from the National Institute of Standards and Technology (NIST) with a good entropy compared to related works. The TRNG passes all NIST SP800-22 with and without the post-processing stage. Besides, the TRNG and the post-processing stage pass all tests of Application notes and Interpretation of the Scheme (AIS31). The TRNG implementation on a Xilinx Artix-7 XC7A100TCSG324 FPGA occupies less than 1% of the resources. This work presents 0.62 μs up to 9.92 μs of sampling latency and 1.1 Mbps up to 9.1 Mbps of bit rate throughput.

INDEX TERMS TRNG, NIST, AIS31, frequency collapse.

I. INTRODUCTION

Random Number Generators (RNG) conform a crucial part of cryptographic systems. RNG circuits are implemented as in-core key generation, where the data can be ciphered with such random keys. A TRNG implements an entropy source to the circuit, hence reducing its predictability for known-key attacks [1]. The NIST provides a group of constraints and tests for TRNG implementations within a crypto-core [2]. For implementation, such TRNG needs to be implemented inside of the system, and use secure channels to the crypto-core. Moreover, the NIST offers some tests to evaluate the quality of entropy sources. Due to the implementation needs, the RNG circuit needs to be included within the same system that contains the cryptography engine. However, the quality of the TRNG circuit depends mainly on the entropy source, which is often found external to the digital system.

The entropy sources exploit the random characteristics of noise, sampling the physical phenomenon, and finally applying a processing stage to deliver random numbers. Different physical phenomenons are used to capture the entropy of the noise. Peyroula *et al.* [3] shows an entropy source based in Resistive Random-Access Memory (RRAM). The entropy is extracted in the bitset operations. However, the characterization of the V_{set} and R_t is necessary to extract the best entropy in this source. Li *et al.* [4] present a TRNG based on metastability implemented in FPGA. The entropy source is based on metastability extraction of cross-coupled NAND gates. This design requires the construction of symmetrical NAND gates by using forced Look-Up Tables (LUTs) configurations. Although it can be constructed with special FPGA configurations, the FPGA manufacturers usually adopt

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

some technologies to decrease the probability of metastable events. Sarkisla and Ergun [5] show two methods to optimize area in TRNG based on metastability in Flip-Flops (FF). Using several RO as clock sources, the jitter generates random delays for the different clocks, and then the metastability is sampled using a FF. However, an external clock and manual placement of the ROs are necessary for this design. Mathew *et al*. [6] describe a micro random number generator combining multiple entropy sources, fabricated in a 14 nm FinFET CMOS technology, using three independent self-calibrating all-digital entropy sources with cross-coupled inverted pairs. Also, the sources are coupled with an XOR feedback shift register. Lu *et al*. [7] presents a TRNG based on extremely small Vernier interval. The jitter is quantified by a Time-to-Digital Converter. Besides, the presented TRNG is robust against Process, Voltage, and Temperature variations. Several works implement multimodal ROs as an entropy source [8]–[10]. The implementation of the TRNG depends heavily on the used technology. Some TRNG extracts the entropy source from an external source, requiring analog design or external components to the SoC [3], [11], [12]. All-digital sources can be implemented in both ASIC and FPGA with some modifications to the entropy circuits depending on the available resources. As for internal digital sources, the TRNG may present security problems due to entropy manipulation attacks [13], [14]. Particularly, multimodal ROs have resistance against power side-channel attacks [8].

In this work, we implement the entropy source based on frequency collapse in multimodal RO. We describe the implementation in FPGA by utilizing regular LUT blocks to construct the multimodal and common ROs. A Phase and Frequency Detector (PFD) circuit is used to compare the output of a multimodal RO and a regular RO. The TRNG counts the clock cycles from the multimodal RO during the comparison. Although the entropy source and the PFD can be implemented inside the same circuit, manual placement of the ROs is necessary. Moreover, there is a trade-off between the throughput of random numbers and the quality of randomness, which is determined by the length of the ROs. We check the randomness of FPGA implementation using the NIST SP800-22 and AIS31 tests, with the last reporting a Shannon entropy of 7.996012. The quality of the entropy source is tested using the NIST SP800-90B entropy test with 0.892911 of minimum entropy normalized. In addition, a post-processing stage based on a linear corrector is implemented. The FPGA implementation occupies 58 LUTs and 21 FFs in the TRNG, and the post-processing stage uses 4 LUTs, which represents less than 1% of the resources available. The TRNG presents 0.62 μs up to 9.92 μs of latency and 1.1 Mbps up to 9.1 Mbps of bit rate throughput.

The remainder of this paper is organized as follows. Section II present the analytical model of the entropy source. Section III shows the architecture of TRNG. Section IV shows the latency and bit rate of the implementation, and presents the test results of NIST SP800-90B, NIST SP800-22, and AIS31. Finally, section V concludes the paper.

