IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 21, 2022, accepted April 8, 2022, date of publication April 18, 2022, date of current version April 25, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3167690

A Robust and Healthy Against PVT Variations TRNG Based on Frequency Collapse

RONALDO SERRANO[1], (Graduate Student Member, IEEE),
CKRISTIAN DURAN[1], (Graduate Student Member, IEEE),
MARCO SARMIENTO[1], (Graduate Student Member, IEEE),
TRONG-THUC HOANG[1,2], (Graduate 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 in part by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP16007; and in part by the Activities of VLSI Design and Educational Center (VDEC), The University of Tokyo, in collaboration with NIHON SYNOPSYS G.K., Cadence Design Systems and Mentor Graphics.

ABSTRACT True Random Number Generator (TRNG) is used in many applications, generally for generating random cryptography keys. In this way, the trust of the cryptography system depends on the quality of the random numbers generated. However, the entropy fluctuations produced by external perturbations generate some false positives in the random sequence. These false positives can generate a disastrous scenario, depending on the application. This work presents the results of different tests to demonstrate the robustness and health of the TRNG based on frequency collapse. The TRNG passed all entropy tests provided for NIST SP800-90B and AIS31. The entropy test denotes a 0.9789 minimum normalized entropy and 7.998 Shannon entropy. In addition, the TRNG passes the health tests provided for NIST SP800-90B. The health test shows a number of identical values $I_v = 0\%$, $I_{v-1} < 0.004\%$ and a maximum cutoff value $MC_v = 10$ with $LMC_v = 13$ in the repetition count and adaptive proportion tests, respectively. The implementation passed all the statistical tests provided for NIST SP800-22 and AIS20. Besides, the implementation passes the different tests with Process, Voltage, and Temperature (PVT) variations. The TRNG is implemented in a $0.18\mu m$ General Purpose (GP) CMOS technology, occupying $25600\mu m^2$ with four entropy sources. Finally, the implementation presents a 7.3 until 9.2-Mb/s of bit rate, 0.56 until 1.88-mW of power consumption, and 77.2 until 204.3-pJ/bit of energy per bit using an entropy source with 16 and 2 delay stages, respectively.

INDEX TERMS TRNG, NIST, AIS, frequency collapse.

I. INTRODUCTION

Random Number Generator (RNG) is vital for cryptography systems. The specifications of the RNG need high throughput or energy efficiency, depending on the application in the system. However, the RNG must provide a minimum level of trust independent of the application. The level of trust in RNG circuits is usually measured using a statistical test. For example, the National Institute of Standards and Technology (NIST) and the German Federal Office for Information Security (BSI) provide the SP800-22 [1], and AIS20 [2] test, respectively. Nevertheless, the statistical tests are not enough to provide a good level of trust. The external perturbations create a variance of the minimum entropy, generating a false positive in the random sequences. The NIST SP800-90B provides an entropy test to determine the robustness of the entropy source [3]. The entropy test evaluates the minimum entropy in the entropy source, depending on the Independent and Identically Distributed (IID) assumption. Besides, the AIS31 presents an entropy test to verify and extract the Shannon entropy, using an additional stochastic model [4]. In another way, the health test controls the quality

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

of the TRNG to prevent disastrous scenarios in the random sequence. Typically, the health test measures only the entropy or the statistical results on random numbers generated into the chip [5]–[7]. However, a disastrous scenario can be generated only with a false positive in a small part of the sequence. The NIST 800-90B also provides tests for verifying the health of the entropy source, detecting fixed and repetition number generation scenarios in the entropy source [8]. The health of the entropy source is quantified using the number of identical and cutoff values in the repetition count and adaptive proportion tests.

The physical phenomenons in the entropy source and the different forms to digitize the entropy determine the affectation of the external perturbations. For example, some TRNG uses an external entropy source [9]–[12]. However, external sources can be affected by external perturbation. Also, the difficulty of external attacks is reduced. The analog-TRNG uses an entropy source based on the noise presented in some analog circuits [13]–[16]. However, an analog/time to digital converter is necessary to digitize the entropy, increasing the area of the TRNG. The different kinds of Random Access Memory (RAM) are used to generate a random number, using the time response or leakage current [17]–[22]. Nevertheless, these entropy sources present the problem of characterizing the time response to obtain good digitization, and temperature variations affect the leakage current in RAM-TRNG or the additional mask for the resistive RAM. The architectures of TRNG based on metastability present a high bit rate, digitizing by flip-flops or latches [7], [23]–[26]. Nonetheless, the metastability sources can be affected for PVT variations, resulting in low minimum entropy [23]. In this way, the TRNG based on metastability solves the problem of the minimum entropy using a pulse generator [7] or a multi-entropy source [25] to improve the quality of the entropy source. Finally, the jitter is used in some architectures of TRNG, digitizing the entropy with two forms. First, the entropy is digitized by a flip-flop, using two clocks with low and high frequency, respectively [27]–[30] or the self-timed in Chaotic Cellular Automata [31]. Second, the jitter is measured using a physical phenomenon caused for the jitter accumulation in a multi-modal Ring Oscillator (RO). The entropy is digitized using the time of a frequency collapse [32]–[36]. However, the mismatch can affect the frequency collapse, generating dependencies used for Physical Unclonable Function (PUF) applications [37].

This work is a continuation of the work presented in [32], when the problems and solutions to implement the TRNG based on frequency collapse in Field Programmable Gate Array (FPGA) is shown. The main contribution of the current work is the demonstration of the robustness and healthy of the TRNG based on the same physical phenomena in an ASIC implementation. In addition, we extend an analytical model for ASIC implementation of three-edges multi-modal RO used in the entropy source, based on the model presented for FPGA implementation. A suggestion for the layout of the TRNG is proposed based on the analytical model, reducing the mismatch in the entropy source. Besides, the relations between power and minimum entropy with the number of stages in the multi-modal RO are presented. The implementation passes the NIST SP800-22 and AIS20 statistical tests. The entropy source passed the NIST 800-90B, and the AIS31 test with 0.9789 of minimum normalized entropy and 7.998 of Shannon entropy, using 16-stages multi-modal RO. Besides, the entropy source passes the repetition count and the adaptive tests with $I_v = 0\%$, $I_{v-1} < 0.004\%$ and a $MC_v = 10$ with $LMC_v = 13$, respectively. The statistical, entropy and health tests are applied with PVT variations. The ASIC implementation occupies a $25600\mu m^2$ in a $0.18\mu m$ GP CMOS technology with four different entropy sources implemented. Also, the TRNG reports a 7.3 until 9.2-Mb/s of bit rate, 0.56 until 1.88-mW of power consumption, and 77.2 until 204.3-pJ/bit of energy per bit, using an entropy source with 16 and 2 delays stages, respectively.

The remainder of this paper is organized as follows. Section II describes the analytical model of the entropy source and the architecture of the TRNG implemented. Section III shows the results of the entropy test described in NIST SP800-90B and AIS31. Section IV describes the health test applied in the entropy source. Section V shows the results of the NIST SP800-22 and AIS20 statistical tests. Section VI illustrates the area, power, and bit rate of the TRNG implemented. In addition, the relations between power and minimum entropy with the number of stages in the multi-modal RO. Finally, section VII concludes the paper.

