A Closer Look at the Chaotic Ring Oscillators based TRNG Design

Shuqin Su[1], Bohan Yang[1], Vladimir Rožić[1,2], Mingyuan Yang[1], Min Zhu[3], Shaojun Wei[1] and Leibo Liu[1,*]

[1] Tsinghua University, Bejing, China.
ssq18@tsinghua.org.cn;bohanyang@mail.tsinghua.edu.cn;
ymy20@mails.tsinghua.edu.cn;{wsj,liulb}@tsinghua.edu.cn

[2] KU Leuven, Leuven, Belgium.
Vladimir.Rozic@esat.kuleuven.be

[3] Wuxi Micro Innovation Integrated Circuit Design Co., Ltd., Wuxi, China.
zhumin@mucse.com

[*] Corresponding author.

Abstract. TRNG is an essential component for security applications. A vulnerable TRNG could be exploited to facilitate potential attacks or be related to a reduced key space, and eventually results in a compromised cryptographic system. A digital FIRO-/GARO-based TRNG with high throughput and high entropy rate was introduced by Jovan Dj. Golić (TC'06). However, the fact that periodic oscillation is a main failure of FIRO-/GARO-based TRNGs is noticed in the paper (Markus Dichtl, ePrint'15). We verify this problem and estimate the consequential entropy loss using Lyapunov exponents and the test suite of the NIST SP 800-90B standard. To address the problem of periodic oscillations, we propose several implementation guidelines based on a gate-level model, a design methodology to build a reliable GARO-based TRNG, and an online test to improve the robustness of FIRO-/GARO-based TRNGs. The gate-level implementation guidelines illustrate the causes of periodic oscillations, which are verified by actual implementation and bifurcation diagram. Based on the design methodology, a suitable feedback polynomial can be selected by evaluating the feedback polynomials. The analysis and understanding of periodic oscillation and FIRO-/GARO-based TRNGs are deepened by delay adjustment. A TRNG with the selected feedback polynomial may occasionally enter periodic oscillations, due to active attacks and the delay inconstancy of implementations. This inconstancy might be caused by self-heating, temperature and voltage fluctuation, and the process variation among different silicon chips. Thus, an online test module, as one indispensable component of TRNGs, is proposed to detect periodic oscillations. The detected periodic oscillation can be eliminated by adjusting feedback polynomial or delays to improve the robustness. The online test module is composed of a lightweight and responsive detector with a high detection rate, outperforming the existing detector design and statistical tests. The areas, power consumptions and frequencies are evaluated based on the ASIC implementations of a GARO, the sampling circuit and the online test module. The gate-level implementation guidelines promote the future establishment of the stochastic model of FIRO-/GARO-based TRNGs with a deeper understanding.

Keywords: FIRO-/GARO-based TRNGs · periodic oscillation · gate-level implementation guidelines · design methodology · online test

1 Introduction

Following the development of mobile computing, Internet of Things (IoT) and even Internet of Everything (IoE), the security and privacy of these systems face more challenges with increased importance. For example, the ubiquity, openness, wireless communication and presence of device failures will bring security and privacy problems for IoT [Sta14]. To ensure the security, cryptographic algorithms and protocols are utilized where True Random Number Generator (TRNG) is an essential block. TRNGs can be used to generate keys, initialization vectors, one-time pads, challenges in authentication schemes and masks against physical attacks [Roz16]. In modern security applications, a TRNG is usually the single point of failure, thus reliability and robustness are two indispensable characteristics for designing and implementing TRNGs [Dic03].

The true randomness of TRNGs originates from unpredictable effects, such as thermal noise, jitters, user behaviors, interruptions and so on. Ideally, adversaries cannot distinguish the output of a TRNG from an uniformly distributed random variable [Yan18]. However, the implementation of an ideal TRNG is not easily achieved. The reliability and robustness of actual TRNGs should be evaluated to satisfy application requirements. TRNGs can be divided into two categories based on their randomness sources, including Physical True Random Number Generators (PTRNGs) or Hardware True Random Number Generators (HTRNGs) with the randomness sources in hardware and Non-Physical True Random Number Generators (NPTRNGs) with the randomness sources in software.

There are various HTRNG designs, such as the classical ring oscillator TRNG based on phase jitters [SMS07], PLL-based TRNG [DSFC04] and transient effect ring oscillator (TERO) TRNG [VD10]. In 2006, Jovan Dj. Golić proposed a new compact digital HTRNG based on chaotic ring oscillators: Fibonacci ring oscillator (FIRO) and Galois ring oscillator (GARO) [Gol06], where the pseudo and true randomness are blended as the form of oscillating signals. The true randomness of FIRO-/GARO-based TRNGs mainly comes from the unpredictable variations in the delays of internal logic gates and internal metastability [DG07]. The influence of external noise on the behavior of entropy source also exists as a worst case for randomness estimation, and the fusion between pseudo and true randomness is unwanted for testability. However, the fusion makes it difficult to compromise TRNG by manipulating external noise.

Designers tend to build a stochastic model for a TRNG design in the last two decades. However, there is no stochastic model proposed for the TRNGs based on FIROs and GAROs and it is difficult to build a stochastic model due to the chaotic property of the TRNGs. Thus, it is hard to analyse the circuit behaviors and evaluate the quality of FIRO-/GARO-based TRNGs with a precise method. The lack of stochastic model makes FIRO-/GARO-based TRNGs fail to meet the requirement of PTG2 of AIS 31 [KS11] and limited in the applications with stringent requirements, such as key generation. However, these TRNGs comply with PTG1 of AIS 31 and NIST SP 800-90B [TBK+18] and can be utilized for the applications with less precision, such as masking.

In normal situations, the randomness of a FIRO-/GARO-based TRNG is propagated, transformed and enhanced through feedback, causing chaotic oscillations with high output speed and high entropy rate. However, there is a security risk from periodic oscillations instead of chaotic oscillations [Dic15], which implies the risk of using low-reliable FIRO-/GARO-based TRNG for cryptographic applications. Under the periodic oscillations, the circuit states formed by the binary outputs of all the cascaded inverters in a FIRO or GARO and the generated random numbers present a certain periodicity, which may cause reduced entropy and an unreliable TRNG. However, the influence of periodic oscillations on the quality of FIRO-/GARO-based TRNGs has not been clarified in existing researches.

For the analysis of the reasons for periodic oscillations, Markus Dichtl tested several conditions and excluded the primitive criterion of feedback polynomials and length of FIRO as the causes [Dic15]. However, the intrinsic reasons for periodic oscillations

were not figured out. In addition, Schramm Martin et al. conducted an experimental assessment for FIRO-/GARO-based noise sources and illustrated the relationship between periodic oscillations and the amount of contributing XOR connections that maximizing the amount may minimize the probability of periodic oscillations [SDH17]. However, the reasons for periodic oscillations and the relationship are not analysed in [SDH17], which is just an experimental result. Yunfan Yang et al. designed a combined structure to reduce the probability of periodic oscillations by increasing jitters with more elements and feedback loops in the circuit [YJW+17], which didn't analyse the intrinsic reasons and lacked persuasiveness for the solution. Other improved entropy sources are also proposed to improve the chaotic property, but the analysis of periodic oscillations is still missing [LJZ19, WYZL20].

In this paper, we are committed to solving the above problems of the influence clarification, the analysis of reasons and solutions of periodic oscillations. The analysis of reasons advances the establishment of the stochastic models for FIRO-/GARO-based TRNGs. Our contributions are shown as follows.

1. We improve the detection method from [Dic15] to achieve a higher detection rate of periodic oscillations. The entropy estimation from the test suite of NIST SP 800-90B [HM15] enables us to quantify the entropy loss caused by periodic oscillations. The experimental result demonstrates the risk of using compromised FIRO-/GARO-based TRNGs.

2. We are the first to use Lyapunov exponent and bifurcation diagram for analyzing the oscillating behaviors of FIRO-/GARO-based TRNGs. The Lyapunov exponents describe the non-chaotic and chaotic properties of periodic and chaotic oscillations respectively. Bifurcation diagrams verify our cause analysis for periodic oscillations.

3. An in-depth analysis, based on a gate-level (netlist) model with TSMC $28nm$ as the technology, was carried out to inquire into the unwanted periodic oscillations in the compromised chaotic FIRO-/GARO-based TRNGs. The platform parameters of this gate-level model are collected from implementations on mainstream commercial FPGAs (Xilinx Artix-7 FPGAs). A case study on these gate-level models is used to derive implementation guidelines for GARO-based TRNGs.

4. A design methodology is proposed to implement a reliable GARO-based TRNG. In this methodology, the feedback polynomials are evaluated to select a suitable feedback polynomial without periodic oscillations, where the proposed algorithm is realized by a Python script*. Besides, the occasional periodic oscillations caused by active attacks and delay inconstancy can be eliminated by delay adjustment to improve the robustness with little resource consumption.

5. An online test module is designed to detect periodic oscillations and can improve the robustness of FIRO-/GARO-based TRNGs against potential active attacks and the delay inconstancy, by adjusting feedback polynomials or delays to eliminate detected periodic oscillation. The online test module is composed of a lightweight and responsive detector with a high detection rate, which outperforms the existing detector design in [Dic15] and statistical tests.

6. The areas, power consumptions and frequencies of a GARO, the sampling circuit, and the online test module are evaluated in ASIC implementations with TSMC $28nm$ as the technology. The areas/power consumptions of the implemented GARO, sampling circuit and online test module are $80.3\mu m^2(160GE)/0.063mW$,

*The Python script is available on GitHub: https://github.com/ybhphoenix/ACloserLook_FIGARO

$58.3\mu m^2(116GE)/0.025mW$ and $2033.6\mu m^2(4035GE)/0.042mW$ respectively. Besides, the sampling circuit and online test module can work at a frequency higher than $2.8GHz$.

This paper is organized as follows: Section 2 introduces FIRO and GARO, FIRO-/GARO-based TRNGs, implementations of FIRO/GARO on FPGAs, Lyapunov exponent and bifurcation diagram, and the compliance with modern standards. Section 3 describes periodic oscillation problem, including full-state sampling, the existing and modified detector for periodic oscillations, the quantification and the characterizations of non-chaotic and chaotic property with Lyapunov exponents. Section 4 proposes the gate-level implementation guidelines based on the analysis of gate-level model, where the strong relation between delays and periodic oscillations is also illustrated by bifurcation diagram. Section 5 verifies the gate-level implementation guidelines by statistical analysis. Section 6 proposes a design methodology to implement reliable GARO-based TRNGs and describes the ASIC implementations of a GARO and the sampling circuit. Section 7 introduces an online test module to improve the robustness of FIRO-/GARO-based TRNGs with adjustable feedback polynomial or delays. Section 8 concludes the paper.

2 Background

2.1 FIRO and GARO

The FIRO and GARO designed by Jovan Dj. Golić [Gol06] are multiple feedback structures as shown in Figure 1 and Figure 2 respectively. They are implemented by replacing the flip-flops in Fibonacci and Galois linear feedback shift registers (LFSRs) with inverters. The arrangement of feedback taps in a FIRO or GARO can be expressed by a polynomial mod 2, which is called feedback polynomial. The coefficients of a feedback polynomial are binary referred to as feedback coefficients. The feedback structure of a FIRO or GARO is determined by the applied feedback polynomial. The feedback polynomial $f(x)$ is defined as Equation (1), where r is the number of cascaded inverters, x^i represents the i-th tapped bit corresponding to the i-th feedback tap, and f_i is the i-th feedback coefficient. The index i is counted from the right for a GARO and from the left for a FIRO. If $f_i = 1$, the

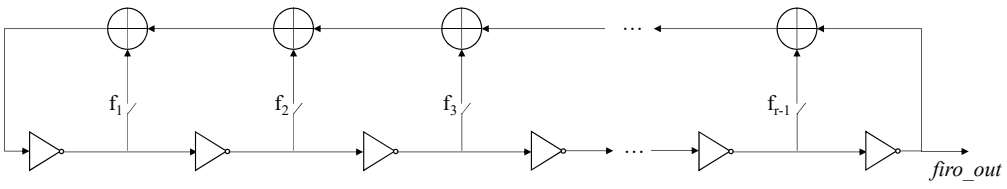


Figure 1: The structure of FIRO.

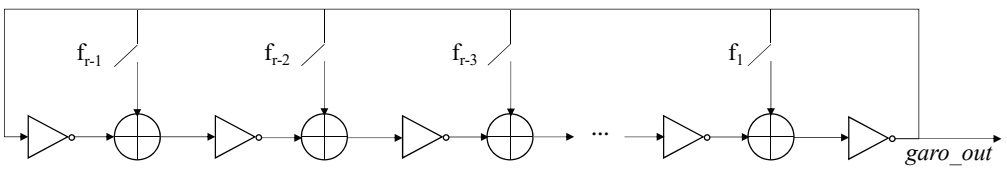


Figure 2: The structure of GARO.

i-th feedback path is closed. Otherwise, the i-th feedback path is open. The input of the leftmost inverter is the feedback signal derived from the output of the rightmost inverter with $f_r = f_0 = 1$.

$$f(x) = \sum_{i=0}^{r} f_i x^i, \text{ where } f_i \in \{0, 1\} \text{ and } f_r = f_0 = 1. \tag{1}$$

A FIRO or GARO may enter a fixed state referred to as a fixed point and stop oscillating with some specific feedback polynomials. For a FIRO, there are two situations with a fixed point as follows [Gol06]:

- If the generated feedback value fed into the leftmost inverter is 1, the state 0101... will be a fixed point. In more detail, if $\sum_{0<j<r} f_j = 0$ or $\sum_{0<j<r} f_j = 1$ where j is even corresponding to the state bits with value 1s, the state 0101...1 or 0101...0 will be a fixed point with the feedback value 1.

