Realization of Re-configurable True Random Number Generator on FPGA

M. Priyatharishini(✉) and M. Nirmala Devi

Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
{m_priyatharishini,m_nirmala}@cb.amrita.edu

Abstract. True random number generation (TRNG) is one of the prominent research areas in present scenario of cryptography and security. It has been reported in the recent past that even TRNG encounters security threats. In order to ensure the security of the random numbers, entropy of random numbers being generated should be high. There are different approaches to generate the random numbers from the physical processes, ranging from jitter to chaos. Various schemes employing the jitter as entropy source have been reported. The usage of jitter in ring oscillator aids in obtaining a high speed real-time random number generation (RNG). On the other hand, the asynchronous architecture ensures high security, which has been implemented in the work. Re-configuring these two architectures develops a RNG with high-speed and security. The statistical tests along with internal tests are conducted to ensure security in the architecture. National Institute of Standards and Technology (NIST) tests validated the unpredictability and randomness of the true random number (TRN) generated.

Keywords: Hardware Trojan · TRNG · Security · Jitter · NIST test · Entropy source

1 Introduction

With the advent of Internet of Things (IoT), security has become a major concern for every physical entity. There are a number of attacks on hardware which becomes a threat to the usage of chips in secured applications. Different approaches to detect and diagnose hardware Trojan is another field of research [1, 2]. It is highly alarming that even the hardware modules like TRNG which are designed to ensure security in financial applications are subjected to malicious modifications [7]. True random numbers (TRNs) are pure random numbers, which does not show any pseudo random property at any long run [5]. These random numbers are generated from physical variations like thermal noise, chaos, jitter or meta-stability [6] as shown in Fig. 1.

In the classification, noise and chaos are implemented in analog component based phenomena. Analog circuits are more prone to malicious attacks than digital circuits [3, 7]. Thus, noise and chaos architectures are not considered in this work. The meta-stability is the uncertain state between zero and one in a circuit. This uncertain state can be sampled for generating the random bits. The Jitter based concept is the most

S. M. Thampi et al. (Eds.): SSCC 2019, CCIS 1208, pp. 247–256, 2020.
https://doi.org/10.1007/978-981-15-4825-3_20

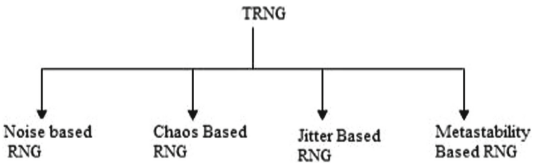

Fig. 1. Classification of TRNG

preferred architecture to generate the bit streams with true randomness because of the easiness to capture the jitter. Two different architectures confined in this work using jitter concept are Free Running Oscillator (FRO) based and Self Timed Rings (STRs) based. The inverter ring oscillator (IRO) based provides a simple implementation [4], while STR provides highly secured random bits [8]. Till now no attacks has been reported to the STR based TRNG, which uses jitter as entropy source. These two architectures have commonality while implementation, in terms of type of noise source used and method of digitization. Reconfiguring these two provides an advantage of providing security with less complex circuit realization.

This paper shows how re-configuring the architecture provides an advantage in- terms of randomness and hence security with effective resource utilization. Section 2 briefs the research methods that have been adopted in conventional TRNG architecture. Section 3 presents the proposed methodology. Simulation results and analysis of the implemented design is presented in Sect. 4. Section 5 concludes the scheme with suggestions for future scope.

2 Related Works

Conventionally, the concept of true random number generation had been attempted using Phase Locked Loops (PLLs) [5]. The analog PLL noise is the source of randomness in the circuit. The jitter is identified by using a correlated signal (clock) generated by PLL to sample the reference signal (clock). The ideal performance is limited between hundreds to several thousand bits and the capability of FPGAs.

The Free Running oscillators (FROs) Based TRNG design [9] is the modified architecture of that shown in [4]. This is designed such that, the post processing stage is not needed for raw bits to ensure the correctness of bits. Every ring is provided with an extra DFF to improve the performance. This is tested by DIEHARD and NIST tests. It provides a fast TRNG with less number of rings. The Chaos based architecture [10] uses well defined switching capacitor. Optimization is done to reduce the influence of supply voltage to provide enough randomness. The sequential circuits consist of memory elements, which may go to unstable state if not properly synchronized. This unstable condition is used to generate the true random bits in several systems [11]. Due to delay variation in clock and data path, setup and hold violations can occur. Sampling is done during this time give rise to random bit sequences. Another method of generation is by using Thermal Noises and are generated using ring oscillators to maximize the throughput and maintain the quality of random bits [12]. In paper [15], various trojan models are explained, in which triggering an analog trojan varies the temperature during the silicon

nitrite layering process and it affects the IC life time. Analog processing increases the vulnerability to attacks and limits the performance. By replacing inverter oscillator rings with self-timing rings, a more secure random number generator is developed [10]. The properties of various types of TRNGs fir the two architectures are shown in Table 1.