II. TRUE RANDOM NUMBER GENERATOR (TRNG)

A. TRNG CORE

This section relates the different parts of the TRNG implemented. The TRNG used a physical phenomenon denominated frequency collapse to obtain the random numbers. The frequency collapse occurs for the jitter accumulation in a Multi-modal RO. The random number generated represents the time necessary for the frequency collapse. Fig. 1. shows the block diagram of the architecture of the TRNG. The architecture is divided into three parts. First, the *Entropy Stage* highlighted in gray is the part when the frequency collapse occurs. In addition, the oscillation signal is used in the *Capture Stage*. Second, the *Compare Stage* highlighted in red compares the oscillation signal generated in the *Entropy Stage* with a reference signal, using a Phase and Frequency Detector (PFD). Finally, the *Capture Stage* highlighted in blue generates the random number, using an 8-bit counter with the clock provided for the *Entropy Stage*. The fourth bit of the counter is used to prevent a false trigger in the PFD.

Fig. 2 depicts the block diagram of the entropy source and the clock reference used in the TRNG implemented. The entropy source (*RO RNG*) consists of three edges multi-modal RO, and each edge has an N-stages of inverters with the respective delay (t_0). In another way, the clock reference is implemented using a conventional RO (*RO REF*). The stages define the initial frequency in the *RO RNG* and the length

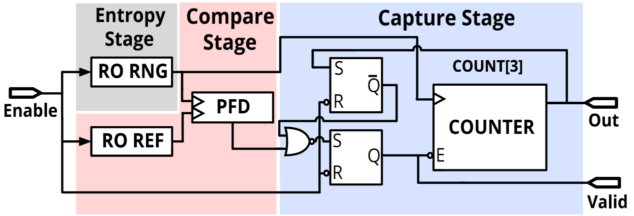


FIGURE 1. Block diagram of the TRNG [32].

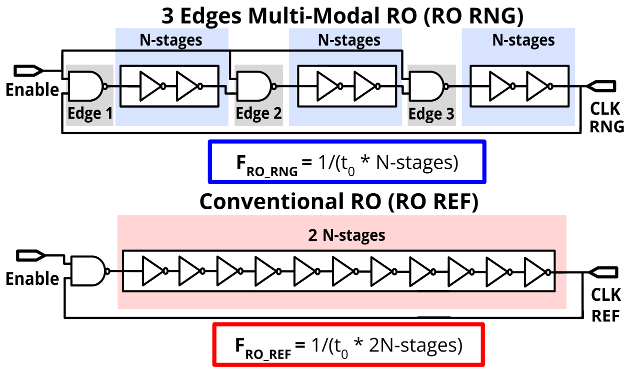


FIGURE 2. Entropy source and reference of the TRNG implemented [32].

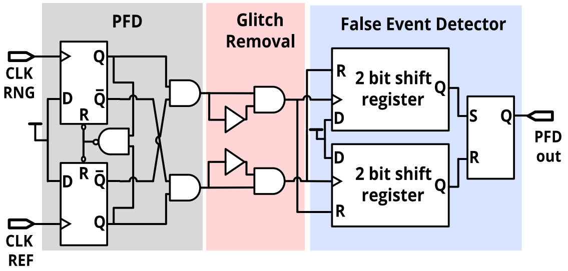


FIGURE 3. Compare stage of the TRNG implemented [32].

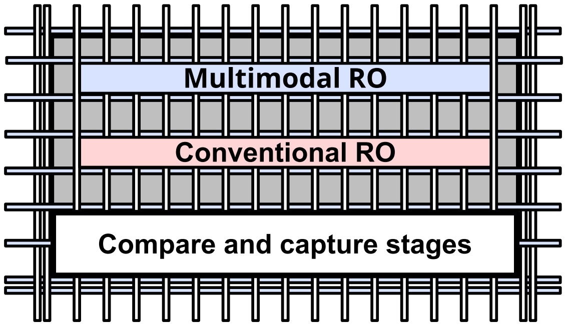


FIGURE 4. Suggestions proposed for the entropy source.

of the *RO REF*, respectively. The number of stages in the *RO REF* is two times the stages of one edge in the *RO RNG*, when one stage represents three inverter cells. In the initial stage, the *RO RNG* frequency is major compared to *RO REF*. However, the frequency collapse causes an approximate reduction of 1/3 of the frequency in the *RO RNG*. The frequency of *RO REF* is configured in 2/3 of the *RO RNG* before the frequency collapse, preventing false detection of the collapse in the *Compare Stage* [32]–[34].

Fig. 3 illustrates the block diagram of the PDF used in the *Compare Stage*. A digital PFD highlighted in gray detects the frequency variations of *CLK RNG* and *CLK REF*. When the frequency of *CLK RNG* is less than *CLK REF*, the PFD triggers a signal to stop the counter in the *Capture Stage*. However, a *Glitch Removal* highlighted in red is implemented to mitigate the false events generating for the small frequency differences between the entropy source and the clock reference before the frequency collapse. In addition, a *False Event Detector* highlighted in blue is implemented to eliminate the false positives generated in the start-up in the *RO RNG* and *RO REF*.

B. ANALYTICAL MODEL

In the odd edges multi-modal RO, the pulse generated in the odd edges changes the path in each oscillation, reducing the relation of rising and falling delays compared to an entropy source with even edges multi-modal RO [35]. An odd edges multi-modal RO analytical model is proposed for FPGA implementation [32], associating the frequency collapse between the noise and systematic mismatch. However, in ASIC implementation is possible to apply other strategies to reduce the mismatch, increasing the relationship between noise-mismatch. The time of any pulse generated by the edges arrives at the output (T_p) in the multi-modal (1).

$$T_p = \sum_{i=nk}^{nk+k/3}[t_0+\delta_{m(n,j)}] + \sum_{j=1}^{nk+k/3}[\delta_{n(n,j)}+\delta_{e(n,j)}] - \sum_{i=1}^{nk}[\delta_{n(n,i)}+\delta_{e(n,i)}] \tag{1}$$

$$T_p \sim N(\frac{k}{3}[t_0+C], (2nk+\frac{k}{3})[\sigma^2+\rho^2]) \tag{2}$$

where k and n represent the numbers of edges and cycles, respectively. In addition, t_0 exhibits the typical delay, and $\delta_{m(n,j)}$ is the delay generated for the mismatch in the inverter cell in the multi-modal RO. However, the mismatch after ASIC fabrication is a random constant. In this way, the delay generated for the mismatch is $\delta_{m(n,j)} = C$. Consequently, this constant is possibly reduced using different techniques. In another way, $\delta_{e(n,i)}$ and $\delta_{e(n,j)}$ are the noise in the even and odd delay cells that originated for the external signals. The inference can be approximate $\delta_e(n,j) = \delta_e(n,i) \sim N(0,\rho^2)$. When ρ represents the variance of the random occurrences of the external signals. Finally, the $\delta_{n(n,i)}$ and $\delta_{n(n,j)}$ are the jitter introduced in the delay cells. The quality of the numbers increases when the jitter is the majority of T_p. In conclusion, the three edges model (2) demonstrates the variance increase linearly with the number of cycles, when $\delta(n,j) = \delta(n,i) \sim N(0,\sigma^2)$ [33].

Fig. 4 illustrates the ASIC recommendations for reducing the undesirable effects related to the analytical model (2).