- If the generated feedback value fed into the leftmost inverter is 0, the state 1010... will be a fixed point. In more detail, if $\sum_{0<j<r} f_j = 1$ or $\sum_{0<j<r} f_j = 0$ where j is odd, the state 1010...1 or 1010...0 will be a fixed point with the feedback value 0.

The rightmost bit of a fixed point is determined by the parity of r of the FIRO. For example, if r is odd, the specific values of the two fixed points 0101... and 1010... are 0101...0 and 1010...1 with 0 and 1 as the rightmost bits respectively. Thus, to prevent a fixed point for a FIRO, the feedback polynomials should satisfy:

1. If the number of inverters r is odd, both the Hamming weights (HWs) of the feedback coefficients at even positions and odd positions are even, except f_r and f_0.

2. If r is even, both the HWs of the feedback coefficients at even positions and odd positions are odd, except f_r and f_0.

For a GARO, the two situations with a fixed point are as below [Gol06]:

- If r is even, the state 1010...0 will be a fixed point with the feedback value 0.

- If r is odd and $\sum_{j=1}^{r-1} f_j = 1$, the state obtained by the bitwise XOR of $0, 1, 0, 1, ...$ and $0, f_{r-1}, f_{r-2}, f_{r-3}, ..., f_1$ will be a fixed point with the feedback value 1.

In other words, if r is even or the HW of all the feedback coefficients is odd when r is odd, a fixed point will be generated for the GARO. Thus, to prevent a fixed point for a GARO, the feedback polynomials should meet:

1. The number of inverters r is odd.

2. The HW of all the feedback coefficients is even.

The conditions to prevent a fixed point for a FIRO and GARO can also be expressed by Equation (2) and Equation (3) respectively, as proposed in [Gol06]. The Equation (2) also means that $f(x)$ is divisible by $1 + x$ and the quotient polynomial $h(x)$ is not divisible by $1 + x$.

$$f(x) = (1 + x)h(x) \text{ and } h(1) = 1. \tag{2}$$

$$f(1) = 0 \text{ and } r \text{ is odd.} \tag{3}$$

2.2 FIRO-/GARO-based TRNGs

Chaotic systems are able to be utilized to construct Pseudo-random number generators (PRNGs) [HKVC22a, HKVC22b] and TRNGs [YSV04a, YSV04b, CD13]. FIRO-/GARO-based TRNGs are based on the chaotic system formed by FIROs or GAROs, where true randomness fuses pseudo-randomness [Yan18, Gol06]. True randomness of FIRO-/GARO-based TRNGs mainly originates from unpredictable variations in the delays of internal logic gates which will be propagated and enhanced through feedbacks. Internal metastability will also contribute to the true randomness [Gol06, DG07]. The pseudo-randomness is from the feedback structure similar to LFSR [FLL+18]. The fusion between true randomness and pseudo-randomness is normally unwanted for testability, but it increases the difficulty to compromise TRNG by manipulating external noise.

A TRNG consists of entropy source, digitization module, post-processing module, total failure test module and online test module. The entropy sources of FIRO-/GARO-based TRNGs are FIROs, GAROs or the combinations of FIROs and GAROs, which are determined by the specific TRNG architectures. An XOR combination of a FIRO and a GARO (FIGARO) is introduced to improve the randomness and robustness [DG07]. One FIRO or GARO can also be applied as the entropy source of a TRNG with low requirement. Online test module is utilized to detect the failure of entropy source [YRG+18] and trigger an alarm to prevent insecure random numbers generated from the faulty entropy source.

Entropy source is the only source of true randomness in a TRNG. A vulnerable entropy source will result in an unreliable TRNG. In normal situations, the random delays and transition time of the logic gates in FIROs and GAROs influenced by internal and external noise make the oscillation signals irregular with true randomness and pseudo-randomness on binary and analog levels. For example, the random non-zero transition time results in various signal amplitudes, such as the low amplitudes of the short signals without enough time to arrive at the complete digital levels. Besides, the random delays randomize the phase relationship of the two inputs of XOR gates causing random outputs of XOR gates. All the random variations are further propagated and enhanced by the feedback loops leading to the chaotic property of oscillation signals in normal situations. However, if the oscillation of a FIRO or GARO presents a certain periodicity during sampling random numbers, the sampled random numbers will present regularity with low attack difficulty.

2.3 Implementations of FIRO/GARO on FPGAs

FIROs and GAROs can be implemented on FPGAs only with digital logic gates. The implementation details of a FIRO and GARO are shown in Figure 3 and Figure 4 respectively. One inverter is replaced by a NAND gate to enable and disable oscillations. All the

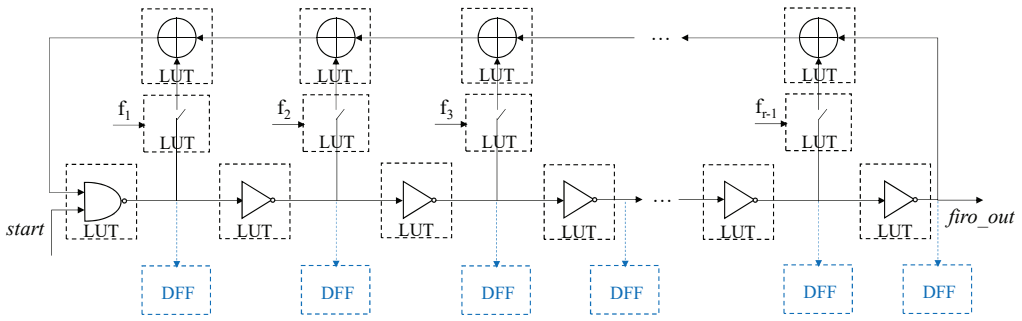


Figure 3: The implementation of a FIRO on FPGAs.

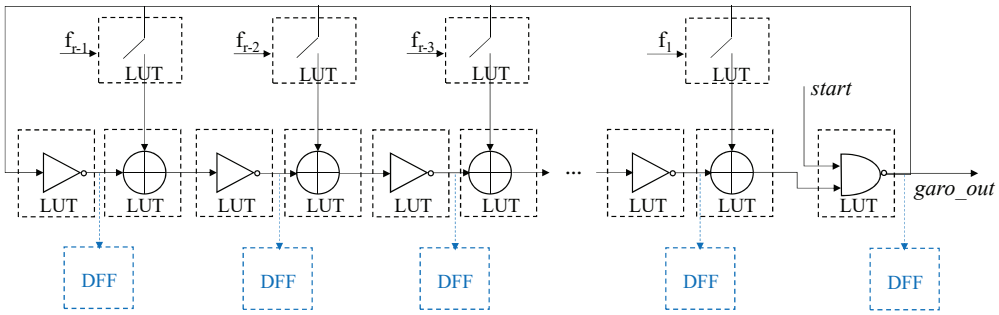


Figure 4: The implementation of a GARO on FPGAs.

combinational logic gates including inverters, NAND gates, XOR gates and switches are implemented by LUTs.

2.4 Lyapunov exponent and bifurcation diagram

Lyapunov exponents can quantify the sensitivity of a dynamical system to the initial conditions or the instability with small changes in the initial conditions [Meh19, VCC09], which describes the average divergence rate of two nearby trajectories. The dynamical system with positive Lyapunov exponents is sensitive to initial conditions with chaotic property [VCC09]. A chaotic TRNG should have at least one positive Lyapunov exponent with high sensitivity to initial conditions. On the contrary, if all the Lyapunov exponents are negative, the TRNG is insensitive to initial conditions with non-chaotic property.

The largest Lyapunov exponent can be calculated from small data sets based on a delayed reconstruction and the selection of the nearest point. The specific operations are shown as follows [RCD93]:

1. Reconstructing from a single time series. The reconstructed trajectory $\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2\ ...\ \mathbf{X}_M]^T$ is a matrix, where each row is a phase-space vector, M is the number of reconstructed points and $\mathbf{X}_i$ is the system state at discrete time i. $\mathbf{X}_i = [x_i\ x_{i+J}\ ...\ x_{i+(m-1)J}]$ is obtained from an N-point time series $\{x_1, x_2, ..., x_N\}$, where J is the reconstruction delay lag and m is the embedding dimension. Thus $\mathbf{X}$ is an $M \times m$ matrix with $M = N - (m-1)J$.

2. Finding the nearest neighbor of each point on the trajectory. The nearest neighbor $\mathbf{X}_{\hat{j}}$ is the point in the reconstructed trajectory $\mathbf{X}$ that minimizes the distance to the reference point $\mathbf{X}_j$, expressed as $d_j(0) = min_{\mathbf{X}_{\hat{j}}} \|\mathbf{X}_j - \mathbf{X}_{\hat{j}}\|$, where $d_j(0)$ is the initial distance, $\|..\|$ is the Euclidean norm and $|j - \hat{j}|$ is larger than the mean period of power spectrum of the time series.

After the reconstruction and selection, the largest Lyapunov exponent is calculated as in Equation (4), where Δt is the sampling period of the time series, $d_j(0)$ is the initial distance between the j-th pair of nearest points, and $d_j(i)$ is the distance between the j-th pair after i discrete-time steps or $i\Delta t$ seconds [SSS87, RCD93]. In this equation, $M - i$ pairs of nearest points are averaged to calculate the largest Lyapunov exponent with a suitable discrete-time step.

$$\lambda(i) = \frac{1}{i\Delta(t)} \frac{1}{(M-i)} \sum_{j=1}^{(M-i)} ln \frac{d_j(i)}{d_j(0)} \tag{4}$$

Bifurcation diagram is another critical method to characterize chaotic systems, which can describe the sensitivity of chaotic systems to design parameters. In bifurcation diagram, all the sampled values of a variable representing system behavior are plotted as a function of a specific design parameter with other parameters fixed, which illustrates the influence of the design parameter on system behavior. With different parameter values, the system will present non-chaotic or chaotic behavior. For non-chaotic behavior, the regularity of system behavior will make system variable keep to a definite range of possible values, causing duplicates in sampled values with a small number of different samples. On the contrary, the chaotic system behavior will make the sampled values spread throughout the diagram with a large number of different values. The few definite variable values or spots in bifurcation diagram illustrate the existence of fixed points or short periodic orbits. The part covered by a large number of scattered spots is called a black region, corresponding to chaotic behavior. To maintain chaotic behavior, we should select the parameter values in the black region [Lam18]. Further, choosing the design parameter value at the center of the black region can improve the robustness to parameter variations [CD13].

2.5 Compliance with modern standards

There are two main approaches to evaluate the qualities of RNGs in modern standards, including AIS 20/AIS 31 with Shannon entropy or conditional entropy and NIST SP 800-90 with min-entropy. AIS 31 and NIST SP 800-90B are the test suites for TRNGs. PTG 2 of AIS 31 is a more stringent but more risky method with the requirement of stochastic model to estimate entropy, where the qualities of entropy estimation, dedicated test and the evaluated TRNG are related to the corresponding stochastic model. PTG 1 of AIS 31 and NIST SP 800-90B is a simpler approach with less precision for entropy evaluation but with less risk, where stochastic models are not prerequisite and the entropy is estimated using tests.

For FIRO-/GARO-based TRNGs, it is difficult to build a stochastic model to describe the chaotic property and no stochastic model is introduced in existing researches. Thus, these TRNGs are non-compliant with PTG 2 of AIS 31, but can be evaluated with PTG 1 of AIS 31 and NIST SP 800-90B without the requirement of stochastic model. According to the PTG 2 of AIS 31 standard, FIRO-/GARO-based TRNGs cannot be used for the generation of cryptographic keys due to the incomplete evaluation. However, these TRNGs can be utilized for the applications without stringent requirement, such as masking.

The entropy estimation of NIST SP 800-90B can be utilized to measure the compromise of FIRO-/GARO-based TRNGs caused by periodic oscillations. For the entropy estimation, the track of the evaluated TRNG should be determined first, including IID (independent and identically distributed) and non-IID tracks. The tracks of FIRO-/GARO-based TRNGs are non-IID track, due to the dependency between the previous and current outputs. The entropy result of a set of sampled random numbers is the minimum entropy among the results estimated by multiple estimation methods, including the Most Common Value Estimate, Collision Estimate, Markov Estimate, Compression Estimate and so on [HM15].

3 Periodic oscillation problem

3.1 Full-state sampling

Markus Dichtl in [Dic15] analyzed the behavior of FIROs by sampling the circuit states which we refer to as full-state sampling. The full-state sampling means that the outputs of all inverters and NAND gate in a FIRO or GARO are sampled by registers simultaneously at a certain frequency. The implementations of full-state sampling for a FIRO and GARO are shown in Figure 3 and Figure 4, where D flip-flops (DFFs) are applied for sampling.

The random numbers are generated by sampling the output $firo\_out$ or $garo\_out$ with a DFF.