II. ENTROPY SOURCE

A. ANALYTICAL MODEL

The multimodal RO have two option in the construction of the entropy source. The first option is to have the number of edges even. In this case, the odd and even pulses travel in different paths through the ring. The rising and falling delays in CMOS inverters are different and change with the process variations. The frequency collapse occurs by the difference of rising and falling delays, and the jitter presents the two paths [15]. In the second option, the number of edges can be odd. In comparison to even edges, the pulses change the path in each oscillation. In this way, the difference between rising and falling delays is inherently reduced. The time of any pulse to arrive at the output (T_{pulse}) in the RO with odd edges is as follows:

$$T_{pulse} = \sum_{i=nk}^{nk+k/3} t_{edge} + \sum_{j=1}^{nk+k/3} \delta_{(n,j)} - \sum_{i=1}^{nk} \delta_{(n,i)} \tag{1}$$

In this equation, k and n represent the number of edges and cycles, respectively. Also, the $\delta_{(n,i)}$ and $\delta_{(n,j)}$ denotes the jitter generated for the thermal noise in the even an odd stages. Finally, t_{edge} represents the typical delay of the logic gates in the all stages. The model of three edges (2) demonstrates the variance increasing linearly with the number of cycles, when $\delta(n, j) = \delta(n, i) \sim N(0, \sigma^2)$ [8].

$$T_{pulse} = N(\frac{k}{3} t_{edge}(2nk + \frac{k}{3})\sigma^2) \tag{2}$$

However, in FPGA implementation, the inverter gate is reproduced with a LUT, introducing undesirable delays in the RO. Besides, the effects of the distance between each LUT and the interference of external signals take a greater role in the frequency collapse. In this way, the FPGA implementation introduces other undesirable delays, increasing the systematic mismatch compared to ASIC implementation.

B. IMPLEMENTATION

Fig. 1.a) shows the schematic of three multimodal RO. The entropy source architecture uses three NAND gates to increase oscillation frequency compared to conventional RO. Each NAND gate has a delay stage, which determines the frequency of the RO. The enable generates a pulse in each edge. The approximate entropy is increased according to the number of the delay stages. However, if the delay stages are increased, the frequency collapse takes more time to happen [8]. Fig. 1.b) illustrates the frequency collapse caused for the accumulation jitter in the multimodal RO. In the initial event, the three pulses are generated separately with a spacing of 120 degrees. The jitter present in the RO generates a break of the balance of the three pulses. Before the collapse event, the accumulation of jitter causes the break of phase balance of the pulses. In the moment of collapse event, the shock of

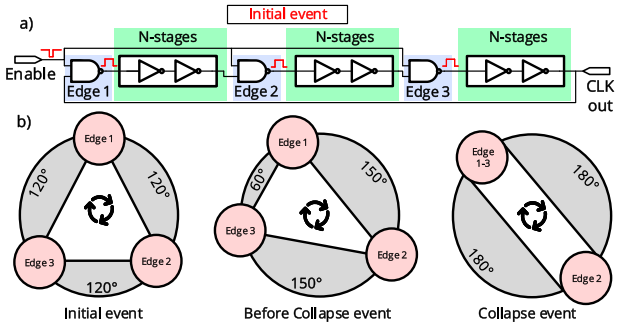

FIGURE 1. a) Three edge multimodal RO. b) Frequency Collapse in phase domain.

two pulses cause come back of balance, and the oscillation frequency of the RO decreases.

The physical effect in the multimodal RO to extract the entropy does not depend on the external clock compared to the metastability entropy sources [4], [5]. The odd edges analytical model stated in (2) determines that the frequency collapse depends only on the systematic mismatch and random jitter. If the systematic mismatch is small, the noise predominant generates a good entropy source. However, if the systematic mismatch is significant, the time for the frequency collapse tends to be constant, reducing the entropy source quality. The multimodal RO with high systematic mismatch variations is not used for random number generation, but rather for Physical Unclonable Functions (PUF) [16].

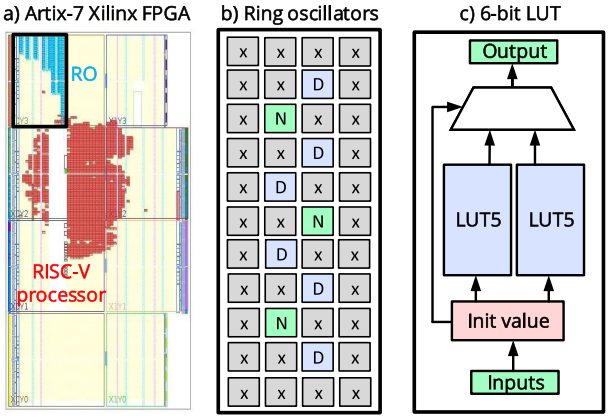

FIGURE 2. a) Layout in FPGA of the entropy source using different delays stages. b) Placement restrictions of the entropy sources implemented. c) Block diagram of the LUT6 in Xilinx FPGA.