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

Non IDD estimators		V = 1.8[V] T= 25[°C]									
		CHIP #1		CHIP #2		CHIP #3		CHIP #4		CHIP #5	
		P-value	Est.	P-Value	Est.	P-value	Est.	P-value	Est.	P-value	Est.
Most common value		0.5009	0.9971	0.5011	0.9966	0.5010	0.9955	0.5014	0.9959	0.5015	0.9956
Collision		0.5031	0.9910	0.5004	0.9988	0.5006	0.9982	0.5002	0.9994	0.5001	0.9997
Markov	P0	0.5005	0.9984	0.5001	0.9997	0.5010	0.9956	0.5003	0.9991	0.5009	0.9974
	P1	0.4994		0.4994		0.4989		0.4997		0.4991	
	P2	0.5005		0.5001		0.5010		0.5003		0.5009	
	P3	0.4994		0.4994		0.4989		0.4997		0.4991	
Compression		0.0163	0.9898	0.0159	0.9958	0.0161	0.9927	0.0163	0.9898	0.0160	0.9942
t-Tuple		0.5038	0.9890	0.5035	0.9899	0.5026	0.9925	0.5013	0.9962	0.5008	0.9976
Longest repeated substring		0.5004	0.9986	0.5038	0.9890	0.5005	0.9984	0.5009	0.9974	0.5003	0.9991
Multi most common		0.5008	0.9975	0.5000	0.9997	0.5005	0.9984	0.5002	0.9994	0.5006	0.9982
Lag prediction		0.5001	0.9995	0.5004	0.9986	0.5005	0.9984	0.5008	0.9976	0.5006	0.9982
Multi MMC prediction		0.5006	0.9980	0.5009	0.9972	0.5014	0.9959	0.5003	0.9991	0.5014	0.9959
LZ78Y prediction		0.5009	0.9973	0.5010	0.9968	0.5014	0.9957	0.5007	0.9979	0.5015	0.9956

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

	V=1.8[V] T=25[°C]																			
Non IID estimators in restart test	CHIP #1				CHIP #2				CHIP #3				CHIP #4				CHIP #5			
	ROW		COLUMN		ROW		COLUMN		ROW		COLUMN		ROW		COLUMN		ROW		COLUMN	
	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.	P-value	Est.
Most common value	0.0041	7.93	0.0041	7.93	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0041	7.95	0.0041	7.95	0.0040	7.96	0.0041	7.95
t-Tuple	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0041	7.94	0.041	7.94	0.0040	7.96	0.0040	7.96
Longest repeated substring	0.0039	7.99	0.0039	7.99	0.0039	7.99	0.0039	7.99	0.0039	7.99	0.0039	7.99	0.0040	7.96	0.0039	7.99	0.0040	7.96	0.040	7.96
Multi most common	0.0039	7.97	0.0039	7.98	0.0039	7.99	0.0039	7.98	0.0040	7.96	0.0040	7.96	0.0039	7.98	0.0039	7.99	0.039	7.99	0.039	7.99
Lag prediction	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0040	7.96	0.0039	7.98	0.0040	7.96	0.0041	7.95	0.0040	7.96	0.0040	7.95	0.0040	7.95
Multi-MMC prediction	0.0039	7.98	0.0039	7.97	0.0039	7.98	0.0039	7.98	0.0039	7.98	0.0039	7.99	0.0040	7.96	0.0039	7.98	0.0040	7.96	0.0040	7.96
LZ78Y prediction	0.0039	7.97	0.0039	7.98	0.0039	7.99	0.0039	7.99	0.0039	7.98	0.0039	7.99	0.0039	7.98	0.041	7.95	0.0039	7.99	0.040	7.94

The δ_m is reduced using the fixed placement of the cells in the same row of the grid in the multimodal and conventional RO. The purpose is to obtain the same variations in each edge in the entropy source. In addition, a routing blockage highlighted in gray is applied in the multimodal and conventional RO, reducing significantly the (δ_e) introduced for the external signals. The compare and capture stages of the TRNG are placement and routing using the conventional digital flow.

III. ENTROPY TESTS

This section presents the entropy source results provided for the NIST SP800-90B and the AIS31 entropy test. The entropy results are presented with a non-IID assumption in the NIST SP800-90B. The NIST and AIS31 results are applied with PVT variations. The minimum and Shannon entropy are presented in the TRNG with 16 delay stages in the entropy source.

Table 1 shows the NIST 800-90B test results with process variations.The data is recollected without a conditional component in nominal conditions (VDD = 1.8[V] and 25[°C]). The first step to determine the quality of the entropy sources is to classify the random number generated in IID or non-IID. The entropy source used in the TRNG depends on the jitter generated by the thermal noise. In addition, the systematic mismatch influences the random numbers generated, according to the analytical model. However, an overestimation of the minimum entropy (H_{min}) is generated for false positives in the IID assumption [38], [39]. In this way, the entropy source is classified with a non-IID assumption for the systematic mismatch in the collapse time. The minimum value of the estimators determines the initial minimum entropy (H_I) in the non-IID test. The entropy source implemented present a $H_I = 0.9890 * 8 = 7.9120$. However, the H_I is determined with a long sequence, generating possible correlations.

Table 2 shows the results of the SP800-90B restart test. The restart test measures the data when the entropy source is restarted 1000 times. One thousand samples are collected directly in the entropy source at each restart time. The entropy

TABLE 3. Results of the AIS31 test.

Entropy test AIS31	PVT	CHIP # 1			CHIP #2			CHIP #3			CHIP #4			CHIP #5		
	V[V]	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0
	T[°C]	-10	25	125	-10	25	125	-10	25	125	-10	25	125	-10	25	125
Uniform distribution	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Multinomial distribution	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Entropy	Shannon Entropy	7.998	7.999	7.999	7.998	7.998	7.999	7.998	7.999	7.999	7.998	7.999	7.999	7.999	7.999	7.999
	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS

results of the restart test are divided into a row (H_R) and column (H_C), respectively. The minimum entropy of the entropy source is determined with $H_{min} = min(H_I, H_R, H_C)$.

Table 3 shows the results of the AIS31 test. The AIS31 test is applied with 5MB of data. The sequence generated for the entropy source is divided into $(b_1, b_2, \ldots, b_{10000})$ sub-sequences. The *Uniform* and *Multinomial* distribution sub-test determine the correlation with distributions in the sequence. In the worst case, the entropy test denotes a 7.998 of Shannon entropy, applying PVT variations.

Fig. 5 illustrates the results of the NIST 800-90B and AIS31 with PVT variations. The minimum entropy is tested in five chips. The NIST 800-90B results present a tendency to decrease when the voltage and temperature are reduced. This result is expected for the dependency of the jitter on the voltage and temperature. The entropy source presents $H_{min} = 0.9789 * 8 = 7.8312$ with -10[°C] and 1.2[V]. In another way, the Shannon entropy presents a high resistance with the PVT variations. The results of the entropy sources demonstrate the quality of the entropy source based on frequency collapse.

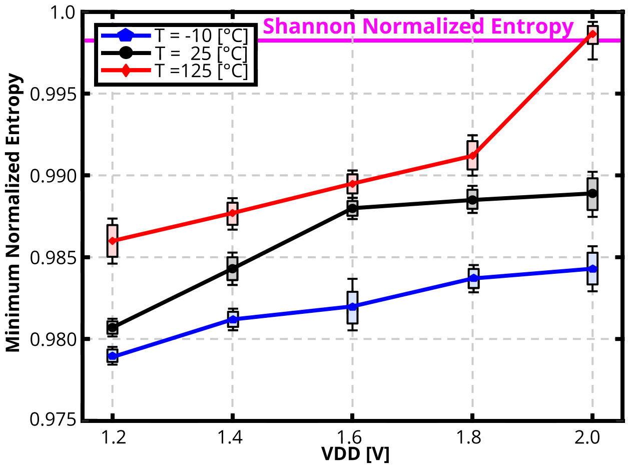


FIGURE 5. Minimum entropy with voltage and temperature variations.