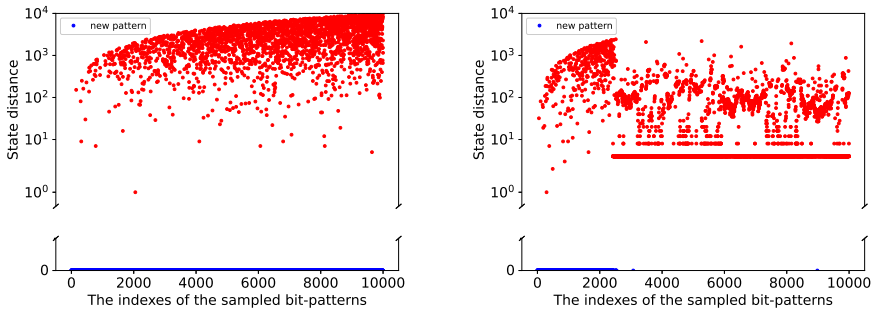
Periodic oscillations may cause reduced entropy of the generated random numbers and the low reliability of the TRNGs based on FIROs and GAROs [Dic15]. We repeat the experiments from [Dic15] on a GARO with 15 cascaded inverters on Xilinx Artix-7 FPGAs. The rightmost inverter is displaced by a NAND gate for control. The sampled outputs of the 14 inverters and NAND gate form the 15-bit state with $2^{15} = 32768$ possible values. For this GARO, there are 14 optional feedback coefficients and $2^{13} = 8192$ feedback polynomials without a fixed point. One feedback coefficient is determined by the other 13 feedback coefficients to realize even HW of feedback coefficients to prevent a fixed point. We select two feedback polynomials as in Equation (5) and Equation (6) in the 8192 feedback polynomials to illustrate the difference between chaotic and periodic oscillations. The results of full-state sampling are shown in Figure 5(a) and Figure 5(b) with the indexes of sampled bit-patterns or states in 10000 samples as the x-axis and the logarithmic form of state distances as y-axis. State distances represent the number of samples between the current sampled state and the last same sampled state in all the samples, which is formalized by Equation (7). In Equation (7), i and j are the indexes of sampled states, $Dis(i)$ is the state distance for the i-th sample, and s_i and s_j represent the i-th and j-th sample values respectively. The state distance of a new bit-pattern of state sampled for the first time is 0, which is represented by a blue dot on x-axis. The results of state distances are similar to the results in [Dic15].

In Figure 5(a), the state distances present no regularity and new bit-patterns are generated throughout the whole sampling process. On the contrary, the state distances tend to several certain values with regularity and much fewer new bit-patterns are generated after 3000 samples in Figure 5(b). It means that some states appear repeatedly with certain intervals and the number of different bit-patterns is reduced compared with Figure 5(a). The regularity in Figure 5(b) illustrates the occurrence of periodic oscillation after 3000 samples for the feedback polynomial expressed by Equation (6).

$$f(x) = x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^6 + x^4 + x^3 + x^2 + 1. \tag{5}$$

$$f(x) = x^{15} + x^{14} + x^{12} + x^{11} + x^7 + x^3 + x + 1. \tag{6}$$

$$Dis(i) = \begin{cases} 0, \ s_i \text{ is a new pattern} \\ min\{i - j \mid s_i = s_j \text{ and } j < i\}, \text{ else} \end{cases} \tag{7}$$



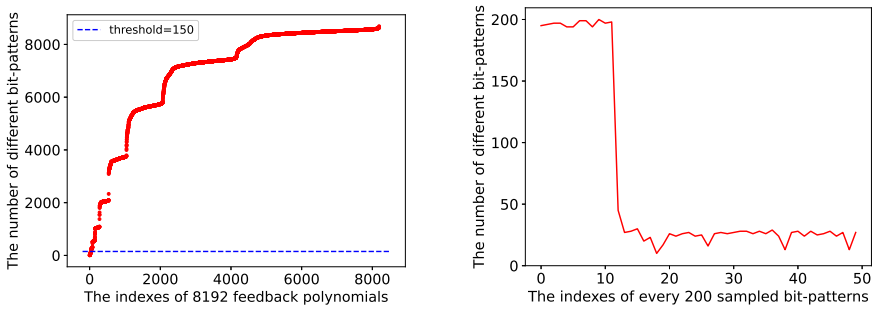
(a) State distances for Equation (5) with chaotic oscillation.

(b) State distances for Equation (6) with periodic oscillation.

Figure 5: The measured results of full-state sampling

3.2 Existing and modified detector

For the periodic oscillations, a conceptual solution is proposed in [Dic15] to detect periodic oscillations and change the applied feedback polynomial until no periodic oscillation is detected. The detector is based on the principle that the same circuit states will occur and be sampled repeatedly with fewer new bit-patterns generated in periodic oscillations, causing a small number of different bit-patterns in all the samples. If the number of different bit-patterns in all samples is smaller than a specific threshold, such as the number smaller than 100 in 10000 samples [Dic15], periodic oscillation is considered to have occurred during sampling. Selecting the threshold value is actually making a trade-off between the Type I and Type II error for the Null hypothesis that the GARO is chaotic. In our experiments, we select 150 as the threshold to detect continuous periodic oscillation throughout the whole sampling. The threshold 150 are determined empirically, which is larger than the threshold 100 in [Dic15] to reduce Type II error. The small value 150 of threshold also maintains the low probability of false alarm for periodic oscillations. We aim to analyze periodic oscillation which will greatly compromise TRNGs with a small number of different state values. The followed entropy and Lyapunov exponent analysis validate our selection of threshold[†].



(a) The number of different bit-patterns in 10000 samples for all the configurations.

(b) The number of different bit-patterns in every 200 samples with Equation (6).

Figure 6: The measured results of the number of different bit-patterns

As shown in Figure 6(a) which indicates the sorted result of the numbers of different bit-patterns in 10000 samples for all the 8192 feedback polynomials, the numbers of different bit-patterns are smaller than 150 for several feedback polynomials. The small number of different bit-patterns is caused by the repeated appearance of a large number of same bit-patterns, which can be utilized to detect periodic oscillations.

However, only utilizing the detector with a low threshold as in [Dic15] will cause a high possibility of missed detection for the intermittent periodic oscillations mixed with chaotic oscillation. An intermittent periodic oscillation will only cause a certain reduction in the number of different bit-patterns and the degree of reduction is determined by the duration of periodic oscillation. A short periodic oscillation will only cause a small reduction of the number of different bit-patterns in all samples, and the reduced number is not small enough to be detected causing missed detection. To decrease the probability of missed detection, we propose a modified detector to detect intermittent periodic oscillations. In the modified detector, all samples are divided into multiple parts and the number of different bit-patterns in each part instead of the number in all samples is calculated to detect periodic oscillations. The principle and feasibility of the modified detector are

[†]The measured data for all the feedback polynomials is available on GitHub, including the challenges, min-entropy, Lyapunov exponents and the numbers of different states in 10000 samples: https://github.com/ybhphoenix/ACloserLook_FIGARO

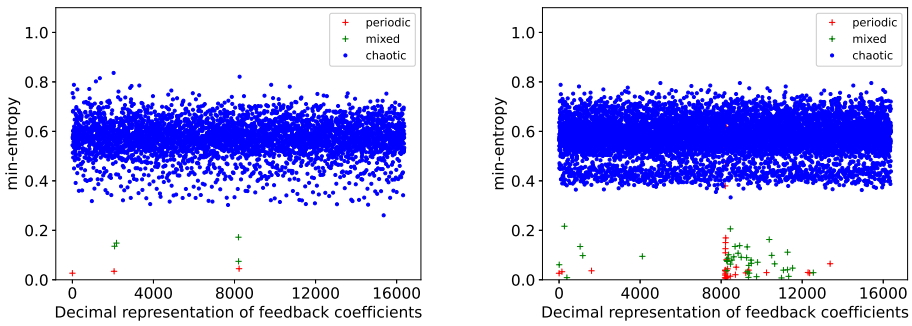
reflected in Figure 6(b), where the 10000 samples are divided into 50 parts and the number of different bit-patterns in a part with 200 samples is reduced a lot when a periodic oscillation occurs with Figure 5(b) as the reference.

With the modified detector, the intermittent periodic oscillations can be detected even with a low threshold to prevent false alarm, which improves the detection rate. Besides, the start and end positions of a periodic oscillation can be determined, except judging whether a periodic oscillation has occurred.

3.3 Quantification

The influence of periodic oscillations on entropy is not quantified, causing the ambiguous impact of periodic oscillations on the qualities of FIRO-/GARO-based TRNGs and lack of persuasiveness for the claim about the risk of periodic oscillations in existing researches. Thus, we need to prove that the periodic oscillations indeed cause compromised TRNGs with low entropy. A GARO and FIRO with 15 cascaded inverters are implemented to test the influence of periodic oscillations on entropy. For this FIRO, there are 14 optional feedback coefficients and $2^{12} = 4096$ feedback polynomials without a fixed point. Except f_r and f_0, one feedback coefficient at an even position is determined by the other 6 feedback coefficients at even positions to realize the even HW of the feedback coefficients at even positions, and one feedback coefficient at an odd position is determined by the other 6 feedback coefficients at odd positions to realize the even HW of the feedback coefficients at odd positions.

We first distinguish the periodic and chaotic oscillations with the detector described in Subsection 3.2, where the continuous periodic oscillations throughout the sampling process are detected based on the number of different sampled states with the threshold 150 of 10000 samples, and the intermittent periodic oscillations mixed with chaotic oscillations are detected based on the change of the number of different sampled states in every 200 samples. The entropy of generated random numbers is estimated by the test suite of NIST SP 800-90B standard. The min-entropy results for all the configurations of the FIRO and GARO are shown in Figure 7(a) and Figure 7(b) with the decimal representation of the binary feedback coefficients $f_{14}f_{13}...f_1$ as x-axis and estimated min-entropy as y-axis. The detected continuous periodic oscillations, intermittent periodic oscillations mixed with chaotic oscillations, and chaotic oscillations are marked with red, green and blue dots respectively. As shown in Figure 7, all the reduced entropy is caused by periodic oscillations. The large reduction in entropy caused by periodic oscillation illustrates that periodic oscillation is a main failure of the TRNGs based on FIROs and GAROs. Besides,



(a) The min-entropy results for FIRO.

(b) The min-entropy results for GARO.

Figure 7: The min-entropy results for the implemented FIRO and GARO with periodic and chaotic oscillations.

all the periodic oscillations with entropy loss are detected successfully, which verifies the effectiveness of the selected threshold and modified detector in Subsection 3.2 for all the feedback polynomials. 0.195% of periodic oscillations mixed with chaotic oscillations has high entropy. This phenomenon implies the limitation of using statistical tests to estimate entropy, where pseudo-randomness cannot be distinguished from true randomness.

3.4 Characterization with Lyapunov exponent

As discussed in Subsection 2.4, the largest Lyapunov exponent of a system indicates whether this system is sensitive to its initial conditions. We calculate the largest Lyapunov exponents for all the 8192 feedback polynomials corresponding to the entropy estimated in Subsection 3.3. For one feedback polynomial, the largest Lyapunov exponent is calculated using Algorithm 1.

Algorithm 1 The calculation of the largest Lyapunov exponent for FIRO-/GARO-based TRNG

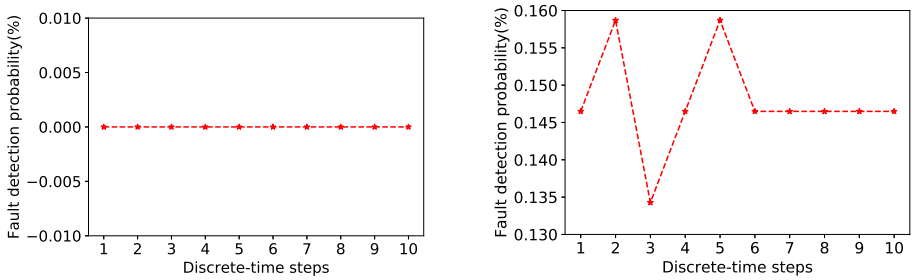
Input: 10000 15-bit sampled states $S = \{s_0, s_1, ..., s_{9999}\}$.
Output: Largest Lyapunov exponent.

1: Obtain the input time series $\{x_0, x_1, ..., x_{9999}\}$, where the i-th value $x_i = HD(s_0, s_i)$.
2: Select the parameters for reconstruction and calculation of largest Lyapunov exponent, including the time delay lag, embedding dimension m, discrete time step i and sampling step Δt.
3: Reconstruct the trajectory $\mathbf{X}$ with $M = N - (m-1)lag$ as described in Subsection 2.4, based on the time delay lag, embedding dimension m, and input time series with $N = 10000$ samples.
4: Find the corresponding nearest point for each point in the $M - i$ points $\{\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_{M-i}}\}$ from reconstruction, as described in Subsection 2.4.
5: Select a suitable i.
6: Calculate the largest Lyapunov exponent $\lambda(i)$ by Equation (4).

We select Hamming distances (HDs) of sampled states instead of states themselves as the input time series of reconstruction, which is due to that Hamming distance can better reflect the magnitude of the changes in circuit states. For the parameter selection in Algorithm 1, the time delay lag is estimated using average mutual information(AMI) [KR11], where lag is the first local minimum of AMI. AMI is a certain generalization of autocorrelation function as represented by Equation (8), where $p(x_i)$ is the occurrence probability of the i-th value x_i in the input time series, $p(x_{i+T})$ is the occurrence probability of the $(i+T)$-th value x_{i+T} after the time delay T, and $p(x_i, x_{i+T})$ is the associated probability of co-occurrence of x_i and x_{i+T}. The embedding dimension m is determined by the order of implemented GARO. Both lag and m are 15 in our case study. Besides, the sampling step Δt is 1 for our discrete system. The value of discrete-time step i is determined by experiment with the smallest fault detection probability or probability of mismatch between oscillation types and calculated Lyapunov exponents. The discrete-time step i equals 3 for our implemented GARO with the smallest fault detection probability 0.134% as shown in Figure 8. For the implemented FIRO, the fault detection probabilities are 0s for all the tested discrete-time steps. We take $i = 3$ as an example to display the Lyapunov exponent results for FIRO.