To reduce the systematic mismatch, we configure the placement, routing, and LUT constraints according to the entropy source model. We reduce this according to the analytical model of three multimodal RO proposed by Yang *et. al* [8], and including the systematic mismatch introduced by the FPGA implementation. We perform a hardware mismatch fixation to distribute the RO elements in the FPGA evenly. Fig. 2 depicted techniques applies to reduce the systematic mismatch in FPGA. Fig. 2.a) shows the entropy sources with different N-stages of delay implemented in FPGA, along with a RISC-V processor for testing and debugging. The arrangement of the RO edges needs to be configured in the FPGA in a special way to guarantee the entropy of the circuit. The RO edges occupy several FPGA cells for both positioning and blockages, which is represented in blue. Fig. 2.b) illustrate the placement cell used in three edges multimodal RO. In FPGA, each slice has different LUTs to implement a logic cell. Nevertheless, using the same LUT is mandatory to reduce systematic mismatch variations. In this case, we used the B6LUT by Nand (N) and Delay (D) cells. In addition, the routing delay contributes to the systematic mismatch present in the FPGA implementation. A solution to reduce the routing delays is the manual placement of the LUTs of the RO in the same form as the figure. In this form, when we applied a manual route, the distance between LUT and LUT needs to be similar as possible. However, the analytical model based on implementing the entropy source does not consider the intervention of the other signals. Thus, the adjacent slices are restricted. Fig. 2.c) shows the content of LUT6 in a Xilinx FPGA [17]. The Xilinx FPGA presents differents types of slices. Nevertheless, the implementation of the entropy source only uses the SLICEL due to their different delay times. The multimodal RO is implemented in 6-bit Look-Up Table (LUT6) in a Xilinx Artix-7 FPGA. A RAM-based function generator reproduces the LUT in FPGA with an initial value. For example, to reproduce an inverter gate, the initial value can take more values. However, the delay of the logic gate depends on the initial value. Consequently, the initial value to implement the entropy source must be the same for the logic cells used.

The quality of the entropy sources is evaluated using the SP800-90B test suite provided by NIST [18]. Determination of the Independent and Identically Distributed (IID) from the data generated from the entropy source is fundamental to prevent overestimating the test suite [1]. The physical phenomenon used to generate the frequency collapse depends on the noise and systematic mismatch. Therefore, the data generated from the entropy source is non-IID according to the analytical model.

III. TRUE RANDOM NUMBER GENERATOR (TRNG)

A. TRNG CORE

Fig. 3 presents the TRNG architecture. The architecture needs a reference oscillation to capture the frequency collapse. The reference oscillation is generated by a conventional RO used as a reference (RO REF). Fig. 3.a) illustrates the TRNG architecture. The PFD takes the signal generated of RO RNG and RO REF by indicating a frequency collapse phenomenon. The random number is generated in a 12-bit counter in the capture stage highlighted in blue. However, the four least significant bits (LSB) are truncated by mitigating the error that introduces the method to the digitization of the entropy. This counter uses the clock generated by RO RNG. The enable of the counter is assert when the TRNG is initialized. To deassert the enable of the counter need the frequency collapse occurs, and the fourth bit of the counter is high to

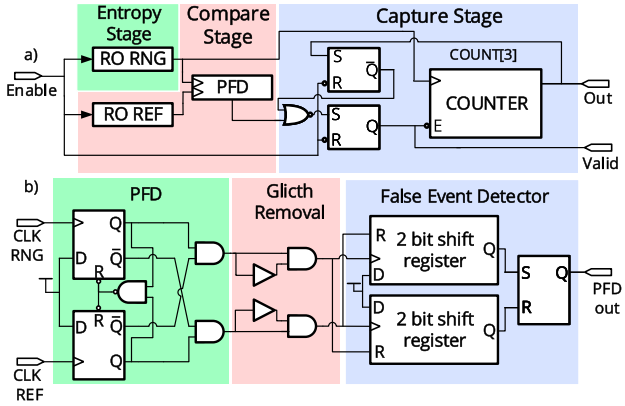

FIGURE 3. a) TRNG system block diagram implementation. b) PFD implementation.