IV. HEALTH TEST

The NIST 800-90B proposes a health test to detect the deviations of the sequence generated for the entropy source. The Repetition Count (RC) and Adaptive Proportion (AP) tests are explained and applied with PVT variations in this section. The health tests are applied in a System on a Chip (SoC) described in section VI.

A. REPETITION COUNT TEST

The RC test detects disastrous failures caused when the entropy source is stuck in a single value for a long time. The number of identical values (I_v) in the RC test is defined based on the H_{min} in the entropy source and the level of acceptance (α_H). Besides, the I_v is the smallest integer satisfying the inequality $\alpha_H \geq 2^{-H(I_v-1)}$. The equation for defining the number of identical values is as follows:

$$I_v = 1 + \left\lceil \frac{-log_2(\alpha_H)}{H_{min}} \right\rceil \quad (3)$$

Algorithm 1 describes the RC test implemented. Where (L) represents the length of the data-set used for the RC test. The entropy source implemented with PVT variations presents a $H_{min} = 7.8312$ of minimum entropy. The NIST 800-90B recommends a $\alpha_H = 2^{-20}$. In this way, the entropy source implemented denotes $I_v = 3$ for the RC test.

Algorithm 1 Repetition Count Test

```
Require: $I_v \in \mathbb{N}, L \in \mathbb{N}$
Ensure: $(pass) = \mathrm{RPT}(I_v, L)$
A ← getRandom()
B ← 1
for i ← 0 to L do
    X ← getRandom()
    if (X = A) then
        B ← B + 1
        if (B = I_v) then
            return FALSE
        end if
    else
        A ← X
        B ← B
    end if
end for
return TRUE
```

The entropy source presents a percentage of occurrences $I_v = 0\%$, applying the PVT variations. In this way, the entropy source passed the RC test. However, a false positive in the RC test can be generated for the length of the sequence generated. In this way, the estimator I_{v-1} is necessary to obtain an estimation when the length of the sequence tends to be infinite. Fig. 6 shows the percentage of the occurrences of I_{v-1} with PVT variations. Each scenario is measured in five different chips, recollecting an 8MB per scenario to apply to the RC test. The data illustrated in each voltage and temperature scenario is the worst case in the five chips. The percentage of occurrences in I_{v-1} is less than 0.004%. In view of the percentage in $I_v < I_{v-1}$, the entropy source implemented demonstrates a robust fixed values generation with PVT variations.

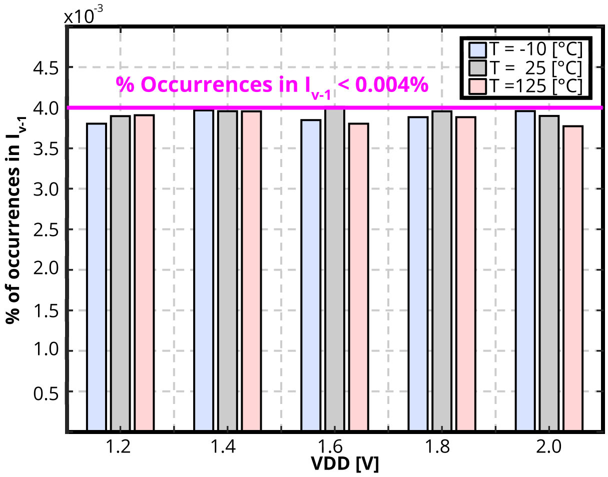


FIGURE 6. Percentage of occurrences in I_{v-1}, applying PVT variations.

B. ADAPTIVE PROPORTION TEST

The AP test exposes the missing of entropy in the time for some physical failure or external disturbance. The test measures the occurrences of a sample in the sequence generated in the entropy source. Algorithm 2 describes the AP test. The occurrences are measured in a window time (W). The size of W depends on the number of output bits in the entropy source. When the generation of the entropy source is binary, the size is 1024-bits. In the other case, when the generation is non-binary, the size is 512-bits. The size of W in the entropy source implemented is the second option. The Limit of the Maximum Cutoff Value (LMC_v) is 13 with $H_{min} = 7.8312$, $W = 512$ and $\alpha_H = 2^{-20}$.

Fig. 7 illustrates the cutoff value occurrences in the entropy source. The cutoff values are measured with PVT variations. Two thousand window samples are measured per case in five different chips. The mean of all cutoff values is 2. However, the entropy source presents a high atypical value in some cases. The maximum cutoff value (MC_V) is 10. Nevertheless, the $LMC_v > MC_v$ passed the AP test with PVT variations.

V. STATISTICAL TEST

This section shows the statistical results of the TRNG implemented. The NIST AIS20 and SP800-22 results are presented with PVT variations.

Algorithm 2 Adaptive Proportion Test

```
Require: $LMC_v \in \mathbb{N}, W \in \mathbb{N}$
Ensure: $(pass) = \mathrm{APT}(LMC_v, W)$
A ← getRandom()
B ← 1
for $i \leftarrow 0$ to $(W-1)$ do
    if $(A = getRandom())$ then
        $B \leftarrow B + 1$
    end if
    if $(B \geq LMC_v)$ then
        return FALSE
    end if
end for
return TRUE
```

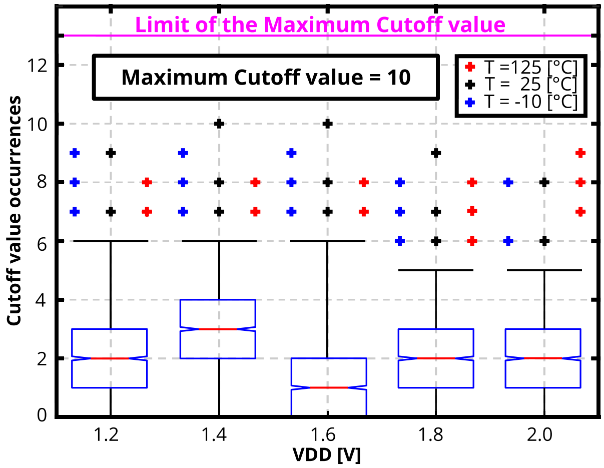


FIGURE 7. Cutoff value occurrences in AP test, applying PVT variations.

A. AIS20

Table 4 shows the results of AIS20 of the TRNG implementation. For the AIS20 the sequence is divided in $(b_1, b_2, \ldots, b_{20000})$ sub-sequences. The AIS20 test does not standardize the acceptance of the sub-test compared to the NIST SP800-22. For example, the disjointness (T_0) test determines the coincidence of non-overlapping patterns in the sequence. The sequence passes only if all sub-sequences are disunity. The monobit test evaluates one and zeros bias in the sequence generated in the TRNG. The estimator (T_1) counts the number of ones into the sequence. When the value obtained in the monobit belongs to $(9654 < T_1 < 10346)$ rank, the sequence passes the monobit sub-test. The poker test examines the pattern occurrences in the sequence. The T_2 estimator is defined as follows in (4).

$$T_2 = \frac{16}{5000} \sum_{i=0}^{5000} f[i]^2 - 5000 \tag{4}$$

where the $F[i] = \left|\{j|C_j = i\}\right|$ for $i = 1, \ldots, 5000$. Also, the vector $C = 8b_{4j-3} + 4b_{4j-2} + 2b_{4j-1} + b_{4j}$ denotes the subsequence accommodation. The poker test pass if the estimator

TABLE 4. Results of the AIS20 test.