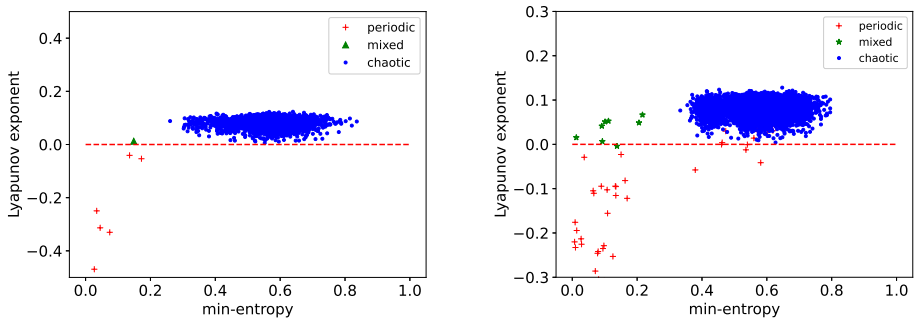
$$AMI(T) = \sum_{x_i, x_{i+T}} p(x_i, x_{i+T}) log_2[\frac{p(x_i, x_{i+T})}{p(x_i)p(x_{i+T})}] \tag{8}$$

After determining the value of i, the Lyapunov exponents are calculated and the results are shown in Figure 9 with estimated min-entropy as the x-axis. The periodic and chaotic



(a) The fault detection probabilities for FIRO. (b) The fault detection probabilities for GARO.

Figure 8: The fault detection probabilities for FIRO and GARO with different i.



(a) The Lyapunov exponents for FIRO. (b) The Lyapunov exponents for GARO.

Figure 9: The Lyapunov exponents for FIRO and GARO with periodic, mixed and chaotic situations.

oscillations are distinguished by the detector described in Subsection 3.2, with the selected threshold 150 and modified detector to detect periodic oscillations. To better analyse the consistency between entropy and Lyapunov exponents, we divide periodic oscillations into periodic situations and mixed situations, where the duration of chaotic oscillation accounts for [75%, 100%) of the sampling time in mixed situations. We can obtain the following conclusions from the results:

- All the periodic situations with entropy loss have negative Lyapunov exponents illustrating the non-chaotic property, and all the chaotic situations have positive Lyapunov exponents indicating the chaotic property for our implemented GARO and FIRO, which reflects the high accuracy of the Lyapunov exponent calculation.
- The low estimated entropy and positive Lyapunov exponents of some mixed situations are due to that a short period of periodic oscillation will greatly reduce the estimated entropy and Lyapunov exponents are obtained by analysing all the sampled states.
- The negative Lyapunov exponents having relatively high estimated min-entropy of some periodic situations imply that our calculated Lyapunov exponents can detect potential failures of chaotic oscillation where the entropy estimation from NIST could not.
- All the periodic oscillations with negative Lyapunov exponents are detected successfully by the detector described in Subsection 3.2, verifying the effectiveness of the selected threshold and modified detector for all the feedback polynomials.

4 Gate-level implementation guidelines

To analyse the intrinsic reasons for periodic oscillations, we implement a GARO with 15 cascaded inverters as an example and traverse all the 8192 feedback polynomials on Xilinx Artix-7 FPGA. All periodic oscillations with different periodic duration are collected for analysis based on a gate-level model and full-state sampling[‡].

We measure the delay of each stage of the GARO with an asynchronous counter, and observe the regularity between delays and periodic oscillations. To demonstrate the regularity and further analyse the inside rules, we simulate the implemented circuit based on the measured delays with SPICE as the tool and TSMC $28nm$ as the technology. The simulated technology is consistent with the technology of utilized FPGAs. The simulated waveforms of internal circuit signals and bifurcation diagrams illustrate the strong correlation between delays and periodic oscillation.

The analysed rules obtained from the implemented circuit and the simulated circuit can be extended to more general cases with the feedback polynomial $f(x) = x^{r_0} + x^{r_1} + x^{r_2} + ... + x^{r_{l-1}} + x^{r_l} + 1$ as the gate-level implementation guidelines, where l is even to prevent a fixed point for the GARO, the closed feedback paths except the rightmost always closed feedback path are indexed by $r_0, r_1, ...r_l$ decreased from left to right and r_0 is the number of cascaded inverters.

4.1 Notation

- INV_i – The i-th inverter, where $i = \{1, 2, ..., r_0\}$.
- XOR_i – The i-th XOR gate, where $i = \{1, 2, ..., r_0 - 1\}$.
- SW_i – The i-th switch, where $i = \{1, 2, ..., r_0 - 1\}$.
- O'_i – The output of INV_i.
- O_i – The output of XOR_i.
- $I_{i,j}$ – An input of XOR_i, where $j = \{1, 2\}$ as the indexes of the two inputs of an XOR gate.
- $g_{r_0}(f_i)$ – The feedback polynomial with $f_i = f_0 = 1$ and the other feedback coefficients as 0s.
- $D(n', n)$ – The delay difference between two feedback loops with $n > 0$, or the delay of one feedback loop with $n = 0$. It is expressed as Equation 9, where $Delay(g_{r_0}(f_{n'})) = D(n', 0)$ as the delay of the n'-th feedback loop with the feedback polynomial $g_{r_0}(f_{n'})$, $Delay(g_{r_0}(f_0)) = 0$, $0 < n' \leq r_0$, and $0 \leq n < r_0$.

$$D(n', n) = |Delay(g_{r_0}(f_{n'})) - Delay(g_{r_0}(f_n))|, \text{ where } n' > n. \tag{9}$$

- T_n – The oscillation period of the n-th feedback loop, expressed as below:

$$T_n = 2D(n, 0), \text{ where } n = \{1, 2, ..., r_0\}. \tag{10}$$

- δ – The tolerable variation of delay differences for periodicity conditions only causing glitches in the specific outputs. The glitches will be filtered out and a larger filter capacity causes a larger value of δ leading to a larger possibility of periodic oscillations. In our tests, $\delta = 0.38ns$.

[‡]The collected data is available on GitHub: https://github.com/ybhphoenix/ACloserLook_FIGARO

- DIV_δ – If two non-zero delay differences $D(n', n)$ and $D(n'_1, n_1)$ satisfy Equation (11), where p and q are two non-zero integers, the two delay differences are called *almost commensurable*. The corresponding common measure $m' = D(n'_1, n_1)/q$ is a positive real value due to the positive delay differences and positive value of q in our analysis. If each delay difference in $\{D(r_0, r_1),\ D(r_2, r_3),...,D(r_{l-2}, r_{l-1})\}$ is almost commensurable with $D(r_l, 0)$, DIV_δ is the greatest common value among all the common measures.

$$|D(n', n) - D(n'_1, n_1)\frac{p}{q}| < \delta \tag{11}$$

- M_i – The natural number of p with $m' = DIV_\delta$ for the positive delay difference $D(r_{2i}, r_{2i+1}) \geq \delta$, or $M_i = 0$ for $D(r_{2i}, r_{2i+1}) < \delta$, expressed as:

$$|M_i \times DIV_\delta - D(r_{2i}, r_{2i+1})| < \delta, \text{ where } i = \{0, 1, ..., l/2 - 1\} \tag{12}$$

- DIV'_δ – The greatest common value among all the common measures between each delay difference in $\{D(r_0, r_1),\ D(r_2, r_3),...,D(r_{l-2}, r_{l-1})\}$ and $D(r_l, 0)$ except the delay differences satisfying the conditions in Decision rule3.

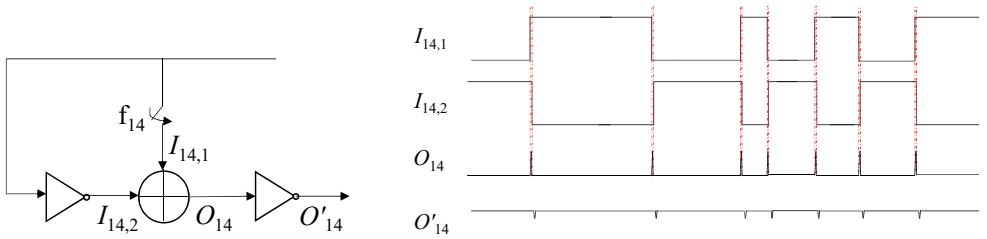
- DIV_{min} – The minimum value of DIV_δ or DIV'_δ. A value not larger than DIV_{min} will make signals tend to be glitches instead of periodic oscillations. In our tests, $DIV_{min} = 0.79ns$.

- M'_i – The natural number of p with $m' = DIV'_\delta$ for a delay difference not smaller than δ or $M'_i = 0$ for a delay difference smaller than δ, where $i = \{0, 1, ..., N - 1\}$ and N is the number of delay differences utilized to calculate DIV'_δ.

4.2 Gate-level model

A gate-level model is built based on T_n obtained from FPGA implementation to analyse the causes of periodic oscillations with SPICE as the tool and TSMC $28nm$ as the technology, which is a gate-level description in the form of a netlist[§]. T_n is measured with an approximate method based on the observation that the sampled values of O'_{14} are heavily biased towards 0s or even fixed at 0s when $f_{14} = 1$.

As shown in Figure 10(a), the input $I_{14,2}$ is the inverse of $I_{14,1}$ with the delay difference $D(15, 14)$ when $f_{14} = 1$. The delay difference $D(15, 14) < \delta$ makes $I_{14,2}$ and $I_{14,1}$ almost toggle simultaneously as shown in Figure 10(b). Thus, O_{14} and O'_{14} are fixed at 1 and 0 only with glitches. The glitches are filtered by INV_{14} as shown with the purer waveform of O_{14} compared with O'_{14} in Figure 10(b). A smaller value of $D(15, 14)$ will cause a larger bias in the sampled values of O'_{14} with fewer glitches.



(a) The leftmost part of GARO circuit.

(b) The waveforms with $f_{14} = 1$.

Figure 10: The leftmost part of GARO circuit and waveforms with $f_{14} = 1$.

[§]One example model is available on GitHub: https://github.com/ybhphoenix/ACloserLook_FIGARO

Once O'_{14} is fixed at 0, a GARO with the feedback polynomial $f(x) = x^{15} + x^{14} + x^m + 1$ will degenerate to a GARO$'$ with the feedback polynomial $f'(x) = x^m + 1$, where $m = \{1, 2, ..., 13\}$. The degenerated GARO$'$ is equivalent to a classical ring oscillator only with one feedback loop. Thus, we implement a GARO with $D(15, 14) < \delta$ to measure T_n. T_m can be measured approximately as $T_m = Per(GARO'(f'(x) = x^m + 1))$ by applying the feedback polynomial $f(x) = x^{15} + x^{14} + x^m + 1$, where $Per(GARO'(f'(x) = x^m + 1))$ represents the measured oscillation period of the degenerated GARO$'$ with the degenerated feedback polynomial $f'(x) = x^m + 1$. Besides, T_{15} and T_{14} can be measured approximately as $T_{15} \approx T_{14} = Per(GARO(f(x) = x^{15} + 1))$, where $Per(GARO(f(x) = x^{15} + 1))$ represents the measured oscillation period of the GARO equivalent to a classical ring oscillator with the feedback polynomial $f(x) = x^{15} + 1$.

An asynchronous counter is utilized to count the number of rising edges of *garo_out* in one clock cycle to calculate T_n, which is composed of cascaded T flip-flops (TFFs) as shown in Figure 11. The asynchronous counter is disabled after each cycle count to stabilize the count values. The calculation of T_n is shown in Equation 13, where T_{clk} is the clock period, N is the number of clock cycles used to count the rising edges and C_j is the count value in the j-th clock cycle. $D(n', n)$ is calculated as Equation 14, which is also the delay difference between the two inputs of XOR_n through the n'-th and n-th feedback paths.

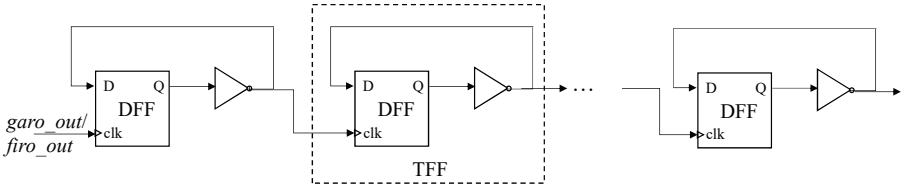


Figure 11: The architecture of an asynchronous counter.

$$T_n = \frac{NT_{clk}}{\sum_{j=1}^{N} C_j} \tag{13}$$

$$D(n', n) = \frac{T_{n'} - T_n}{2} \tag{14}$$

All the measured values of T_n are shown in Table 1. After obtaining T_n, a gate-level model can be built to simulate the timing relationships in the implemented GARO with SPICE as the tool, where the delays of routes are imitated by buffers and the switches are implemented by AND gates as shown in Figure 12. The aspect ratios of buffers are adjusted according to T_n. The feedback polynomials with periodic oscillations in actual tests are applied to the gate-level model to analyse the relationships between delay differences and periodic oscillations.

Table 1: The measured values of T_n. (time unit: ns)

T_{15}/T_{14}	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8
34.4923	32.3499	29.7053	27.9767	25.1206	23.4170	20.8960
T_7	T_6	T_5	T_4	T_3	T_2	T_1
18.6123	16.4995	14.0861	11.0914	8.8842	6.3834	3.7998

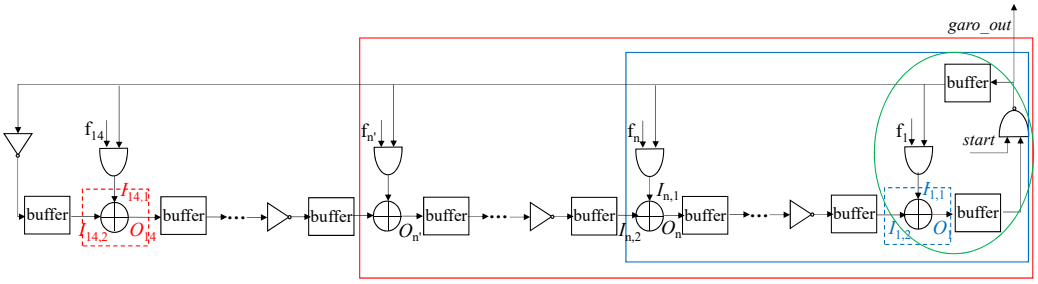


Figure 12: The circuit structure of the gate-level model.