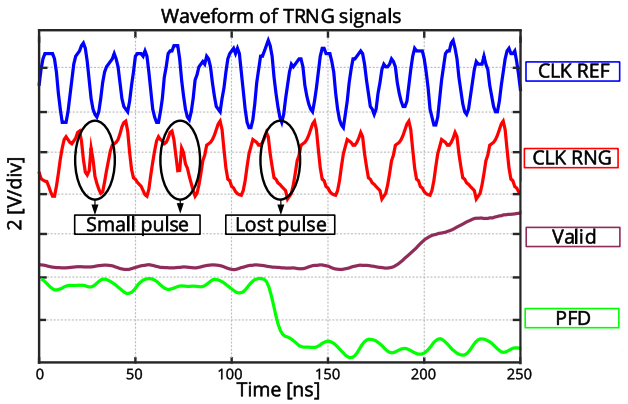

FIGURE 5. Measured waveform of the TRNG signals.

prevent false triggers of the PFD in the initial stages before the frequency collapse. After the frequency collapse, the *Valid* signal is triggered, indicating a random number in the Out port. Fig. 3.b) depicts the implementation of PFD used in the compare stage. The fully digital PFD is highlighted in green and consists of two FF, one NAND gate, and two AND gates. The input of the FF is connected to VDD, and the inputs are the clock generated for RO REF and RO RNG, respectively. When one clock changes to high, the FF charges and changes the output to high. The function of the gates is to prevent a false event when the two clocks are in phase. In the event of frequency collapse, the oscillation signal is deformed by the accumulation jitter in RO. The deformed signals generate glitches, causing false events in the PFD. A glitch removal circuit is implemented highlighted in red to prevent the false events for the comparison signals deformation. A 2 bit shift register highlighted in blue is implemented by evading other false events caused for the variations of the time response of the PFD.

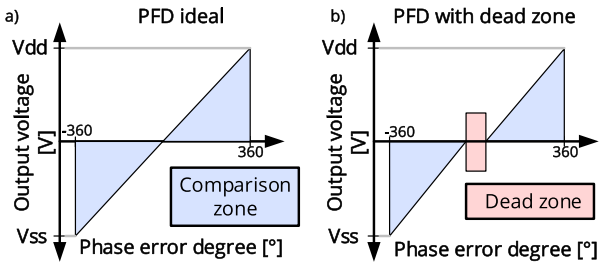

FIGURE 4. a) Ideal PFD. b) PFD with dead zone.

The fully digital PFD used in the TRNG architecture presents a dead zone in the frequency comparison range. The dead zone is a small difference in the phase of the inputs that the PFD does not detect, caused by the delay time of logic components and the feedback path of the FF. Fig. 4 illustrates the phase error in the ideal PFD on the left and digital PFD implemented in the right. The dead zone causes errors when the oscillation frequency of RO RNG and RO REF are in phase. The truncation of the four LSB in the 12 bit counter removes the errors introduced by the dead zone. In addition, the number of delay stages in RO REF is approximately 2/3 of all delay stages in the RO RNG, ensuring when the collapse frequency occurs, the frequency of the oscillation RO REF is upper of RO RNG. The power consumption of RO REF is reduced without harm to the functionality of the frequency reference.

Fig. 5 illustrates the signals CLK RNG, CLK REF, PFD, and *Valid* signals from the TRNG implemented in Xilinx Artix-7 FPGA. The signal generated in the entropy source is highlighted in red. This signal is captured at the moment of frequency collapse. Before the collapse, a small pulse indicates the imbalances of the pulses in multimodal RO. When the jitter accumulation achieves the limit by two pulses crash, the small pulse is lost, indicating the frequency collapse. At this moment, the PFD indicates the REF frequency is higher compared to the RNG frequency, and the logic of the capture stage triggers the valid signal.

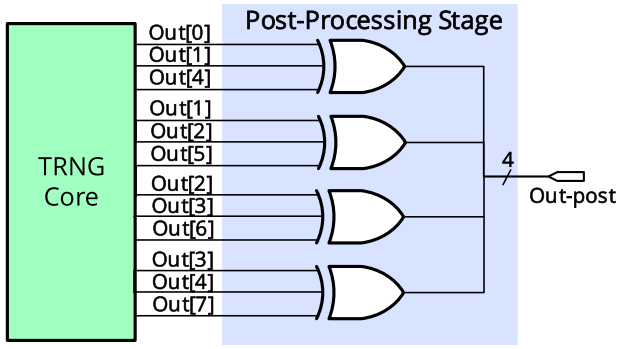

FIGURE 6. Post-processing stage implementation.

B. POST-PROCESSING

The methodology for implement the entropy source described in Section II reduces the systematic mismatch to improve the quality of the number generated. However, in modern TRNGs, the post-processing stage is an integral part. Fig. 6 shows a linear corrector used in the post-processing stage implemented, which uses to input the 8 bits generate of the TRNG. The systematic mismatch with the methodology

presented influences more in the LSB. The post-processing stage masks the most significant bits (MSB) with the LSB through an XOR operation to reduce the systematic mismatch influence in the LSB bits, generating a final output of 4 bits.