Table 1. Various types of TRNG

TRNG	Type	Noise source	Implementation details	Limitations/advantages
PLL TRNG [4]	Based	Jitter	Altera FPGA	Restricted only to FPGA with analogue components
FRO TRNG [10]	Based	Jitter	Altera Cyclone II FPGA	Simple design to implement
Chaos TRNG [11]	Based	Chaos	Mixed-Signal PSoC	Does not provide high randomness
Metastability TRNG [12]	Based	Metastability	Xilinx Virtex 5	Delay variations in the system is checked to generate random bits
Noise TRNG [13]	Based	Thermal Noise	CMOS process of TSMC	Noisy Analogue Behaviors limits performance
Self-timed rings TRNG [9]	Timed based	Jitter	Xilinx Virtex 6	More Secure TRNG

In [4], basic inverter ring oscillators are developed to generate the true randomness in bits. It involves random switching at the XOR tree before registering the raw data. It is modified [9] such that combinational gate switching is considerably reduced. The PLL [5] is used to generate the random bits streams but it is dependent on the FPGA vendors. Analog component based generators also provide true randomness [10–12], but these circuits have high sensitivity to attacks [3, 7]. The meta-stability of bi-stable circuits are the commonly existing phenomenon, that can be used for generating the bits [11], by sampling the uncertainty caused due to the violation of setup and hold window.

Device independence with improved security and unpredictability are the most important traits of a good random number generator. From the comparisons made, Phase Locked Loops (PLL) based is more devices dependent and the aim is to generate a random number which shows true randomness and need to be implemented on FPGA. The Free Running Oscillators oscillates due to the delay variations in the gates. Those can be sampled such that the frequency deviation is almost same. Since the most secured one among these is STR based, it is used in the proposed implementation along with FRO. The challenging task is to generate high speed architecture with more se- cured random bits in a single chip. This can be accomplished by using reconfigurable architecture of both inverter ring oscillator and self-timed rings.

3 Methodology

The method used for the development of the architecture is shown in Fig. 2. Each TRNG consists of noise sources. The noise source generates the true random bits from number of oscillations in each architecture. The noise source can be IRO or STR. The ring oscillators are produced by connecting the odd number of inverters [13]. The feedback loop causes the inverter to oscillate and hence produces the unpredictable random numbers. The delay of all the components causes the period as 2X, that is X is the delay of all the components. The phenomenon of any electronic circuit involving a switching digital signal is represented as Jitter. The ring oscillator uses clock jitter to sample the data signal. The several equal length ring oscillators produce the jitter signals, which are sampled using DFF and then combined together using a XOR tree. Self-timed rings are basically the asynchronous ripple FIFO (First in First Out) memories, connected in the form of a ring [14]. The data transfer is accomplished using asynchronous handshake protocol. The protocol assures the even distribution of events through the different stages in the ring. The operation is such that upon request the data is sent with an acknowledgment. There is a forward input F and a reverse input R to a stage. If both the forward and reverse inputs are same, the output takes same value of forward input F. Else the previous value is maintained.

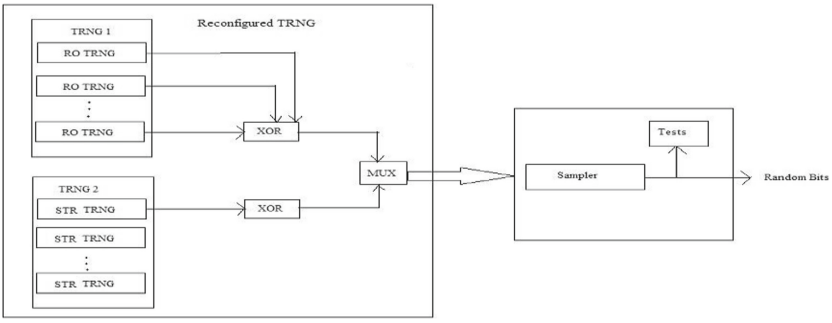

Fig. 2. Block diagram of the reconfigured architecture

The, raw random numbers obtained from noise sources are thus evaluated. The mode selection is done at this phase based on the requirement of the bits. The coherent sampling (CS) is the sampling procedure done for both modes, where CS is a technique, which allows a fixed number of samples to confine to the sampling interval. The sampling interval can be predefined, which makes it advantages without any loss in bits. Mathematically [9], it can be represented as