4.3 Deduction rule

Deduction rule. If all the delay differences $D(r_0, r_1), D(r_2, r_3), ..., D(r_{p-2}, r_{p-1})$ are small than δ and $D(r_p, r_{p+1}) \geq \delta$ where p is even, all the outputs O_{r_1}, O'_{r_1}, O_{r_3}, O'_{r_3}, ... , $O_{r_{p-1}}, O'_{r_{p-1}}$ will be fixed at 1s or 0s, and the GARO will degenerate to an equivalent ring oscillator GARO′ with the feedback polynomial $f'(x) = x^{r_p} + x^{r_{p+1}} + ... + 1$. The oscillation situation is determined by GARO′.

Deduction rule is obtained based on the observation described in Subsection 4.2 with the feedback polynomial $f(x) = x^{15} + x^{14} + x^m + 1$. If the two inputs of XOR_n are the same or opposite with the delay difference $D(n', n) < \delta$, O_n and O'_n will be fixed only with glitches and the glitches will be filtered out by the gates and routes between XOR_n and the next XOR gate with feedback input. In Deduction rule, from the leftmost to right for the GARO, the input $I_{r_1,2}$ is the inverse or the same with $I_{r_1,1}$ with the delay difference $D(r_0, r_1)$. If $D(r_0, r_1) < \delta$, O_{r_1} and O'_{r_1} will be fixed no matter how the inputs change causing the degeneration of the GARO. Under the effect of the fixed value of O_{r_1}, XOR_{r_2} will be equivalent to a buffer or an inverter and the two inputs of XOR_{r_3} will be opposite or the same with the delay difference $D(r_2, r_3)$. Thus, if $D(r_2, r_3) < \delta$, O_{r_3} and O'_{r_3} will be fixed causing further degeneration. And so on, the degeneration propagates from the leftmost to right until one delay difference is not smaller than δ. The degenerated ring oscillator has fewer stages and reduced complexity.

To verify the Deduction rule, a feedback polynomial $f(x) = x^{15} + x^{14} + x + 1$ is applied to the built gate-level model as an example, where $D(15, 14) < \delta$. With this feedback polynomial, $I_{14,2}$ is the inverse of $I_{14,1}$ with the delay difference $D(15, 14) < \delta$ which causes the almost simultaneous toggles of $I_{14,2}$ and $I_{14,1}$. Thus, O_{14} and O'_{14} are fixed at 1 and 0 as shown in the red frame in Figure 13, which is consistent with Deduction rule. Under the effect of the fixed values, the GARO degenerates to GARO′ with the feedback polynomial $f'(x) = x + 1$. The equivalent circuit GARO′ is simulated with $I_{1,2}$ connected to VSS to verify the degeneration. The outputs *garo_out* and *garo_deg* of GARO and GARO′ are the same verifying the degeneration, as shown in the blue frame in Figure 13. Thus, the correctness of the Deduction rule is verified. The glitches due to the small non-zero delay difference $D(15, 14)$ are filtered, as illustrated by the purer waveform of O'_{14} or $I_{1,2}$ compared with O_{14}.

For FIROs, the gate-level model establishment and Deduction rule are similar to GAROs. The continuous delay differences smaller than δ will make the degeneration propagate from the rightmost to left until one delay difference is not smaller than δ. Small glitches will be filtered by the next gates and routes.

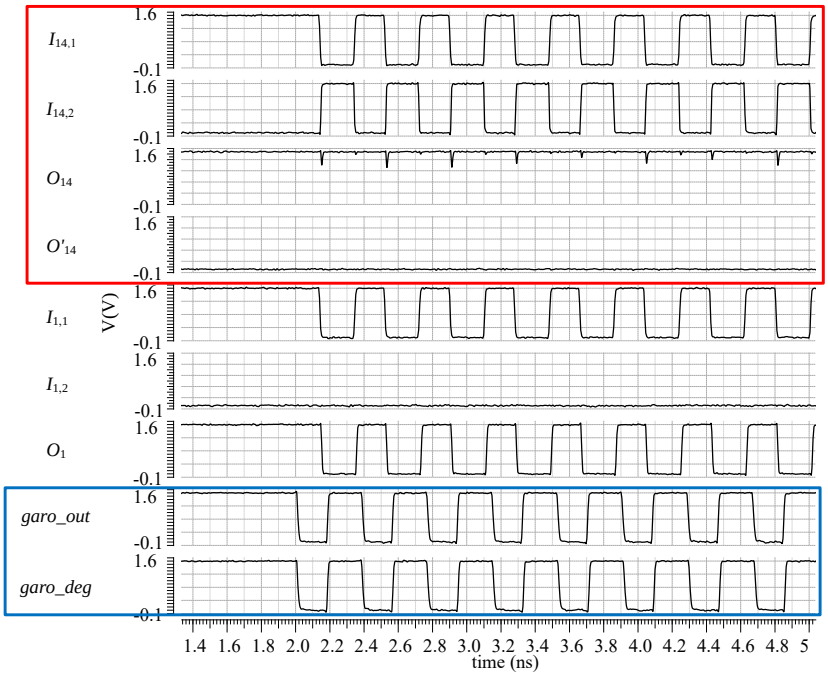


Figure 13: The simulated waveforms with feedback polynomial $f(x) = x^{15} + x^{14} + x + 1$.

4.4 Decision rules

4.4.1 Decision rule1

Decision rule1. If $DIV_\delta > DIV_{min}$, $D(r_l, 0)/DIV_\delta$ is odd, and the sum S of all the values of M_i is even, the outputs $O_{r_1}, O_{r_3}, ..., O_{r_{l-1}}$ tend to be fixed at 1s or 0s, and it is possible for the GARO to degenerate to a classical ring oscillator with the feedback polynomial $f'(x) = x^{r_l} + 1$. In this situation, the GARO will oscillate periodically with $2DIV_\delta$ as the period and 50% as the duty cycle.

The two inputs of XOR_{r_1} are the same or inverse with the delay difference $D(r_0, r_1)$. For $D(r_0, r_1) < \delta$, the corresponding M_0 is 0 and O_{r_1} will be fixed according to Deduction rule. For $D(r_0, r_1) \geq \delta$, a fixed value of O_{r_1} will be generated if the following two conditions are met:

- $|M_0 \times DIV_\delta - D(r_0, r_1)| < \delta$, where $DIV_\delta > DIV_{min}$.
- There is a short period of periodic toggles in *garo_out* with $2DIV_\delta$ as the period, and 50% as the duty cycle.

The periodic toggles are transmitted to the two inputs of XOR_{r_1} with the delay difference $D(r_0, r_1)$, and condition 1 causes the almost simultaneous toggles in the two inputs leading to the fixed value of O_{r_1}. The fixed value of O_{r_1} will cause the GARO degenerate to GARO$'$ with the feedback polynomial $f'(x) = x^{r_2} + ... + x^{r_{l-1}} + x^{r_l} + 1$ and the two inputs of XOR_{r_3} are the same or inverse with the delay difference $D(r_2, r_3)$. Once $|M_1 \times DIV_\delta - D(r_2, r_3)| < \delta$, a fixed value of O_{r_3} will also be generated causing a further degeneration. And so on, under the effect of the even value of S, the fixed value of $O_{r_{l-1}}$ will make the GARO degenerate to a classical ring oscillator GARO$''$ with the feedback polynomial $f''(x) = x^{r_l} + 1$ and odd number of inverter functions finally. If $D(r_l, 0)/DIV_\delta$ is odd, the degenerated GARO$''$ can oscillate periodically with $2DIV_\delta$ as the period and

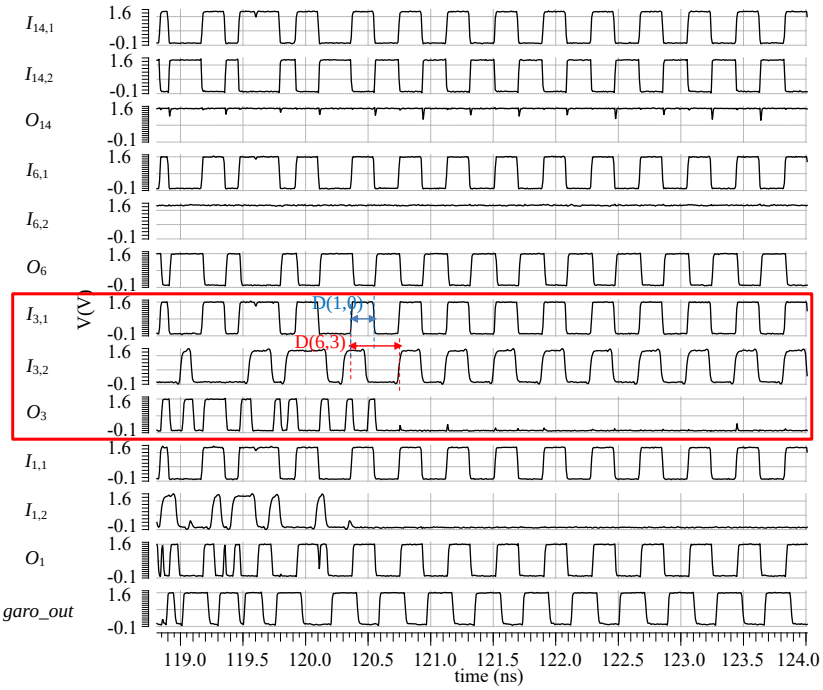


Figure 14: The simulated waveforms with feedback polynomial $f(x) = x^{15} + x^{14} + x^6 + x^3 + x + 1$.

50% as the duty cycle, which maintains the periodic toggles, fixed values and degeneration causing periodic oscillations.

The fixed value of one output $O_{r_{2i+1}}$ is determined by the parity of M_i and the number of inverter functions between $XOR_{r_{2i}}$ and $XOR_{r_{2i+1}}$ including the equivalent function of $XOR_{r_{2i}}$ for $i > 0$ or between INV_{r_0} and XOR_{r_1} including INV_{r0} for $i = 0$. For example, if the number of inverter functions between INV_{r_0} and XOR_{r_1} including INV_{r0} are odd and M_0 is even, $I_{r_1,2}$ is the inverse of $I_{r_1,1}$, and O_{r_1} will be fixed at 1. The fixed values determine whether the leftmost XOR gates in the degenerated circuits are equivalent to inverters or buffers.

In a word, only needing a period of periodic toggles with very short duration as a trigger where the period is $2DIV_\delta$ and duty cycle is 50%, such as the duration $4DIV_\delta$ even smaller than a sampling period, the outputs of specific XOR gates will be fixed and the GARO will degenerate to a classical ring oscillator causing periodic oscillations if Decision rule1 is met.

A feedback polynomial $f(x) = x^{15} + x^{14} + x^6 + x^3 + x + 1$ is applied to the gate-level model to verify Decision rule1, where $D(15, 14) < \delta$, $DIV_\delta = D(1, 0) = 3.7998/2 = 1.8999$ and $|2DIV_\delta - D(6, 3)| = |2 \times 1.8999 - (16.4995 - 8.8842)/2| = 0.00785 < \delta$. There are two values for M_i that $M_0 = 0$ and $M_1 = 2$. Thus, the sum S is even, $D(1, 0)/DIV_\delta$ is odd and $DIV_\delta > DIV_{min}$, which satisfies Decision rule1 and causes the periodic oscillation as shown in Figure 14. $M_0 = 0$ causes O_{14} and $I_{6,2}$ fixed leading to the degeneration according to Deduction rule. There is a short period of periodic toggles with the duration $4D(1, 0)$, the period $2D(1, 0)$ and the duty cycle 50% in *garo_out* as the trigger. The fixed value 1 of $I_{6,2}$ makes XOR_6 equivalent to an inverter and there is an even number of inverter functions between XOR_6 and XOR_3 including the equivalent inverter function of XOR_6. Thus, the two inputs $I_{3,1}$ and $I_{3,2}$ are the same with the delay difference $D(6, 3)$ and $M_1 = 2$ causes the almost simultaneous toggles of the two inputs leading to the fixed value 0 of O_3 as shown in the red frame. Under the effect of the fixed value 0 of $I_{1,2}$ derived

from O_3, XOR_1 is equivalent to a buffer and there is only one inverter function in the degenerated circuit, which is equivalent to a classical ring oscillator. The even value of S ensures the odd number of inverter functions in the degenerated circuit. Thus, the periodic toggles are maintained with the odd value of $D(1,0)/DIV_\delta$, which in turn maintains the fixed value of O_3 and degeneration causing the periodic oscillation. In this situation, the GARO degenerates to a classical ring oscillator, which has the periodic oscillation with $2D(1,0)$ as the period and 50% as the duty cycle. The small deviation between $D(6,3)$ and $2DIV_\delta$ causes glitches in O_3, which are filtered out as illustrated by the purer waveform of $I_{1,2}$ compared with O_3.

4.4.2 Decision rule2

Decision rule2. If $DIV_\delta > DIV_{min}$ and the sum S of all the values of M_i is odd, the GARO may oscillate periodically with the period determined by $I_{r_l,2}$.