IV. RESULTS

This section presents results to evaluate the quality of entropy source based on frequency collapse in multimodal RO with several tests. First, the entropy source's quality is evaluated using the estimators described in NIST SP800-90B [19]. Second, the latency and bit rate of the TRNG is presented. Next, the TRNG without and with post-processing stage is tested applying the NIST SP800-22 [2]. Lastly, the AIS31 [20] test is applied to demonstrate the strength of the quality output of this work. All tests were implemented in a Xilinx Artix-7 FPGA to obtain the data.

TABLE 1. Non-IID results of the NIST SP800-90B test.

Non-IID estimators		1-bit		8-bit	
		P-value	Est.	P-Value	Est.
Most common value		0.5015	0.9956	0.0042	7.8566
Collision		0.5230	0.9351	—	—
Markov	P0	0.4989	0.9957	—	—
	P1	0.4989			
	P2	0.4989			
	P3	0.4989			
Compression		0.0244	0.8929	—	—
t-Tuple		0.517605	0.9501	0.0048	7.7188
Longest repeated substring		0.5008	0.9976	0.0048	7.3538
Multi most common		0.5007	0.9980	0.0040	7.9528
Lag prediction		0.5007	0.9980	0.0039	8.0000
Multi MMC prediction		0.5014	0.9959	0.0040	7.9503
LZ78Y prediction		0.5015	0.9957	0.0040	7.9506

A. ENTROPY SOURCE

In section II, the output data is classified as non-IID, according to the analytical model presented. Table 1 applying the non-IID track without a conditional component. The Collision, Markov, and Compression estimates tests are only applied to binary inputs. The minimum of all estimates in the non-IID test determines the minimum entropy initial (H_I). In this case, the minimum entropy is the estimation of the Compression test is $H_I = 0.892911 * 8 = 7.143286$. However, the entropy source estimation is calculated with a single and long-output sequence. The entropy source can generate a correlated sequence after restarts.

TABLE 2. Restart test results of the NIST SP800-90B test.

Non-IID estimators in restart test	8-bit row		8-bit column	
	P-value	Est.	P-value	Est.
Most common value	0.0043	7.8566	0.0043	7.8566
t-Tuple	0.0061	7.3538	0.0061	7.3538
Longest repeated substring	0.0048	7.7188	0.0041	7.9436
Multi most common	0.0040	7.9528	0.0041	7.9214
Lag prediction	0.0041	8.0000	0.0041	7.9218
Multi-MMC prediction	0.0040	7.9503	0.0041	7.9434
LZ78Y prediction	0.0040	7.9506	0.0040	7.9445

Table 2 shows the results to apply the restart test. The data used in this test, the entropy source is restarted 1000 times. For each restart, 1000 samples shall be collected directly from the entropy source. Two datasets are constructed concatenating in rows and columns. The entropy source passes the sanity check with a $\alpha = 0.01$. The minimum entropy in rows (H_R) and columns (H_C) are 7.353758. The results of the NIST SP800-90B with non-IID data is determinate of $H_{min} = min(H_R, H_C, H_I)$. The entropy source implemented presents a $H_{min} = 7.143286$. Finally, Table 3 shows the comparison of the minimum entropy of the sources with different physical phenomenons. The entropy source presents a 0.892911 of minimum entropy normalized without conditional components. The data is recollected in nominal conditions. Also, the entropy source implemented has eight delay stages per edge.

TABLE 3. Summary and comparison of minimum entropy.

	This work	[7]	[12]
Type of data	Non-IID	Non-IID	Non-IID
Number of bits	8	1	8
Minimum entropy normalized	0.892911	0.888864	0.720727
Type of entropy source	Noise	Noise	Radio isotope
Physical phenomenon	Frequency collapse	Jitter	Natural decay

B. STATISTICAL TESTS

This section explains the frequency collapse time to obtain the latency and the bit rate of the implementation. In addition, it presents the resources occupied in the TRNG implementation. Finally, a different statistical test is applied to determine the quality of the implemented TRNG.

Fig. 7 illustrates the occurrences in the time required for each frequency collapse. The number of samples is 5Mb.

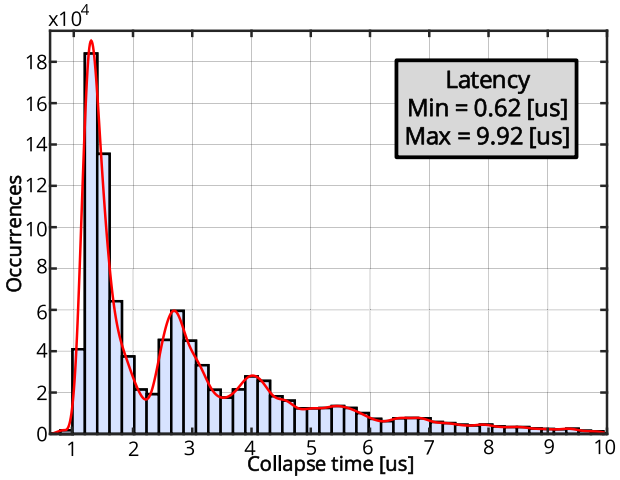

FIGURE 7. Time of the frequency collapse.