Statistical test AIS20	PVT	CHIP # 1			CHIP #2			CHIP #3			CHIP #4			CHIP #5		
	V[V]	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0
	T[°C]	-10	25	125	-10	25	125	-10	25	125	-10	25	125	-10	25	125
Disjointness	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Monobit	T_1	9997	10055	10112	9950	10066	9983	9923	10046	10055	9841	9999	9938	10068	10035	9941
	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Poker	T_2	49.46	55.80	44.19	53.13	13.75	9.24	12.74	19.46	44.1	11.11	12.44	12.80	17.13	16.92	39.37
	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Run	1-Run	2522	2490	2521	2598	2467	2488	2441	2472	2553	2401	2479	2474	2457	2487	2514
	2-Run	1213	1268	1195	1241	1274	1276	1223	1303	1232	1264	1253	1251	1206	1274	1215
	3-Run	637	665	600	608	653	605	644	619	657	643	646	631	677	621	664
	4-Run	320	311	342	329	314	313	311	309	309	308	311	318	310	307	303
	5-Run	150	135	167	151	144	160	153	150	139	158	157	156	153	160	159
	6-Run	156	149	155	146	152	156	167	151	151	156	155	167	170	153	164
	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Long Run	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS
Autocorrelation	T_5	2474	2512	2461	2518	2508	2503	2468	2548	2571	2507	2541	2465	2424	2512	2490
	Result	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS

TABLE 5. Results of the NIST SP800-22 test.

Statistical test	PVT	CHIP # 1			CHIP #2			CHIP #3			CHIP #4			CHIP #5		
	V[V]	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0	1.2	1.8	2.0
	T[°C]	-10	25	125	-10	25	125	-10	25	125	-10	25	125	-10	25	125
Frequency	P-value	0.3190	0.7597	0.8343	0.6993	0.5955	0.3345	0.2022	0.4280	0.4288	0.8051	0.5141	0.4011	0.9642	0.4011	0.4010
	Rate	99	96	99	100	98	99	97	97	99	99	100	99	99	99	100
Block frequency	P-value	0.2368	0.4372	0.3838	0.6371	0.8513	0.5141	0.5544	0.1025	0.6155	0.1626	0.6579	0.4559	0.7571	0.9240	0.2492
	Rate	99	100	99	100	99	100	97	99	100	99	99	100	98	98	99
Cumulative sums	P-value	0.6579	0.2896	0.4559	0.0805	0.3669	0.1372	0.0909	0.3955	0.1916	0.9942	0.6993	0.8558	0.2492	0.1153	0.2133
	Rate	99	97	99	99	100	99	98	98	99	99	100	98	98	100	99
Cumulative sums	P-value	0.5194	0.6371	0.8343	0.2054	0.5749	0.4749	0.1087	0.5749	0.4190	0.2757	0.7254	0.3504	0.2492	0.1626	0.1025
	Rate	99	96	99	100	97	99	98	99	98	98	100	98	99	98	100
Runs	P-value	0.9834	0.6579	0.2622	0.8977	0.0219	0.8676	0.7791	0.1296	0.4749	0.2022	0.6786	0.2526	0.4011	0.6222	0.4372
	Rate	99	99	99	100	100	100	98	99	100	98	98	99	98	99	99
Longest run	P-value	0.5141	0.4372	0.1626	0.8676	0.9357	0.6971	0.7399	0.1372	0.6993	0.5141	0.5544	0.4011	0.2510	0.1223	0.7981
	Rate	100	99	99	100	99	99	99	100	98	100	99	99	98	98	99
Rank	P-value	0.5749	0.4591	0.7399	0.6786	0.0269	0.3898	0.5341	0.2622	0.4559	0.8180	0.8831	0.7542	0.2289	0.4372	0.4081
	Rate	99	98	100	99	100	100	97	99	100	99	98	100	100	100	100
FFT	P-value	0.7570	0.1025	0.3504	0.4749	0.3838	0.1815	0.2757	0.0288	0.1916	0.1455	0.2022	0.1757	0.1626	0.7571	0.4280
	Rate	100	99	100	98	99	99	97	99	99	98	98	98	98	99	100
Non-overlap template	P-value	0.8977	0.8977	0.9716	0.7981	0.6371	0.9914	0.3838	0.5141	0.9240	0.7981	0.4372	0.4943	0.9558	0.7197	0.7399
	Rate	99	98	100	99	99	99	99	99	99	99	100	100	98	100	100
Overlap template	P-value	0.3345	0.4372	0.3504	0.5955	0.6163	0.5544	0.2757	0.5341	0.4190	0.1916	0.7197	0.2519	0.0965	0.2757	0.2559
	Rate	100	100	99	99	99	99	100	98	99	99	99	99	100	98	99
Approx. entropy	P-value	0.9882	0.9914	0.9992	0.9812	0.9944	0.9991	0.9878	0.9911	0.9994	0.9892	0.9914	0.9985	0.9890	0.9911	0.9996
	Rate	100	99	100	100	100	100	99	100	100	99	100	99	98	99	100
Universal	P-value	0.7791	0.1372	0.7981	0.8165	0.8343	0.3190	0.4190	0.4578	0.3838	0.7597	0.4943	0.3838	0.3838	0.4190	0.6163
	Rate	99	96	100	99	98	99	99	99	99	99	100	99	98	99	100
Random excursions	P-value	0.1815	0.3504	0.2248	0.2622	0.4749	0.5544	0.5749	0.4559	0.5944	0.2519	0.7399	0.5716	0.6282	0.6371	0.5536
	Rate	99	99	99	99	98	99	99	99	99	99	98	100	99	100	99
Random excursions variant	P-value	0.9978	0.5341	0.5341	0.5341	0.2757	0.4943	0.4190	0.4190	0.3190	0.6394	0.6579	0.3144	0.9942	0.5955	0.7791
	Rate	99	100	100	100	99	99	99	98	99	100	100	99	99	99	100
Serial	P-value	0.6163	0.1153	0.4559	0.8977	0.9357	0.7981	0.7791	0.7399	0.5749	0.2022	0.8558	0.2248	0.1791	0.7597	0.7397
	Rate	100	100	99	99	100	99	99	99	100	99	100	99	100	100	99
Serial	P-value	0.2622	0.6786	0.4559	0.3190	0.7981	0.1087	0.1453	0.3669	0.1815	0.6126	0.8513	0.5341	0.9716	0.5358	0.9878
	Rate	98	100	100	99	99	100	98	98	100	99	98	99	99	100	99
Linear complexity	P-value	0.9357	0.5544	0.6371	0.5141	0.2622	0.9435	0.5733	0.3041	0.9558	0.3190	0.4943	0.6371	0.2492	0.4749	0.3292
	Rate	99	99	99	98	98	100	99	100	99	99	98	100	98	99	99

$\ast$ The P-values for the non-overlap template, random excursion and random excursion variant are the average of the P-values of all sub-test.

belongs to ($1.03 < T_2 < 57.4$) rank. The general run tests ($T_{3,4}$) expect a number $r(n, i) = (n - i + 3)/2^{i+2}$ in the different runs of length i in an independent unbiased binary sequence of n bits. The run estimator T_3 is obtained in 5.

$$T_3 = \sum_{i=0}^{k} \frac{(\hat{r}(n, i, 0) - r(n, i))^2}{r(n, i)} + \sum_{i=0}^{k} \frac{(\hat{r}(n, i, 1) - r(n, i))^2}{r(n, i)} \tag{5}$$