Similar to Decision rule1, M_i will cause a period of fixed values of specific outputs once there is a short period of periodic toggles in *garo_out* with $2DIV_\delta$ as the period and 50% as the duty cycle. However, the odd value of S will cause even number of inverter functions in the circuit part corresponding to the feedback polynomial $f''(x) = x^{r_l} + 1$, under the effect of the fixed value of $I_{r_l,2}$ derived from O_{r_l-1}. Thus, the periodic toggles and fixed values cannot be maintained. However, the generation and destruction of these periodic toggles and fixed values will alternate causing another form of periodic oscillations with the period determined by the alternating cycles.

The feedback polynomial $f(x) = x^{15} + x^{14} + x^{13} + x^9 + x^3 + 1$ is taken as an example to verify the Decision rule2, where $D(15,14) < \delta$, $DIV_\delta = D(3,0) = 8.8842/2 = 4.4421$ and $|DIV_\delta - D(13,9)| = |4.4421 - (32.3499 - 23.4170)/2| = 0.02435 < \delta$. There are two values

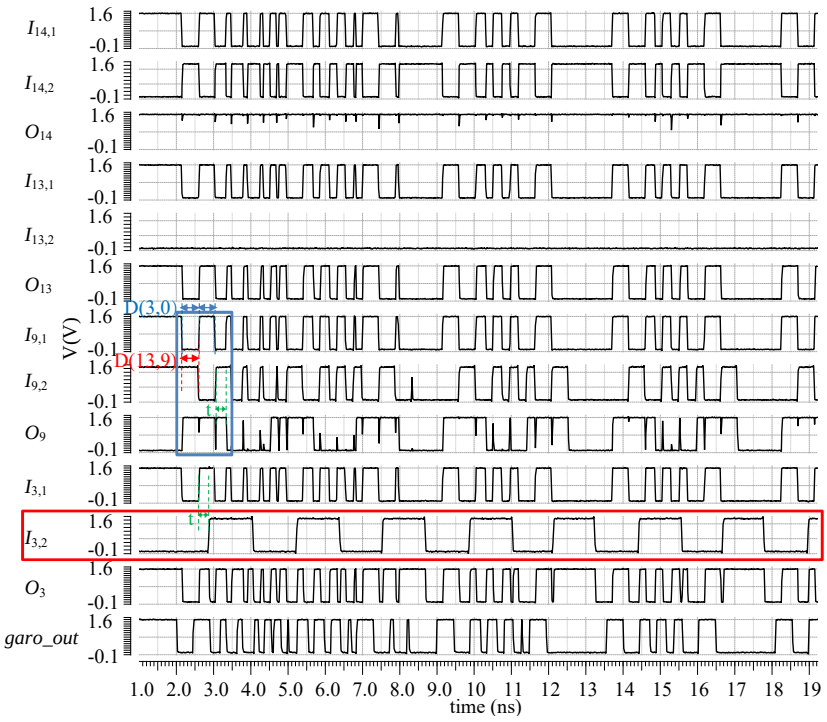


Figure 15: The simulated waveforms with feedback polynomial $f(x) = x^{15} + x^{14} + x^{13} + x^9 + x^3 + 1$.

of M_i that $M_0 = 0$ and $M_1 = 1$. Thus, the sum S is odd and $DIV_\delta > DIV_{min}$, which satisfies Decision rule2 and causes the periodic oscillation as shown in Figure 15, where $t = D(9,3) - D(3,0)$. $M_0 = 0$ causes O_{14} fixed leading to the degeneration according to Deduction rule. Under the trigger of a period of periodic toggles with the duration $2D(3,0)$, the period $2D(3,0)$ and the duty cycle 50%, $M_1 = 1$ causes the almost simultaneous toggles of $I_{9,1}$ and $I_{9,2}$ leading to a fixed value of O_9 as shown in the blue frame. Under the effect of the odd value of S, a period of fixed value 1 of $I_{3,2}$ derived from O_9 makes XOR_3 equivalent to an inverter causing an even number of inverter functions in the circuit part corresponding to the feedback polynomial $f''(x) = x^3 + 1$. Thus, the periodic toggles in the blue frame are not maintained. However, the periodic toggles and fixed value as in the blue frame are generated and destroyed alternately under the effect of the value 0 and 1 of $I_{3,2}$ respectively, which causes the periodic oscillation controlled by $I_{3,2}$. In this situation, the oscillation period of *garo_out* is twice that of $I_{3,2}$ with the value $4(D(9,3) + D(3,0))$.

4.4.3 Decision rule3

Decision rule3. If $S' + N'$ is even, $DIV'_\delta > DIV_{min}$, and $D(r_l, 0)/DIV'_\delta$ is odd, where S' is the sum of all the values of M'_i and N' is the number of sets of the delay differences satisfying situation 2 in the following two situations, the GARO may degenerate to GARO$'$ with the feedback polynomial $f'(x) = x^{r_l} + 1$ and get into periodic oscillation with 50% as the duty cycle and $2DIV'_\delta$ as the period.

- Condition 1 with an odd value of x_3, condition 2 with an even value of x_3, condition 3 with an even value of x_3 or condition 4 with an odd value of x_3.
- Condition 1 with an even value of x_3, condition 2 with an odd value of x_3, condition 3 with an odd value of x_3 or condition 4 with an even value of x_3.

In the above two situations, the conditions are shown as follows, where x_1, x_2 and x_3 are integers not less than 0 and $i = \{0, 1, ..., l/2 - 2\}$.

- $D(r_{2i}, r_{2i+1}) = 2x_1 DIV'_\delta + h$ where $0 < h < DIV'_\delta$, $|D(r_{2i+1}, r_{2i+2}) - x_2 DIV'_\delta| < \delta$, and $|D(r_{2i+2}, r_{2i+3}) - (x_3 DIV'_\delta + DIV'_\delta - h)| < \delta$.
- $D(r_{2i}, r_{2i+1}) = 2x_1 DIV'_\delta + h$ where $0 < h < DIV'_\delta$, $|D(r_{2i+1}, r_{2i+2}) - (x_2 DIV'_\delta + DIV'_\delta - h)| < \delta$ and $|D(r_{2i+2}, r_{2i+3}) - (x_3 DIV'_\delta + h)| < \delta$.
- $D(r_{2i}, r_{2i+1}) = 2x_1 DIV'_\delta + h$ where $DIV'_\delta < h < 2DIV'_\delta$, $|D(r_{2i+1}, r_{2i+2}) - x_2 DIV'_\delta| < \delta$ and $|D(r_{2i+2}, r_{2i+3}) - (x_3 DIV'_\delta + 2DIV'_\delta - h)| < \delta$.
- $D(r_{2i}, r_{2i+1}) = 2x_1 DIV'_\delta + h$ where $DIV'_\delta < h < 2DIV'_\delta$, $|D(r_{2i+1}, r_{2i+2}) - (x_2 DIV'_\delta + 2DIV'_\delta - h)| < \delta$ and $|D(r_{2i+2}, r_{2i+3}) - (x_3 DIV'_\delta + h - DIV'_\delta)| < \delta$.

Similar to Decision rule1, if the two inputs of $XOR_{r_{2i+1}}$ are the same or inverse with the delay difference $D(r_{2i}, r_{2i+1})$ and there is a short period of periodic toggles in *garo_out* with $2DIV'_\delta$ as the period and 50% as the duty cycle, a fixed value of $O_{r_{2i+3}}$ will be generated causing the GARO degenerate to GARO$'$ with the feedback polynomial $f'(x) = x^{r_{2i+4}} + x^{r_{2i+5}} + ... + 1$ once one situation is met. If GARO$'$ can oscillate periodically with $2DIV'_\delta$ as the period and 50% as the duty cycle, the periodic toggles, fixed values and degeneration can be maintained causing periodic oscillations.

If there is an odd number of inverter functions between $XOR_{r_{2i}}$ and $XOR_{r_{2i+3}}$ including the equivalent function of $XOR_{r_{2i}}$, and situation 1 is met, a fixed value 1 of $O_{r_{2i+3}}$ may be generated, which is similar to the effect of an even value of M_i in Decision rule1. Similarly, the situation 2 also has a similar effect with an odd value of M_i in Decision rule1. Thus, we can determine the periodicity condition of Decision rule3 with Decision rule1 as the reference. Once $S' + N'$ is even and under the trigger of a short period of periodic toggles

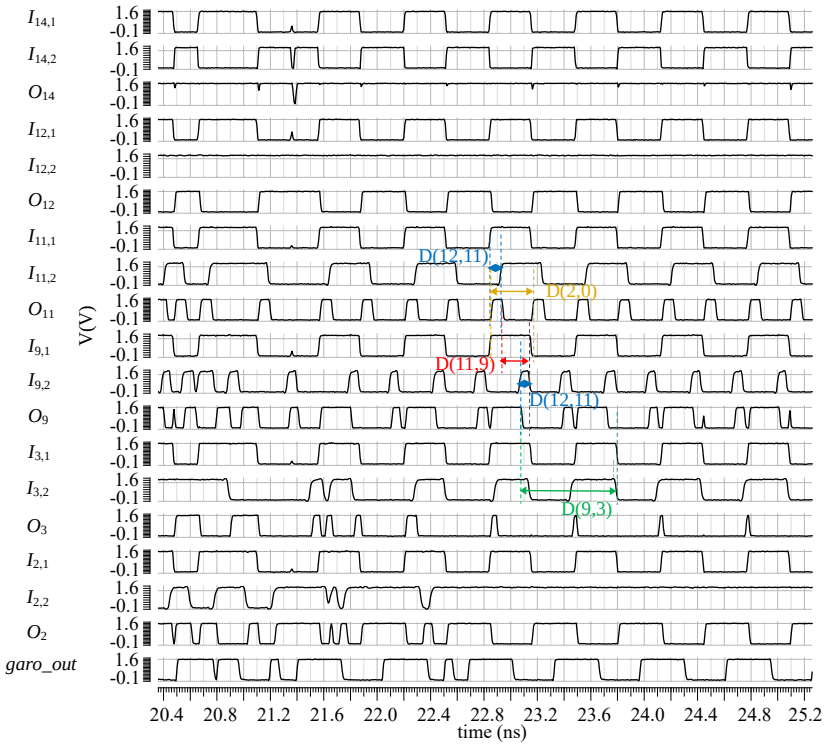


Figure 16: The simulated waveforms with feedback polynomial $f(x) = x^{15} + x^{14} + x^{12} + x^{11} + x^9 + x^3 + x^2 + 1$.

with $2DIV'_\delta$ as the period and 50% as the duty cycle, M'_i and the two situations will cause almost simultaneous toggles leading to the fixed values of the specific outputs and the fixed value of $I_{r_l,2}$ will cause an odd number of inverter functions in the degenerated GARO$''$ with the feedback polynomial $f''(x) = x^{r_l} + 1$. The GARO$''$ is equivalent to a classical ring oscillator and can oscillate periodically with 50% as the duty cycle and $2DIV'_\delta$ as the period, if $D(r_l, 0)/DIV'_\delta$ is odd. Thus, the periodic toggles, fixed values and degeneration can be maintained causing periodic oscillations.

A GARO with the feedback polynomial $f(x) = x^{15} + x^{14} + x^{12} + x^{11} + x^9 + x^3 + x^2 + 1$ is taken as an example to verify the Decision rule3. For this GARO, condition 2 in situation 1 is met, where $D(15, 14) < \delta$, $DIV'_\delta = D(2, 0) = 6.3834/2 = 3.1917$, $D(12, 11) = h = (29.7053 - 27.9767)/2 = 0.8643$, $0 < h < DIV'_\delta$, $|D(11, 9) - (DIV'_\delta - D(12, 11))| = |(27.9767 - 23.4170)/2 - (3.1917 - 0.8643)| = 0.04755 < \delta$, and $|D(9, 3) - (2DIV'_\delta + D(12, 11))| = |(23.4170 - 8.8842)/2 - (2 \times 3.1917 + 0.8643)| = 0.0187 < \delta$. Thus, $S' + N' = M'_0 + N' = 0 + 0 = 0$, $D(2, 0)/DIV'_\delta$ is odd and $DIV'_\delta > DIV_{min}$ satisfying Decision rule3, which causes the periodic oscillation as shown in Figure 16. Under the trigger of a short period of periodic toggles with $2D(2, 0)$ as the period and 50% as the duty cycle, the generated pulse signal O_{11} with $D(12, 11)$ as the width of high level makes the output O_9 equivalent to the input $I_{9,1}$ shifting to the left by $D(12, 11)$ under the effect of $|D(11, 9) - (D(2, 0) - D(12, 11))| < \delta$. Furthermore, $I_{3,1}$ is the same with $I_{9,1}$ derived from the feedback signal, and $|D(9, 3) - (2D(2, 0) + D(12, 11))| < \delta$ causes the almost simultaneous toggles of $I_{3,2}$ and $I_{3,1}$, resulting in the fixed value 0 of O_3. The fixed value 1 of $I_{2,2}$ derived from O_3 makes XOR_2 equivalent to an inverter causing an odd number of inverter functions in the degenerated circuit. Thus, the GARO degenerates to GARO$'$ with the feedback polynomial $f'(x) = x^2 + 1$ which is equivalent to a classical ring oscillator and can oscillate periodically with 50% as the duty cycle and $2DIV'_\delta$ as the period, under

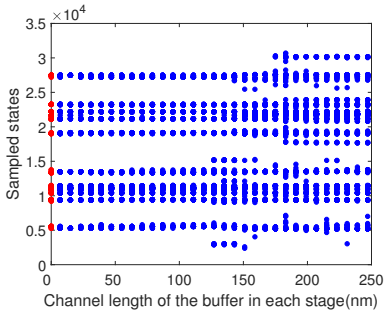
the effect of the odd value of $D(2,0)/DIV'_\delta$. Thus, the periodic toggles, fixed values and degeneration are maintained causing the periodic oscillation.