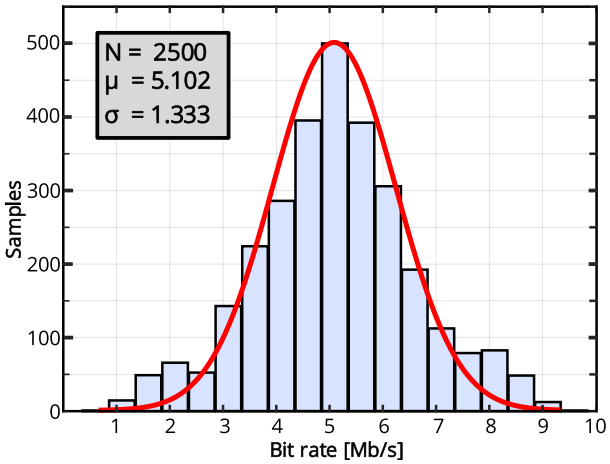

FIGURE 8. Bit rate of the TRNG implemented.

Because the nature of the physical phenomenon of the entropy source, the time necessary in each frequency collapse event occurs is aleatory. The minimum and maximum time for the event are 0.62 μs and 9.92 μs, respectively.

Fig. 8 shows the bit rate of the TRNG implemented. The bit rate of the TRNG implemented depends on the time for all collapses in the sequence generated. In this way, 2500 samples are taken, showing a mean of 5.102 and a deviation standard of 1.333, respectively. The range of bit rate using 3σ is 1.102 up to 9.109 Mbps.

Fig. 9 depicts the histogram of normalized data generated for the TRNG. The data present a mean $\mu = 0.498$ and deviation standard $\sigma = 0.288$. The ideal standard distribution presents a statistic parameter of $\mu = 0.5$ and $\sigma = 0.288$.

Table 4 shows the result of applying the NIST statistical test suite in a stream of 5MB. The significance level applied in the test is $\alpha = 0.01$, indicating that one would expect one sequence in 100 sequences to be rejected. Also, a P-value $\geqslant 0.01$ would mean that the sequence would be considered to be random with confidence of 99%. The TRNG passes all the tests with and without the post-processing stage. The test data are generated by Xilinx Artix-7 FPGA in nominal conditions.

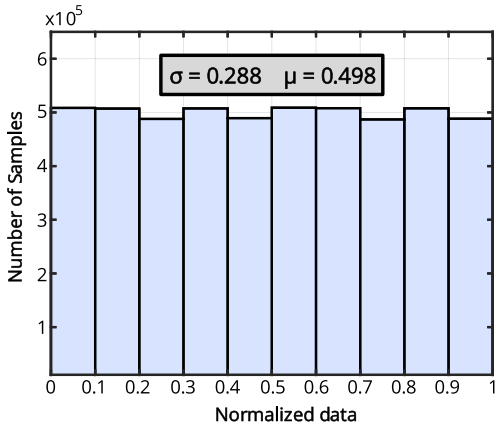

FIGURE 9. Histogram of TRNG-generated data used in the test with post-processing stage.

TABLE 4. Results of the NIST SP800-22 test.

Statistical test	TRNG without post-processing		TRNG with post-processing	
	P-value	Rate	P-value	Rate
Frequency	0.3342	95	0.5498	98
Block frequency	0.5544	99	0.3544	100
Cumulative sums	0.8434	94	0.7892	98
Cumulative sums	0.2142	96	0.7981	99
Runs	0.0909	99	0.2246	98
Longest run	0.7198	100	0.8422	100
Rank	0.6579	99	0.7954	98
FTT	0.1719	100	0.1976	100
Non-overlap template	PASS	PASS	PASS	PASS
Overlap template	0.1296	98	0.3248	99
Approx. entropy	0.7792	99	0.8922	100
Universal	0.0909	97	0.3445	98
Random excursions	PASS	PASS	PASS	PASS
Random excursions variant	PASS	PASS	PASS	PASS
Serial	0.5956	97	0.2430	98
Serial	0.8165	97	0.8514	99
Linear complexity	0.1088	97	0.5447	100

TABLE 5. Results of the AIS31 test.

Statistical test		Result
Number	Name	
T0	Disjointness	PASS
T1	Monobit	PASS
T2	Poker	PASS
T3	Run	PASS
T4	Long run	PASS
T5	Auto-correlation	PASS
T6	Uniform distribution	PASS
T7	Multi-nominal distributions	PASS
T8	Entropy	PASS

Table 5 shows the results of the AIS31 test. The AIS31 standard has two parts of applying the test [20].

TABLE 6. Performance summary and comparison in FPGA.