$$\frac{\mathrm{fin}}{\mathrm{fs}} = \frac{\mathrm{Nc}}{\mathrm{Ns}} \tag{1}$$

where fin is the frequency of sampled signal, fs is the frequency of sampling signal, Nc is the number of sampled signal cycles and Ns is the samples strength. The design should ensure that Nc and Ns are high and should be co-prime to obtain a high resolution of

sampled signal. The random data is selected as per need and statistical tests are conducted for those bit sequences. The procedure followed is as shown in pseudo code.

PSEUDO CODE
Step 1: Generation of raw random bits by two architectures.
Step 2: Calculation for entropy of the raw bits.
Step 3: Digitizing the generated raw noise data.
Step 4: Selecting the mode of operation of TRNG.
Step 5: Securing true random bits.
Step 6: Applying Statistical Tests.

The reconfigurable architecture is as shown in Fig. 3. These two architectures provide two important aspects of the true random number generators; speed and security. The IRO oscillator involving number of rings connected together with the same ring structure is used to sample the data bits.

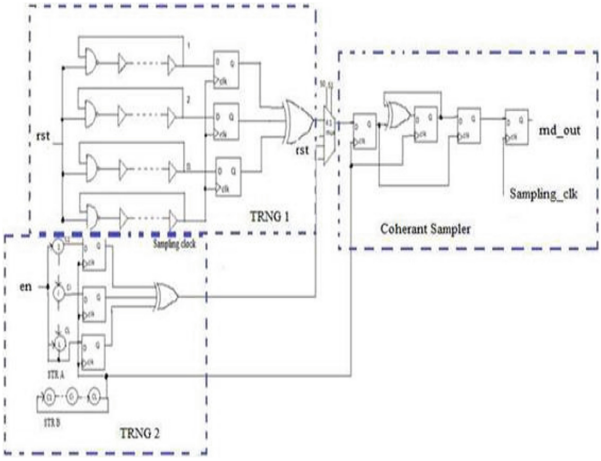

Fig. 3. Proposed reconfigurable architecture

If the inverter rings are replaced by self-timed rings (STR) are included, then the structure becomes more complex but provides an added advantage of security. These two architectures used on a single system helps the bits to be more secured with shared resource utilization. The several statistical test are performed to ensure the quality of each random numbers. The general statistical test suites employed to test the random sequence are from NIST (National Institute of standards and technology). The security level is evaluated and enhanced statistical analysis is done using these tests.

4 Simulation Results and Analysis

Reconfigured architecture is implemented and the results are validated using the standard random number tests. After the accumulation of the jitter, the jitter is sampled. The

standard deviation of the bits obtained is shown in Fig. 4 for frequency measured (MHz) and in Fig. 5 for period measured (ns). The average count is shown in X-axis with respect to STR and RO based architectures. The jitter variation in RO is more in between the limits as shown in Fig. 5. The standard deviation of the bits varies more from zero indicating the randomness property of the sequences. The ring oscillator and self-timed rings are used as noise source when considering the jitter based sampling. Each Ring oscillators is connected to a DFF to form single TRNG unit.