We note that the corresponding delay differences should satisfy the conditions described in Subsection 4.1 to be almost commenusurable to make sure the existence of corresponding DIV_δ and DIV'_δ for all the three decision rules.

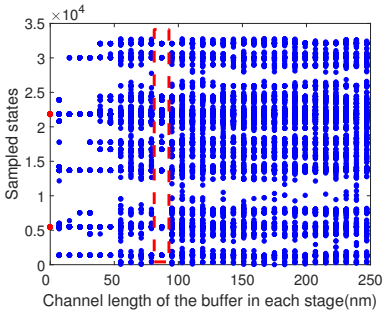
4.5 Bifurcation diagram

As described in Subsection 2.4, bifurcation diagram is an essential method to characterize the sensitivity of chaotic system to design parameter. With the delay in each stage of GARO as the design parameter and sampled states to represent system behavior, we plot the bifurcation diagrams corresponding to the feedback polynomials analysed in Subsection 4.3 and Subsection 4.4, which verify the high sensitivity and strong dependence of chaotic behavior to delay. The reflected decisive influence of delay on oscillatory behavior is consistent with the above cause analysis.

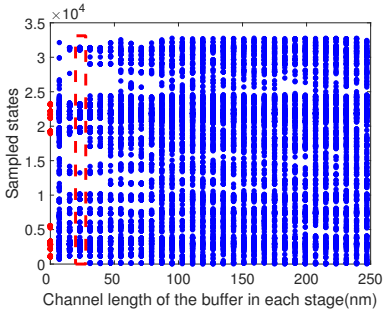
For every feedback polynomial, the corresponding GARO structure frames a specific nonlinear dynamic system. One feedback polynomial with different delay relationships can be regarded as another feedback polynomial with the same delays. For example, if the initial delays $D(5,4) = D(1,0)$ and $D(6,4) = 2D(1,0)$, and the altered delay $D'(5,4) = 2D'(1,0)$, the feedback polynomial $f(x) = x^{15} + x^{14} + x^5 + x^4 + x + 1$ will be equivalent to the



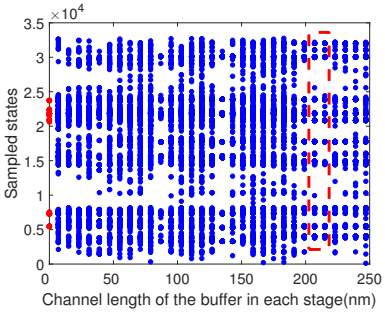
(a) The bifurcation diagram with feedback polynomial $f(x) = x^{15} + x^{14} + x + 1$.



(b) The bifurcation diagram with feedback polynomial $f(x) = x^{15}+x^{14}+x^6+x^3+x+1$.



(c) The bifurcation diagram with feedback polynomial $f(x) = x^{15} + x^{14} + x^{13} + x^9 + x^3 + 1$.



(d) The bifurcation diagram with feedback polynomial $f(x) = x^{15} + x^{14} + x^{12} + x^{11} + x^9 + x^3 + x^2 + 1$.

Figure 17: The simulated bifurcation diagrams corresponding to the example feedback polynomials in the rules, where the states without and with extra delays are marked with red and blue dots respectively, and the red dashed rectangles highlight the recurrence of periodic oscillations.

feedback polynomial $f'(x) = x^{15} + x^{14} + x^6 + x^4 + x + 1$ with $D(15, 14) < \delta$. Thus, we change the delay in each stage at the same time to the same extent to analyze the influence of delay on system behavior of one dynamic system. To make it easier to control the delay in each stage, we add a buffer in each stage to add extra delay and adjust its aspect ratio to change the delay simultaneously. The aspect ratio is controlled by changing the channel length with the width fixed, where a larger channel length represents a smaller aspect ratio and a larger delay.

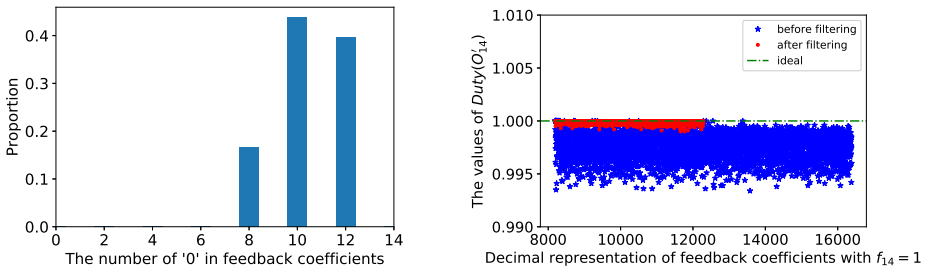
The bifurcation diagrams in Figure 17 are obtained using Candence virtuoso. With the channel length as the x-axis, each point represents a different sampled state in 4000 samples corresponding to a specific channel length or delay. The sampled states without and with extra delays from buffers are marked with red and blue colors respectively. We can distinguish the periodic and chaotic oscillations based on the distribution and the number of dots or the different sampled states in bifurcation diagrams as described in Subsection 2.4, where the sampled values tend to keep to a definite range with a small number of different samples for periodic oscillation and the sampled values spread throughout the y-axis of the bifurcation diagram with a large number of samples for chaotic oscillation.

For the feedback polynomial $f(x) = x^{15} + x^{14} + x + 1$ as shown in Figure 17(a), the smaller range of black region with chaotic property than the other three feedback polynomials in Figure 17(b),(c) and (d) represents higher possibility to get into periodic oscillation, due to the periodic condition that is easier to be satisfied. There are only two delay differences $D(15, 14)$ and $D(1, 0)$ to determine the oscillatory behavior and the circuit degenerates to a classical ring oscillator before $D(15, 14)$ is enlarged to δ. Once $D(15, 14)$ and $D(1, 0)$ is almost commensurable, the periodic oscillation may occur. On the contrary, for the other three feedback polynomials, the small variation of delay may cause periodic oscillation to revert to chaotic oscillation. Especially in Figure 17(c), the number of dots increases a lot after adding the extra delay $14ps$ corresponding to the channel length $10nm$. Thus, we should exclude the feedback polynomials where the periodic conditions are easily met, such as the feedback polynomials only with two delay differences to determine the oscillatory behavior. In other words, the FIFO-/GARO-based TRNGs with more decisive delay parameters determined by feedback polynomials have a larger black region with chaotic property, which are more reliable to be implemented as TRNGs. Besides, as the delay changes, the periodic condition may be satisfied again leading to the recurrence of periodic oscillation with a small number of sampled states as shown in the red dashed rectangles in Figure 17(b)(c)(d).

The alternation between periodic oscillation and chaotic oscillation with various delays illustrates the high sensitivity of system behavior to the delay parameter and that specific delay relationship is the cause of periodic oscillation for FIRO-/GARO-based TRNGs, which verifies our cause analysis above.

5 Verification

The full-state sampling results from the actual implemented circuit are consistent with the above analysis from simulation. We perform statistical analysis on the sampled states to further verify the gate-level implementation guidelines. The proportion of 1s (or 0s) in the sampled values of a specific state signal O'_x can reflect the duty cycle of corresponding state signal, which is denoted by $Duty(O'_x)$. Thus, the analysed fixed values in Deduction rule, Decision rule1 and Decision rule3 can be verified by the proportion $Duty(O'_x)$ with the ideal value 1 which represents that the corresponding state bit is constant. However, the approximate delay relationships may cause small number of unexpected toggles and glitches resulting in the non-ideal value of $Duty(O'_x)$. Thus, the values of $Duty(O'_x)$ close to 1 can verify the guidelines. Besides, there may be no fixed state bits and specific duty cycles in Decision rule2, thus Decision rule2 cannot be verified with $Duty(O'_x)$. We only



(a) The proportions of different number of 0s in feedback coefficients.

(b) The $Duty(O'_{14})$ in the sampled values.

Figure 18: The statistics of the feedback polynomials with measured periodic oscillations and the sampled values of O'_{14}.

conduct the verification for Deduction rule, Decision rule1 and Decision rule3.

For the feedback polynomials with periodic oscillations, there are more 0s than 1s in the 14 feedback coefficients as shown in Figure 18(a). Except for the only feedback polynomial $f(x) = x^{15} + 1$ with 14 zeros in the feedback coefficients which will definitely cause the periodic oscillation, there are 8, 10 or 12 zeros in the 14 feedback coefficients for all the feedback polynomials with periodic oscillations, where more 0s are in the feedback coefficients. A larger number of 1s in the feedback coefficients complicates the relationships among delay differences with more feedback paths, causing higher difficulty to satisfy the periodicity conditions and lower possibility of periodic oscillations. This situation is consistent with the observation in [SDH17].

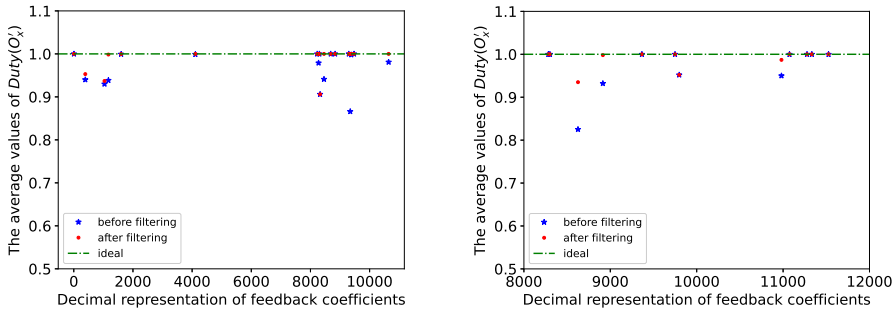
5.1 Verification of Deduction rule

We test all the 4096 feedback polynomials with $f_{14} = 1$ in the 8192 feedback polynomials to verify the Deduction rule. The values of $Duty(O'_{14})$ in the sampled values of O'_{14} for the 4096 feedback polynomials are shown in Figure 18(b), where the values before and after filtering are marked with blue and red colors respectively. For the feedback polynomials with $f_{14} = f_{13} = 1$, O'_{13} is the inverse of the feedback signal under the effect of the fixed value of O'_{14}, which is not the filtered signal of O'_{14}. Thus, there is no filtered signal of O'_{14} in the state bits. The $Duty(O'_{14})$ after filtering are obtained only for the feedback polynomials with $f_{14} = 1$ and $f_{13} = 0$ as shown in the left half of Figure 18(b).

When $f_{14} = 1$, all the values of $Duty(O'_{14})$ are close to 1, which illustrates that the small delay difference $D(15, 14) < \delta$ causes O'_{14} almost fixed at one value and will lead to the degeneration of the GARO verifying the correctness of Deduction rule. Besides, the larger values of $Duty(O'_{14})$ after filtering illustrate that the glitches generated from the small non-zero delay difference $D(15, 14)$ are filtered by the following gates and routes.

5.2 Verification of decision rules

For the 18 feedback polynomials meeting Decision rule1 and 12 feedback polynomials satisfying Decision rule3 with periodic oscillations, the average value of $Duty(O'_x)$ of all the specific state signal O'_x for a specific feedback polynomial is represented by a point in Figure 19, where the average values before and after filtering are marked by blue and red colors respectively. As shown in Figure 19(a) and Figure 19(b), almost all the values of $Duty(O'_x)$ are close to the ideal value 1 and the deviations between the actual values and ideal value are reduced after filtering, which verifies the correctness of Decision rule1 and Decision rule3.



(a) The average values of $Duty(O'_x)$ for Decision rule1.

(b) The average values of $Duty(O'_x)$ for Decision rule3.

Figure 19: The verification for Decision rule1 and Decision rule3 with measured data.

6 Design methodology and implementation

Based on the gate-level implementation guidelines, we propose a design methodology to implement a reliable TRNG based on GAROs, as shown in Figure 20. We first evaluate the feedback polynomials to select a proper feedback polynomial with chaotic behavior, where the feedback polynomials with possible periodic oscillations are filtered and the other feedback polynomials can be applied. We implement a Python script to realize the proposed algorithm to select a suitable feedback polynomial without periodic oscillation. The GARO with the feedback polynomial $f(x) = x^r + 1$ is equivalent to a classical ring oscillator, thus this feedback polynomial will definitely cause periodic oscillations without the requirement of evaluation. Besides, delay adjustment can eliminate the periodic oscillation caused by environmental conditions, aging, attacks and so on, to realize a robust FIRO-/GARO-based TRNG. The analysis of delay adjustment provides a further study of periodic oscillation and a better understanding of FIRO-/GARO-based TRNG. Also, a possible structure without periodic oscillations for all the feedback polynomials can be implemented by the delay adjustment, where the filtered feedback polynomials can be applied by eliminating the periodic oscillations with delay adjustment.

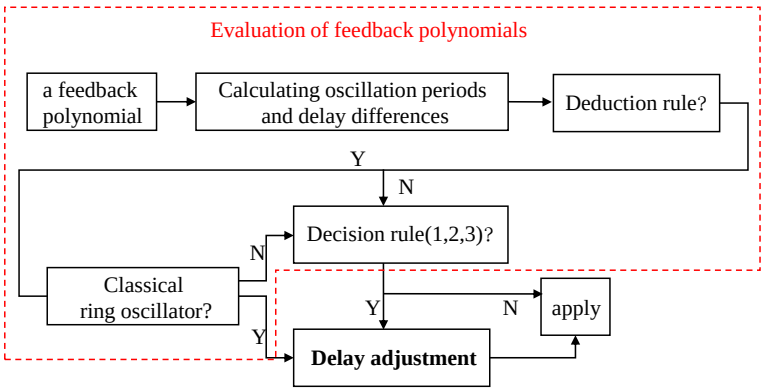


Figure 20: The design methodology.