The $\hat{r}$ denoted the observed numbers in the λ-runs. The run test passes when the result of the estimator is into the rank per each λ-runs. The rank in each λ-runs is ($\sum_{\lambda}^{20000} T_3(\lambda, 0)$, $\sum_{\lambda}^{20000} T_3(\lambda, 1)$). In addition, the long run test is a run test with ($\lambda > 34$). The autocorrelation test checks the similarity of the sub-sequence in the time. However, the estimator T_5 is obtained with 10000 sub-sequence, depending on the values of the parameter $\tau \in (1, 2, \ldots, 5000)$. The autocorrelation test follows an N(0,1) distribution. In this way, the test needs to be two-sided in the length of the sequence. The TRNG passed all AIS20 tests with PVT variations.

$$T_5(\tau) = \sum_{j=1}^{5000} (b_j \oplus b_{j+\tau}) \tag{6}$$

B. NIST SP800-22

Table 5 shows the results of NIST SP800-22, applying the PVT variations. The level of significance (α) applied in the test is $\alpha = 0.01$. In this way, one sequence of 100 sequences is to be rejected. In addition, the test needs a P-value ≥ 0.01, indicating the sequence generated is 99% confidence. The TRNG is proved in the different chips with low, typical, and high voltage and temperature values. The TRNG passes all tests NIST SP800-22.

VI. RESULTS

This section presents the TRNG implemented in 0.18 μm GP CMOS technology. First, a comparison of the minimum entropy is present with entropy sources based on different physical phenomenons. Second, the results of the TRNG of power, area, and the bit rate.

Fig. 8 shows a block diagram of the microcontroller used for measuring the TRNG implemented. The SoC is based on a RISC-V ISA processor [40] with *AHB-Lite* for the system bus and *APB* for the peripheral bus [41]. In addition, the system has a 1-KB of RAM, a Timer, a JTAG-based debug module, a UART, a QSPI for external flash memory, a General-Purpose In-Outs (GPIO), and the TRNG.

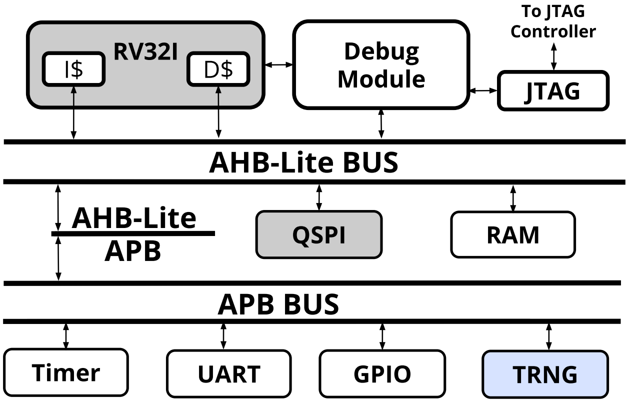


FIGURE 8. Block diagram of the SoC implemented.

Table 6 depicts a comparison of the minimum normalized entropy with entropy sources based on different phenomena. The minimum entropy is obtained with the NIST SP800-90B test with PVT variations in the worst result. The TRNG implemented presents a competitive minimum entropy compared to entropy sources based on leakage and resistance variations. However, the entropy sources with different physical phenomena show better results with an IID assumption, generating the possibility of overestimation in the minimum entropy. In addition, the implementation passes the health test provided by NIST SP800-90B, demonstrating the health of the TRNG.

Fig. 9 shows the micrograph of the TRNG in 0.18 μm GP CMOS technology. The micrograph contains an SoC with the TRNG. The TRNG occupies 25600μm^2 with four entropy sources. Each entropy source with the *RO REF* represents 12.5% of the area. The compare and capture stages represent 50% of the resources. The suggestions implemented in the entropy source present a 1.5× of area overhead.

TABLE 6. Summary and comparison of minimum entropy.

	Type of Data	Min. Entropy	Physical Phenomena	PVT Variations	Health Test
This Work	Non-IID	0.978	Collapse Frequency in RO	Yes	Yes
[7]	IID	0.991	Metastability in Latches	Yes	Yes
[14]	IID	0.995	Jitter in Analog-RO	Yes	No
[16]	IID	0.996	Charge-Trapping in FinFET	Yes	No
[17]	IID	0.993	Leakage in SRAM	Yes	No
[18]	IID	0.996	Resistance variations in RRAM	No	No
[21]	IID	0.997	Resistance variations in RRAM	Yes	No
[23]	Non-IID	0.407	Metastability in Latches	Yes	No
[25]	IID	0.992	Multiple Entropy Sources	Yes	No
[27]	Non-IID	0.931	Jitter Quantization	Yes	No
[31]	IID	0.883	Self-time in Chaotic system	No	No

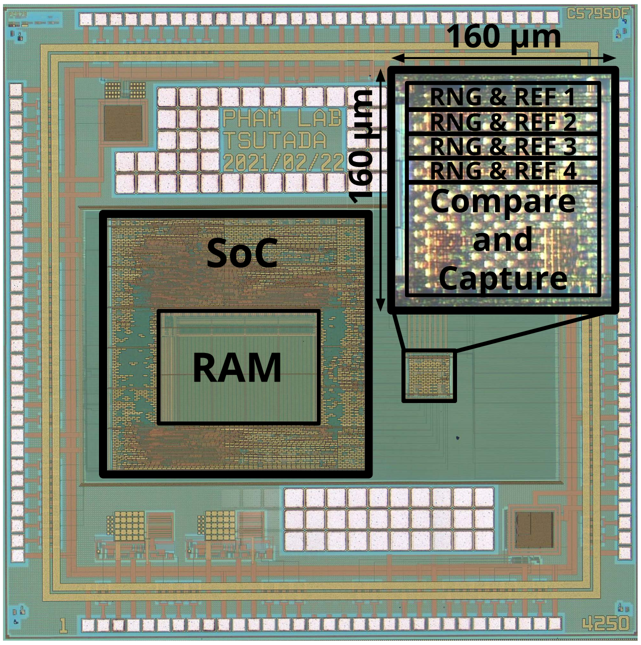


FIGURE 9. Micrograph of the TRNG in 0.18μm GP CMOS technology.

Fig. 10 illustrates the power consumption of the TRNG implementation in nominal conditions, using the different entropy sources. The four entropy sources area implemented with 2, 4, 8, and 16 delay stages. The power consumption of the TRNG is determined in major with the number of the delay stages into the multi-modal RO. The TRNG presents a 564.21-μW until 1.88-mW with 16 and 2 delay stages in the entropy sources, respectively.

TABLE 7. Summary and comparison results.