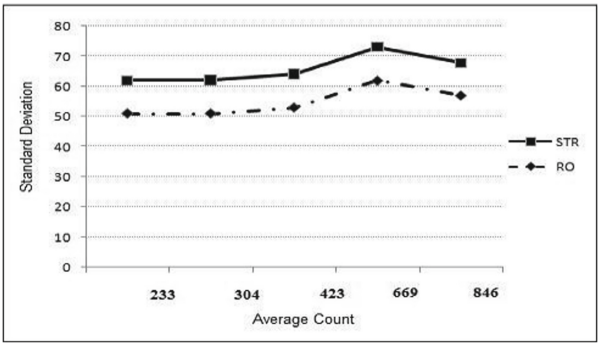

Fig. 4. Standard deviation in frequency

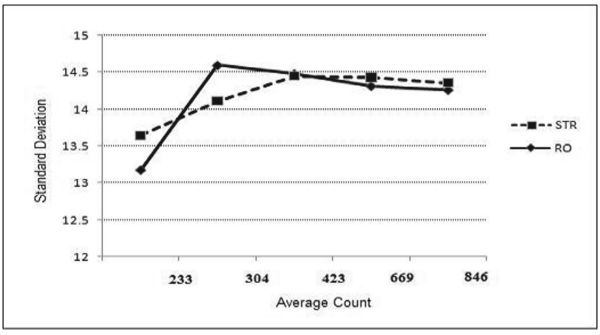

Fig. 5. Standard deviation in period

The standard deviation in frequency in each count is uniform and alike. At each count, the deviation is high for STR based TRNG compared to RO Based architecture as shown in Fig. 5. The average counting is done till 846 to determine the frequency deviation. Standard deviation is high for both when the count reached 699 indicating more variation in frequency from zero indicates true randomness.

Figure 5 shows the standard deviation in period for both the architectures. When the count is 304, the ring oscillator based deviates more than self-timed rings. This average count gives a high jitter accumulation since the deviation is high.

The number of TRNG units are connected together using XOR tree to increase the randomness. By replacing ROs in the above with self-timed rings (STRs), FIFO architecture is obtained. The power calculations for both structures implemented using Synopsys Design compiler are as shown Table 2. The power calculations are made in Watts (W). The ring oscillator TRNG consumes more than half internal power compared to STR TRNG. The STR architecture power consumption is more pronounced in terms of leakage and internal power. The area calculations of corresponding individual implementation are as shown in Table 3.

Table 2. Power calculations

Power (W)	RO TRNG	STR TRNG
Cell internal power	18.325	37.3103
Net switching power	186.27	0.1619835
Total dynamic power	204.595	37.4723
Cell leakage power	1.471	6.8288

Table 3. Area calculations

Area (nm square)	RO TRNG	STR TRNG
Combinational area	18.465	479.231
Non combinational area	593.043	1032.19
Net interconnect area	34.619	43.0398
Total cell area	611.508	1511.424
Total area	646.127	1554.46

The evaluation of the random bit's sequence is done using the NIST tests suite and the results are shown in Table 4. The p-value is the probability value which sets a standard limit for determining the quality of random bits. The p-value range should be more than 0.01 (>0.01) to say the numbers as random. The highlighted value is the values that are complimentary to the values of the corresponding tests of whole block. Since the architecture has increased its complexity, more resources are being used.

The resource utilization of different TRNG along with the proposed method is shown in Table 5, along with the entropy value per bit. The area is measured based on LUT count for the realization. The power and area are obtained after implementing in Xilinx ISE design suite. The aggregate of all the hardware modules utilized as per the exposed results in [16] are 71.25% of area utilization.

The implementation in FPGA indicates the proof of the concept being stated. The hardware implementation of the TRNG is done in SPARTAN-6 XC6SLX45-2-CSG484 An-vyl boards as shown in Fig. 6. The visualization of the output is done Mixed Signal Oscilloscope (MSO) of 100 MHz 4GSa/s. The Agilent 54620-61601. Logic analyzer

Table 4. P-values of proposed and simple ring oscillator TRNG architecture

Tests	Simple RO TRNG	Multiple RO TRNG	Multiple STR TRNG
Frequency test	0.1718	0.2495	0.4251
Block frequency	0.3504	0.3846	0.3258
Runs	0.1521	0.1864	0.6975
Longest run of ones	0.7048	0.8476	0.1758
Non-overlapping matching	0.2475	0.1446	0.3214
Overlapping template matching	0.4792	0.5224	0.1514
Cumulative Sums	0.2314	0.3148	0.2434

Table 5. Resource utilization's

TRNG type	AREA(LUT)	Power (mW)	Entropy
Simple RO TRNG	67	2.16	0.98
Multiple RO TRNG	523	54.72	0.999
STR TRNG	346	68.9	0.998
Reconfigurable TRNG	602	115.7	0.999

probe cable is used as the interface for connecting the oscilloscope with FPGA board. The coding is done in the Xilinx ISE design suite 14.7.