	This work	[9]	[4]	[5]	[7]
LUT	62	409^{+}	220	236	128*
FF	21	36	0	0	104*
Bit rate [Mb/s]	1.1 to 9.1	N/A	30	N/A	127
Entropy source	multi-modal RO	multi-modal RO	cross-coupled	RO	Vernier interval
Physical phenomenon	freq. collapsed	freq. collapsed	meta-stability	wake-up & shut-down	jitter
Post processing	Yes	Yes	Yes	Yes	No

* Resources are reported in slices, data are converted to LUT and FF.
$^{+}$ The authors implements four entropy sources.

The first part is denoted P1 (T0-T5), and it's to prove the output of the post-processing stage of the TRNG. The second part, P2 (T6-T8) is used to test the output of the noise source. The TRNG implemented passes all tests of AIS31, using a dataset of 5MB. Besides, test T8 indicates the average information content of the random number or the Shannon entropy. The Shannon entropy in the implementation is 7.996012.

Table 6 compiles the comparison of performance results of some TRNG reported in FPGA. The TRNG occupies 62 LUT and 21 FF with the post-processing stage, using less resources compared to other TRNG with other physical phenomenons. The bit rate of the TRNG implemented decreases due to the physical phenomenon of the entropy source. However, the architecture implemented does not depend on the external clocks, denoting robustness to clock attacks.

V. CONCLUSION

This work introduces a fully digital implementation of a TRNG in FPGA. The TRNG is based on the frequency collapse phenomenon using a multimodal RO and a regular RO. The implementations follow an analytical model of the frequency collapse in multimodal RO, proposing strategies to extract the most entropy possible. This TRNG was implemented in a Xilinx Artix-7 FPGA. The entropy source assumes a non-IID to apply the NIST SP800-90B. The entropy source passed all non-IID tests and the restart test. The minimum entropy normalized is 0.892311. The TRNG in FPGA passes all tests of NIST SP800-22 with and without the post-processing stage using a dataset of 5MB. Besides, the entropy source and TRNG with the post-processing stage pass all AIS 31 tests. The T8 sub-test of AIS31 shows a 7.996012 of Shannon entropy. The FPGA implementation of TRNG based on frequency collapse occupies 64 LUT and 21 FF, occupying less than 1% of the total Xilinx Artix-7 FPGA resources. The TRNG presents a 0.62 μs up to 9.92 μs of latency and 1.1 Mbps up to 9.1 Mbps of bit rate throughput.

REFERENCES

[1] S. Zhu, H. Chen, W. Xi, M. Chen, L. Fan, and D. Feng, "A worst-case entropy estimation of oscillator-based entropy sources: When the adversaries have access to the history outputs," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 152–159.
[2] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, and L. Bassham, *NIST Special Publication 800-22: A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications*, document NIST Special Publication 800-22, Apr. 2010.
[3] F. Pebay-Peyroula, T. Dalgaty, and E. Vianello, "Entropy source characterization in HfO_2 RRAM for TRNG applications," in *Proc. 15th Design Technol. Integr. Syst. Nanosc. Era (DTIS)*, Apr. 2020, pp. 1–2.
[4] C. Li, Q. Wang, J. Jiang, and N. Guan, "A metastability-based true random number generator on FPGA," in *Proc. IEEE 12th Int. Conf. ASIC (ASICON)*, Oct. 2017, pp. 738–741.
[5] M. A. Sarkisla and S. Ergun, "An area efficient true random number generator based on modified ring oscillators," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Oct. 2018, pp. 274–278.
[6] S. K. Mathew, D. Johnston, S. Satpathy, V. Suresh, P. Newman, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu, G. Chen, and R. K. Krishnamurthy, "μrNG: A 300-950 mV, 323 Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 51, no. 7, pp. 1695–1704, May 2016.
[7] Y. Lu, H. Liang, L. Yao, X. Wang, H. Qi, M. Yi, C. Jiang, and Z. Huang, "Jitter-quantizing-based TRNG robust against PVT variations," *IEEE Access*, vol. 8, pp. 108482–108490, 2020.
[8] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester, "16.3 A 23Mb/s 23pJ/b fully synthesized true-random-number generator in 28nm and 65nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 280–281.
[9] J. Cartagena, H. Gomez, and E. Roa, "A fully-synthesized TRNG with lightweight cellular-automata based post-processing stage in 130nm CMOS," in *Proc. IEEE Nordic Circuits Syst. Conf. (NORCAS)*, Nov. 2016, pp. 1–5.
[10] H. Gomez, J. Arenas, and E. Roa, "Low-cost TRNG IPs," *IET Circuits, Devices Syst.*, vol. 14, no. 7, pp. 942–946, Oct. 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cds.201%9.0535
[11] B. Jun and P. Kocher, "The Intel random number generator," Cryptogr. Res., San Francisco, CA, USA, White Paper, 1999, pp. 1–8, vol. 27.
[12] S. Park, B. G. Choi, T. W. Kang, K. W. Park, J. J. Lee, S. W. Kang, and J. B. Kim, "Analysis of entropy estimator of true random number generation using beta source," in *Proc. 34th Int. Tech. Conf. Circuits/Syst., Comput. Commun. (ITC-CSCC)*, Jun. 2019, pp. 1–3.
[13] Y. Cao, V. Rozic, B. Yang, J. Balasch, and I. Verbauwhede, "Exploring active manipulation attacks on the TERO random number generator," in *Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Oct. 2016, pp. 1–4.
[14] A. T. Markettos and S. W. Moore, "The frequency injection attack on ring-oscillator-based true random number generators," in *Cryptographic Hardware and Embedded Systems—CHES*, C. Clavier and K. Gaj, Eds. Berlin, Germany: Springer, 2009, pp. 317–331.
[15] K. Yang, D. Blaauw, and D. Sylvester, "An all-digital edge racing true random number generator robust against PVT variations," *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 1022–1031, Apr. 2016.
[16] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "14.2 A physically unclonable function with BER <10-8 for robust chip authentication using oscillator collapse in 40nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2015, pp. 1–3.
[17] *Vivado Design Suite 7 Series FPGA Libraries Guide*, UG953, Xilinx, San Jose, CA, USA, 2012. [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2012_2/ug953-vivado-7series-libraries.pdf
[18] A. K. McKay. (2016). *SP800-90B Entropy Assessment*. [Online]. Available: https://github.com/usnistgov/SP800-90B_EntropyAssessment
[19] M. Sonmez, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," NIST SP, Gaithersburg, MD, USA, Jan. 2018, doi: 10.6028/NIST.SP.800-90b.
[20] W. Killmann and W. Schindler, "AIS 31: Functionality classes and evaluation methodology for true (physical) random number generators, version 3.1," Bundesamt fur Sicherheit in der Informationstechnik, Bonn, Germany, 2001.