	Technology	Power	Bit rate	Energy	Area	Normalized Area $^{+}$	NIST	Entropy
	[nm]	[mW]	[Mb/s]	[pJ/bit]	[μm^2]	[10^3 F^2]	AIS	Source
This work	180	0.56† - 1.88‡	7.3† - 9.2‡	77.2† - 204.3‡	16000	493.82	YES	Multi-modal
					(25600*)		YES	RO
[14]	65	0.36	52	6.9	60000	14201.18	YES	Stochastic Delta
							NO	Sigma Modulator
[17]	28	34.56	3.6	9.6 - 17.2	15400	19642.85	YES	SRAM
							NO	memory
[25]	14	1.5	162.5	9.23	1008	5142.85	YES	Multiple Entropy
							NO	Sources
[31]	40	0.528	1600	0.33	270	168.75	YES	Chaotic-Cellular
							YES	Automata
[33]	65	0.159	2.8	56.79	960	227.21	YES	
							NO	Multi-modal
	28	0.54	23.16	23.32	375	478.31	YES	RO
							NO	
[36]	180	0.34 - 0.42	1.6 - 3.7	92 - 264	8700	268.51	YES	Multi-modal
							NO	RO

† Entropy source with 16 stages. ‡ Entropy source with 2 stages. * Including 3 additional entropy sources.
$^{+}$ Normalized Area = (area)/(minimum feature size of the process)^2

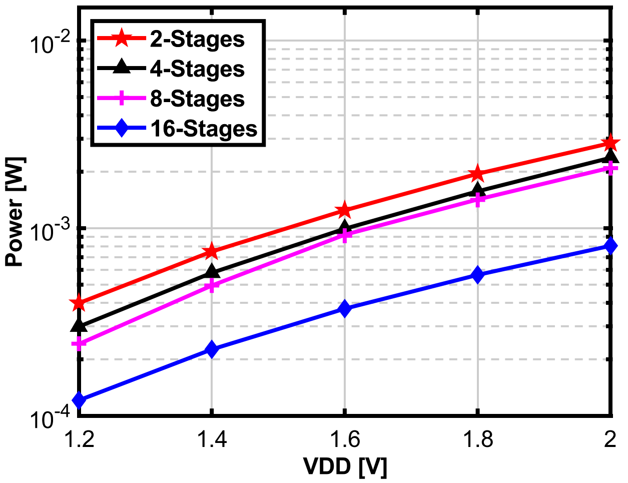


FIGURE 10. Power consumption of the TRNG implemented using the different entropy sources.

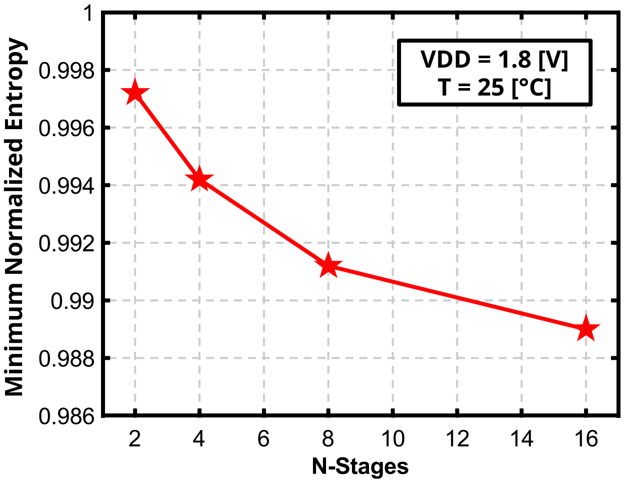


FIGURE 11. Minimum entropy in the entropy sources implemented per number of delay stages.

Table 7 shows a comparison of the TRNG implementation with the related works. The statistical and entropy tests are tested using a 16 delay cell in each edge. The bit rate of the TRNG depends on the random time per collapse. In this way, the time of the collapse frequency is measured 10Mtimes, reporting a 7.3 and 9.2-Mb/s of bit rate, using 16 and 2 delay stages in the entropy source. In addition, the implementation reports a 493.82-10^{3}F^2 of normalized area.

Fig. 11 plots the minimum entropy of the TRNG using the different entropy sources. The TRNG denotes a 0.9890 until 0.9972 of minimum entropy in the 16 and 2 delay stages in multi-modal RO. The results show an increase in the minimum entropy when the number of stages is reduced. However, the uses of a reduced number of delay stages increase the power consumption and the energy per bit of the TRNG implemented.

VII. CONCLUSION

This work presents a robust and healthy TRNG in ASIC based on collapse frequency. The TRNG is implemented using a suggestion based on an analytical model, reducing the undesirable effects in the random number generation. The implementation passed the NIST SP800-90B and AIS31 entropy tests. The NIST SP800-90B is applied using a non-IID assumption based on the ASIC analytical model. The entropy sources implemented present a 0.9789 minimum normalized entropy and 7.998 of Shannon entropy, using 16 delay stages in the entropy source. Besides, the TRNG implementation passed the NIST SP800-22 and AIS20 statistical test. In addition, the implementation passed the two health tests provided for the NIST SP800-90B. The RC test presents 0% and 0.004% of occurrences in the I_v and I_{v-1} estimator, respectively. The AP test reports a $MC_v = 10$ with

$LMC_v = 13$. The TRNG passed the entropy, health, and statistical tests with PVT variations, demonstrating the robustness and health of the implementation. The data used for the tests are recollected using a RISC-V microcontroller. The TRNG occupies $25600\mu m^2$ using four entropy sources. Each entropy source represents 12.5% of the area in the implementation. The suggestions applied in the entropy source present 1.5× of area overhead. Finally, the TRNG denotes a 0.56 until 1.88-mW of power consumption, 7.3 until 9.2-Mb/s of bit rate, and 77.2 until 204.3 pJ/bit of energy per bit using 16 and 2 delay stages in the entropy source, respectively.

ACKNOWLEDGMENT

The VLSI chip in this study has been fabricated in the chip fabrication program of through the activities of VDEC, the University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation.

REFERENCES

[1] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, and J. Dray, "NIST special publication 800-22: A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-22 Rev 1a, Apr. 2010.

[2] W. Schindler, "AIS 20–functionality classes and evaluation methodology for deterministic random number generators," Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany, Tech. Rep. AIS20V1, Dec. 1999.

[3] E. Barker, A. Roginsky, and R. Davis, "Recommendation for the entropy sources used for random bit generation," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-90B, Jan. 2018.

[4] W. Killmann and W. Schindler, "AIS 31: Functionality classes and evaluation methodology for true (physical) random number generators, version 3.1," Bundesamt für Sicherheit in der Informationstechnik, Bonn,Germany, Tech. Rep. AIS31An, Sep. 2001.

[5] G. Cox, C. Dike, and D. J. Johnston, "Intel's digital random number generator (DRNG)," in *Proc. IEEE Hot Chips Symp. (HCS)*, Stanford, CA, USA, Aug. 2011, pp. 1–13.

[6] S. U. Hussain, M. Majzoobi, and F. Koushanfar, "A built-in-self-test scheme for online evaluation of physical unclonable functions and true random number generators," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 1, pp. 2–16, Jan. 2016.

[7] N. Torii, D. Yamamoto, and T. Matsumoto, "Evaluation of latch-based physical random number generator implementation on 40 nm ASICs," in *Proc. 6th Int. Workshop Trustworthy Embedded Devices*, New York, NY, USA, Oct. 2016, pp. 23–30.

[8] N. Torii, R. Minagawa, H. K. Omae, and K. Hayashi, "Implementation and evaluation of ring oscillator-based true random number generator," in *Proc. 9th Int. Symp. Comput. Netw. (CANDAR)*, Matsue, Japan, Nov. 2021, pp. 189–195.

[9] 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)*, Jeju, South Korea, Jun. 2019, pp. 1–3.

[10] B. Jun and P. Kocker, "The Intel random number generator," Cryptogr. Res., Apr. 1999. [Online]. Available: https://www.rambus.com/intel-random-number-generator/

[11] F. Tehranipoor, P. Wortman, N. Karimian, W. Yan, and J. A. Chandy, "DVFT: A lightweight solution for power-supply noise-based TRNG using dynamic voltage feedback tuning system," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 6, pp. 1084–1097, Jun. 2018.