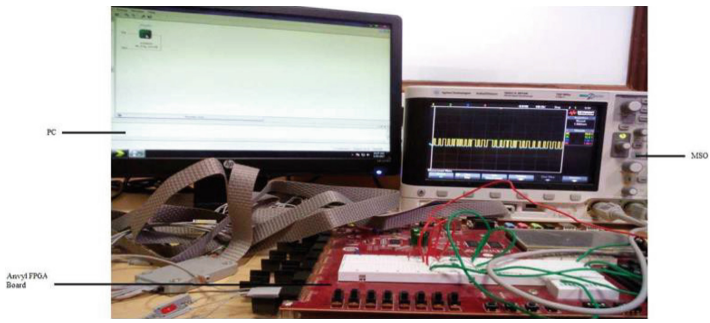

Fig. 6. Hardware implementation of the TRNG

The re-configurable random number scheme is essential in today's embedded system. The ring oscillator based TRNG are less complex compared to STR based TRNG with coherent sampling scheme. Both are combined together making the system more reliable by including the advantages of both the architectures. The power value obtained shows

the usage, which is less than when both the architectures are combined without providing any reconfiguration.

5 Conclusion and Discussion

In this work, a modified TRNG architecture is proposed by re-configuring the architectures such that a RNG can be used for highly random and secured as well as high speed architecture. The security of the random numbers is ensured by validating the true random properties of the bits being generated. The inverter ring oscillators generate the bits so fast indicating the decrease in delay of the inverters in the structure. Even though the STR is more complex than IRO, it provides more secured bits for long run. The proof of the architecture being implemented is done in Xilinx FPGA upon validating the results using NIST tests.

In future, the chip can include the online temperature tracking system to evaluate the robustness conditions and ensure the protection against the hardware Trojan attacks. Metastable architecture can be incorporated with this by evaluating the delay variations of the system. Whenever system is encountering any delay variations and fluctuations in signals such that it may violate the setup and hold time, then automatically the system can be made to operate in metastable mode and hence generate random bits.

References

1. Sree Ranjani, R., Nirmala Devi, M.: Golden-chip free power metric based Hardware Trojan detection and diagnosis. Far East J. Electron. Commun. **17**(3), 517–530 (2017)
2. Karunakaran, D.K., Mohankumar, N.: Malicious combinational Hardware Trojan detection by gate level characterization in 90nm technology. In: Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Hefei, pp. 1–7 (2014)
3. Bayon, P., et al.: Contactless electromagnetic active attack on ring oscillator based true random number generator. In: Schindler, W., Huss, S.A. (eds.) COSADE 2012. LNCS, vol. 7275, pp. 151–166. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29912-4_12
4. Sunar, B., Martin, W.J., Stinson, D.R.: A provably secure true random number generator with built-in tolerance to active attacks. IEEE Trans. Comput. **56**(1), 109–119 (2007)
5. Fischer, V., Drutarovský, M.: True random number generator embedded in reconfigurable hardware. In: Kaliski, B.S., Koç, ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 415–430. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_30
6. Stipčević, M., Koç, Ç.K.: True random number generators. In: Koç, Ç.K. (ed.) Open Problems in Mathematics and Computational Science, pp. 275–315. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10683-0_12
7. Markettos, A.T., Moore, S.W.: The frequency injection attack on ring-oscillator-based true random number generators. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 317–331. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04138-9_23
8. Martin, H.: A new TRNG based on coherent sampling with self-timed rings. IEEE Trans. Industr. Inf. **12**(1), 91–100 (2016)
9. Wold, K., Tan, C.H.: Analysis and enhancement of random number generator in FPGA based on oscillator rings. Int. J. Reconfig. Comput. **2009**, 4 (2009)

10. Drutarovsky, M., Galajda, P.: A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware. In: 2007 17th International Conference on Radioelektronika. IEEE (2007)
11. Majzoobi, M., Koushanfar, F., Devadas, S.: FPGA-based true random number generation using circuit metastability with adaptive feedback control. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 17–32. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23951-9_2
12. Bucci, M., et al.: A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC. IEEE Trans. Comput. **52**(4), 403–409 (2003)
13. Ma, Y., Lin, J., Chen, T., Xu, C., Liu, Z., Jing, J.: Entropy evaluation for oscillator-based true random number generators. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 544–561. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44709-3_30
14. Cherkaoui, A., et al.: A self-timed ring based true random number generator. In: 2013 IEEE 19th International Symposium on Asynchronous Circuits and Systems (ASYNC). IEEE (2013)
15. Chakraborty, R.S., et al.: Hardware Trojan: threats and emerging solutions. In 2009 IEEE International High Level Design Validation and Test Workshop, HLDVT 2009, pp. 166–171 (2009)
16. Sklavos, N., Kitsos, P., Papadomanolakis, K., Koufopavlou, O.: Random number generator architecture and VLSI implementation. In: Proceedings of IEEE International Symposium on Circuits & Systems (IEEE ISCAS 2002), USA, 26–29 May 2002, vol. IV, pp. 854–857 (2002)