RONALDO SERRANO (Student Member, IEEE) received the B.Sc. degree in electronics from the Universidad Industrial de Santander (UIS), Bucaramanga, Colombia, in 2020. He is currently a Research Assistant with The University of Electro-Comunications (UEC), Tokyo, Japan. His research interests include computer architecture, high-speed digital interfaces, and hardware for security.

CKRISTIAN DURAN (Graduate Student Member, IEEE) received the B.Sc. degree in electronics and the M.S degree in telecommunications from the Universidad Industrial de Santander (UIS), Bucaramanga, Colombia, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electronics engineering. He is also a Research Assistant with The University of Electro-Comunications (UEC), Tokyo, Japan.

TRONG-THUC HOANG (Graduate Student Member, IEEE) received the B.Sc. degree in electronics and telecommunications and the M.S. degree in microelectronics from the Vietnam National University Ho Chi Minh City—University of Science, Ho Chi Minh, Vietnam, in 2012 and 2017, respectively. He is currently pursuing the Ph.D. degree in information and network engineering with The University of Electro-Communications (UEC), Tokyo, Japan. He is a Research Assistant with the National Institute of Advanced Industrial Science and Technology (AIST), Tokyo.

MARCO SARMIENTO (Student Member, IEEE) received the B.Sc. degree in electronics from the Universidad Industrial de Santander (UIS), Bucaramanga, Colombia, in 2020. He is currently a Research Assistant with The University of Electro-Communications (UEC), Tokyo, Japan. His research interests include debugging and security for integrated systems.

KHAI-DUY NGUYEN (Student Member, IEEE) received the B.Sc. degree in electronics from The University of Science and Technology, The University of Danang, Danang, Vietnam. He is currently a Research Assistant with The University of Electro-Communications (UEC), Tokyo, Japan.

AKIRA TSUKAMOTO received the M.S. degree in computer science from Columbia University, New York. He is currently with the National Institute of Advanced Industrial Science and Technology (AIST). He has worked on products based on Cell/B.E. and ARM. His research interests include software engineering on a network, operating systems, and system security, and he is enthusiastic regarding any kind of technical development.

KUNIYASU SUZAKI (Member, IEEE) received the B.E. and M.E. degrees in computer science from the Tokyo University of Agriculture and Technology, and the Ph.D. degree in computer science from The University of Tokyo, Tokyo, Japan. He is currently a Senior Researcher with the National Institute of Advanced Industrial Science and Technology (AIST) and a Researcher with the Technology Research Association of Secure IoT Edge Application Based on RISC-V Open Architecture (TRASIO). His research interests include security on CPU, operating systems, and hypervisor.

CONG-KHA PHAM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Sophia University, Tokyo, Japan. He is currently a Professor with the Department of Information and Network Engineering, The University of Electro-Communications (UEC), Tokyo. His research interest includes the design of analog and digital systems using integrated circuits.