[12] A. Degada and H. Thapliyal, "An integrated TRNG-PUF architecture based on photovoltaic solar cells," *IEEE Consum. Electron. Mag.*, vol. 10, no. 4, pp. 99–105, Jul. 2021.

[13] M. Danesh, A. B. Venkatasubramaniyan, G. Kapoor, N. Ramesh, S. Sadasivuni, S. T. Chandrasekaran, and A. Sanyal, "Unified analog PUF and TRNG based on current-steering DAC and VCO," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 11, pp. 2280–2289, Nov. 2020.

[14] S. T. Chandrasekaran, V. E. G. Karnam, and A. Sanyal, "0.36-mW, 52-Mbps true random number generator based on a stochastic delta–sigma modulator," *IEEE Solid-State Circuits Lett.*, vol. 3, pp. 190–193, 2020.

[15] S. Dong, Y. Wang, X. Xin, C. Liu, and X. Tong, "An op-amp sharing folded Bernoulli mapping TRNG for low-power applications," in *Proc. 4th Int. Conf. Circuits, Syst. Simulation (ICCSS)*, May 2021, pp. 113–117.

[16] J. Yang, Q. Ding, T. Gong, Q. Luo, X. Xue, Z. Gao, H. Yu, J. Yu, X. Xu, P. Yuan, X. Li, L. Tai, S. S. Chung, H. Lv, and M. Liu, "Robust true random number generator using stochastic short-term recovery of charge trapping FinFET for advanced hardware security," in *Proc. IEEE Symp. VLSI Technol.*, Honolulu, HI, USA, Jun. 2020, pp. 1–2.

[17] S. Taneja, V. K. Rajanna, and M. Alioto, "36.1 unified in-memory dynamic TRNG and multi-bit static PUF entropy generation for ubiquitous hardware security," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, Feb. 2021, pp. 498–500.

[18] 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. Nanoscale Era (DTIS)*, Marrakech, Morocco, Apr. 2020, pp. 1–2.

[19] B. M. S. B. Talukder, J. Kerns, B. Ray, T. Morris, and M. T. Rahman, "Exploiting DRAM latency variations for generating true random numbers," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2019, pp. 1–6.

[20] M. I. Rashid, F. Ferdaus, B. M. S. B. Talukder, P. Henny, A. N. Beal, and M. T. Rahman, "True random number generation using latency variations of FRAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 1, pp. 14–23, Jan. 2021.

[21] B. Gao, B. Lin, X. Li, J. Tang, H. Qian, and H. Wu, "A unified PUF and TRNG design based on 40-nm RRAM with high entropy and robustness for IoT security," *IEEE Trans. Electron Devices*, vol. 69, no. 2, pp. 536–542, Feb. 2022.

[22] S. Taneja, V. K. Rajanna, and M. Alioto, "In-memory unified TRNG and multi-bit PUF for ubiquitous hardware security," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 153–166, Jan. 2022.

[23] S. Tao and E. Dubrova, "TVL-TRNG: Sub-microwatt true random number generator exploiting metastability in ternary valued latches," in *Proc. IEEE 47th Int. Symp. Multiple-Valued Log. (ISMVL)*, Novi Sad, Serbia, May 2017, pp. 130–135.

[24] P. Z. Wieczorek and K. Gołofit, "True random number generator based on flip-flop resolve time instability boosted by random chaotic source," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1279–1292, Apr. 2018.

[25] 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.

[26] R. Della Sala, D. Bellizia, and G. Scotti, "A novel ultra-compact FPGA-compatible TRNG architecture exploiting latched ring oscillators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1672–1676, Mar. 2022.

[27] 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.

[28] T. Chen, Y. Ma, J. Lin, Y. Cao, N. Lv, and J. Jing, "A lightweight full entropy TRNG with on-chip entropy assurance," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 12, pp. 2431–2444, Dec. 2021.

[29] N. N. Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "Design, implementation and analysis of efficient hardware-based security primitives," in *Proc. IFIP/IEEE 28th Int. Conf. Very Large Scale Integr. (VLSI-SOC)*, Salt Lake City, UT, USA, Oct. 2020, pp. 198–199.

[30] J. Cui, M. Yi, D. Cao, L. Yao, X. Wang, H. Liang, Z. Huang, H. Qi, T. Ni, and Y. Lu, "Design of true random number generator based on multi-stage feedback ring oscillator," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1752–1756, Mar. 2022.

[31] Y. Luo, W. Wang, S. Best, Y. Wang, and X. Xu, "A high-performance and secure TRNG based on chaotic cellular automata topology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4970–4983, Dec. 2020.

[32] R. Serrano, C. Duran, T.-T. Hoang, M. Sarmiento, K.-D. Nguyen, A. Tsukamoto, K. Suzaki, and C.-K. Pham, "A fully digital true random number generator with entropy source based in frequency collapse," *IEEE Access*, vol. 9, pp. 105748–105755, 2021.

[33] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester, "A 23 Mb/s 23pJ/b fully synthesized true-random-number generator in 28 nm and 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2014, pp. 280–281.

[34] J. Cartagena, H. Gomez, and E. Roa, "A fully-synthesized TRNG with lightweight cellular-automata based post-processing stage in 130 nm CMOS," in *Proc. IEEE Nordic Circuits Syst. Conf. (NORCAS)*, Copenhagen, Denmark, Nov. 2016, pp. 1–5.

[35] 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.

[36] H. Gomez, J. Arenas, and E. Roa, "Low-cost TRNG IPs," *IET Circuits, Devices Syst.*, vol. 14, no. 7, pp. 942–946, Oct. 2020.

[37] K. Yang, J. H. Baek, Y.-D. Kim, J. Lee, D. K. Kim, and B.-D. Choi, "A physically unclonable function with BER $< 10^{-8}$ for robust chip authentication using oscillator collapse in 40 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2015, pp. 254–255.

[38] 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)*, Rotorua, New Zealand, Aug. 2019, pp. 152–159.

[39] J.-S. Kang, H. Park, and Y. Yeom, "On the additional chi-square tests for the IID assumption of NIST SP 800-90B," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Calgary, AB, Canada, Aug. 2017, pp. 375–3757.

[40] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovi, "The RISC-V instruction set manual, volume I: User-level ISA, version 2.0," EECS Dept., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-54, May 2014. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-54.html

[41] ARM, "AMBA AXI and ACE protocol specification," ARM, Cambridge, U.K., Tech. Rep. ARM IHI 0022D, 2013. [Online]. Available: https://developer.arm.com/architectures/system-architectures/amba/specifications

RONALDO SERRANO (Graduate 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 at The University of Electro-Communications (UEC), Tokyo, Japan. His research interests include computer architecture, high-speed digital interfaces, and hardware for security.

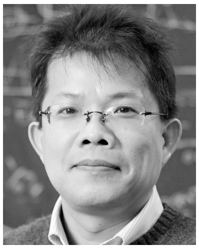
CKRISTIAN DURAN (Graduate Student Member, IEEE) received the B.Sc. degree in electronics and the M.S. degree in telecommunications from the Universidad Industrial de Santander (UIS), Bucaramanga, Colombia, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electronics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

MARCO SARMIENTO (Graduate 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 at The University of Electro-Communications (UEC), Tokyo, Japan. His research interests include debugging and security for integrated systems.

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 University of Science, Vietnam National University, Ho Chi Minh City, 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 also a Research Assistant at the National Institute of Advanced Industrial Science and Technology (AIST), Tokyo.

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

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 interests include the design of analog and digital systems using integrated circuits.