6.1 Evaluation of feedback polynomials

The evaluation of feedback polynomials is to select a reliable configuration for a GARO-based TRNG with the steps as follows, which corresponds to the flow chart highlighted with the red frame in Figure 20.

- Implementing a GARO with $D(r_0, r_0 - 1) < \delta$.
- Applying the feedback polynomials $f(x) = x^{r_0} + x^{r_0-1} + x^m + 1$ to the GARO where $m = \{1, ..., r_0 - 2\}$, and utilizing an asynchronous counter to measure the oscillation periods T_m as shown with Equation 13.
- Calculating all the delay differences $D(n', n)$ for the evaluated feedback polynomial as shown with Equation 14.
- Applying Algorithm 2 for the evaluation, where the output *abandon* signal represents whether the GARO with the evaluated feedback polynomial has a high probability to get into periodic oscillations. The evaluated feedback polynomial should be abandoned with possible periodic oscillations if $abandon = 1$.

Algorithm 2 The evaluation of a feedback polynomial

Input: a feedback polynomial $f(x) = x^{r_0} + x^{r_1} + x^{r_2} + ... + x^{r_l} + 1$, and all the delay differences $D(r_0, r_1)$, $D(r_1, r_2)$,...,$D(r_l, 0)$.
Output: *abandon*

```
if Deduction rule is met then
    obtaining the degenerated feedback polynomial f'(x)
    if f'(x) = x^{r_l} + 1 then
        abandon = 1
    else if Decision rule1, Decision rule2 or Decision rule3 is met then
        abandon = 1
    else
        abandon = 0
    end if
else if Decision rule1, Decision rule2 or Decision rule3 is met then
    abandon = 1
else
    abandon = 0
end if
```

For the applicator who does not want to take a deep understanding of the implementation guidelines, we provide a simpler and straightforward way to select a proper feedback polynomial in a rapid hardware development. However, it leads to a confined selection range of feedback polynomials compared to the selection range using Algorithm 2. The simpler way is as follows, where x, x_1 and x_2 are natural numbers, $\delta = 0.38ns$ and $DIV_{min} = 0.79ns$ in our implementation.

- Excluding the feedback polynomials with $r_1 = r_0 - 1$, where there is a small delay difference $D(r_0, r_0 - 1)$ only from one inverter and corresponding routes. This exclusion can eliminate the degeneration of at least one stage as illustrated in Deduction rule.
- Excluding the feedback polynomials with $|D(r_0, r_1) - xDIV_\delta| < \delta$, where $DIV_\delta > DIV_{min}$ is a common measure between $D(r_0, r_1)$ and $D(r_l, 0)$. This can filter the feedback polynomials which may satisfy Decision rule1 and Decision rule2.

- Excluding the feedback polynomials with $|D(r_0, r_2) - x_1 DIV_\delta| < \delta$ and $|D(r_1, r_2) - x_2 DIV_\delta| < \delta$, where $DIV_\delta > DIV_{min}$ is a common measure among $D(r_0, r_2)$, $D(r_1, r_2)$ and $D(r_l, 0)$. This can prevent Decision rule3 from being satisfied.

6.2 Delay adjustment

Delay adjustment can eliminate periodic oscillations caused by the environmental conditions, aging, attack, and so on, for the applied feedback polynomial to improve robustness. Besides, the influence of delay adjustment on oscillatory behavior is analysed for a further study of periodic oscillation and better understanding of FIRO-/GARO-based TRNGs.

For a feedback polynomial $f(x) = x^{r_0} + x^{r_1} + x^{r_2} + x^{r_3} + ... + x^{r_l} + 1$, $D(r_0, r_1) \geq \delta$ should be met to prevent the degeneration in Deduction rule. For the decision rules, the periodic oscillations will not occur once one delay difference does not satisfy the periodicity conditions. Under the premise of $D(r_0, r_1) \geq \delta$, the principle of the delay adjustment for decision rules is as follows.

- For Decision rule1 and Decision rule2, if $D(r_0, r_1)$ and $D(r_l, 0)$ are not almost commensurable, the corresponding periodic oscillations will be eliminated.

- For Decision rule3, if $D(r_0, r_1)$ and $D(r_l, 0)$ are not almost commensurable and $D(r_0, r_1)$, $D(r_1, r_2)$, $D(r_2, r_3)$ do not meet the conditions in Decision rule3, the corresponding periodic oscillations will be eliminated.

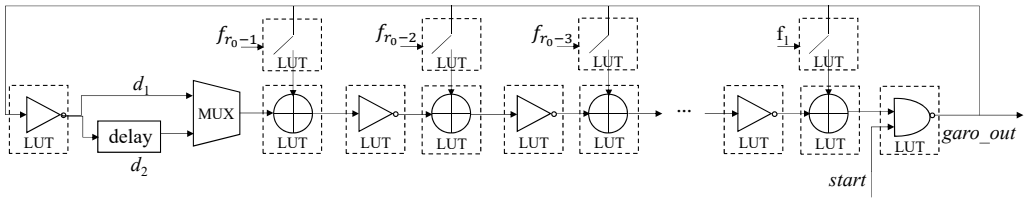
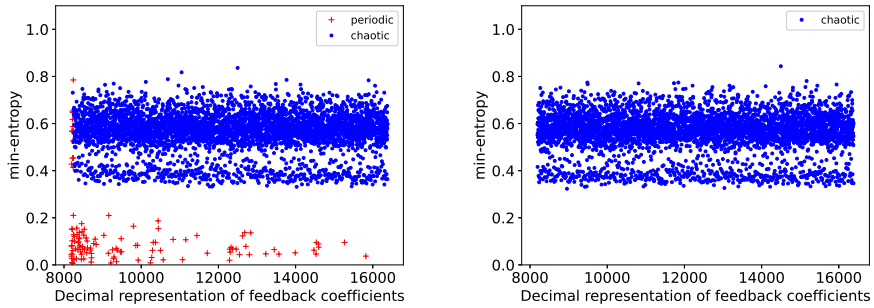


Figure 21: The implementation of the test circuit.

Based on the principles, the periodic oscillations can be eliminated for all the rules by only adjusting $D(r_0, r_1)$. To verify the effectiveness of delay adjustment, a multiplexer implemented with a MUX primitive is applied to switch between the two paths with original delay d_1 and the adjusted delay d_2 as shown in Figure 21. d_2 is adjusted by changing the route or adding extra buffers between INV_{15} and the multiplexer to obtain a suitable value of $D(r_0, r_1)$ to eliminate periodic oscillations. All the inverters, NAND gate, XOR gates, switches and buffers are implemented with LUTs.

We take a GARO with 15 cascaded inverters as an example for verification, where $D(15, 14)$ is adjusted for the feedback polynomials with $f_{14} = 1$. The entropy results and Lyapunov exponents before and after the adjustment are shown in Figure 22 and Figure 23 respectively. All the periodic oscillations are eliminated after adjusting $D(15, 14)$ with the probability of periodic oscillations reduced from 2.59% to 0, which verifies the effectiveness of the delay adjustment, and illustrates the correctness of the guidelines indirectly.

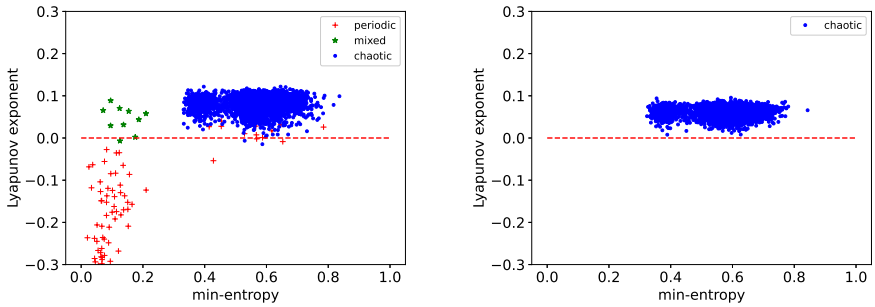
With the delay adjustment to eliminate periodic oscillations, no extra resource is consumed when $D(r_0, r_1)$ is adjusted by changing the route or the extra resource consumption is only several LUTs to implement the buffers when $D(r_0, r_1)$ is adjusted by adding buffers. Thus, the delay adjustment is lightweight.



(a) The entropy results with original $D(15, 14)$.

(b) The entropy results with adjusted $D(15, 14)$.

Figure 22: The measured entropy results for the GARO with $f_{14} = 1$ and 15 cascaded inverters.



(a) The Lyapunov exponents with original $D(15, 14)$.

(b) The Lyapunov exponents with adjusted $D(15, 14)$.

Figure 23: The measured Lyapunov exponents for the GARO with $f_{14} = 1$ and 15 cascaded inverters.

6.3 ASIC implementation

Except for the FPGA implementations for flexible analysis and verification, we also implement a GARO with 15 cascaded inverters and a full-state sampling circuit in ASIC for the evaluation of area, power consumption and frequency. The utilized technology is TSMC $28nm$, which is consistent with the technology of FPGA implementation and SPICE simulation. The area and frequency are evaluated with Design Compiler (Version P-2019.03), and power consumption results are obtained from PrimeTime PX (Version P-2019.03-SP3). The areas of the implemented GARO and sampling circuit are $80.3\mu m^2(160GE)$ and $58.3\mu m^2(116GE)$ respectively. The power consumptions are $0.063mW$ and $0.025mW$ respectively. Besides, the sampling circuit can work at a frequency higher than $2.8GHz$.

7 Online test

Online test module is an essential block for a TRNG design to ensure the correctness during the work of the TRNG, which is indispensable according to most TRNG standards. FIRO-/GARO-based TRNGs designed with the proposed methodology still have chance to enter the periodic oscillations due to the effect of active attacks and the delay inconstancy of implementations. Self-heating, temperature and voltage fluctuation, and the process variation among different silicon chips may cause the delay inconstancy. As long as the

changed delays satisfy the periodic conditions, the periodic oscillation will reappear. Thus, we design an online test module composed of a detector to detect periodic oscillations, which can improve the robustness by adjusting feedback polynomial or delays to eliminate detected periodic oscillations. In detail, a possible workaround to counteract environmental conditions can be:

- Once detecting a periodic oscillation, disable oscillation and stop TRNG output.
- Switch to another "good" polynomial or adjust corresponding delays, and restart oscillation for a while with detector enabled.
- If no periodic oscillation is detected, re-enable TRNG output, otherwise go to the second step.

The existing detector described in Subsection 3.2 needs large storage to store all the sampled states and calculates the number of different bit-patterns offline to detect periodic oscillations, which is not responsive with large resource consumption. Besides, although the probability of missed detection is reduced by the modified detector in Subsection 3.2, the existing detector has a high probability of false alarm, which will output an alarm signal for chaotic oscillations by mistake. For a GARO with the feedback polynomial $f(x) = x^r + x^{r-1} + x^{r-r'} + ... + 1$ and $D(r, r-1) < \delta$, the Deduction rule will cause all the outputs $O'_{r-1}, O'_{r-2}, ..., O'_{r-(r'-1)}$ fixed at 1s or 0s. Thus, the number of possible bit-patterns is reduced from 2^r to $2^{r-r'}$, which may be smaller than the threshold for detection causing false alarm even if the degenerated circuit oscillates chaotically. Taking the feedback polynomial $f(x) = x^{15} + x^{14} + x^7 + ... + 1$ as an example, almost all the sampled values of $O'_{14}O'_{13}O'_{12}O'_{11}O'_{10}O'_9O'_8$ are 0101010 under the effect of $D(15, 14) < \delta$, which makes the number of possible bit-patterns reduced from 2^{15} to 2^7. Thus, it is necessary to design a lightweight and responsive detector with a high detection rate to realize online testing of FIRO-/GARO-based TRNGs.

The design principle of our online test module is that the number of toggles of the output *garo_out* or $firo\_out$ in a certain time will tend to several certain values due to the periodicity of the output signal in periodic oscillations, while the number of toggles in a certain time is varied for chaotic oscillations. Thus, the variances of the number of toggles for periodic oscillation will be smaller than the variances for chaotic oscillation. Based on the difference in variances for periodic and chaotic oscillations, a lightweight and responsive online test module or detector with a high detection rate is designed.

7.1 Architecture of detector

The detector consists of an asynchronous counter to measure the number of toggles of the output *garo_out* or $firo\_out$ in a certain time, and an ALU to calculate and compare the variances of the number of toggles represented by the counter values as shown in Figure 24. The architecture of asynchronous counter is shown in Figure 11. The asynchronous counter is disabled by the first TFF after each count in one clock cycle for the counter values to be stable excluding the influence of the delays in asynchronous counter on counter values. After disabling for two clock cycles, the counter is reset for the count enabled in the next clock cycle in our tests. Thus, the asynchronous counter counts the toggles every four clock cycles with the enable in one clock cycle. An alarm signal representing whether a periodic oscillation occurs is generated by comparing the variances with a specific threshold. If the calculated variance is smaller than the threshold, a positive pulse is generated in the alarm signal representing the occurrence of periodic oscillation.

In the detector, only the output *garo_out* or $firo\_out$ is used to detect periodic oscillations without the need to save all the sampled bit-patterns, which makes our detector more lightweight than the detector in [Dic15]. Our detector only consumes 149 registers

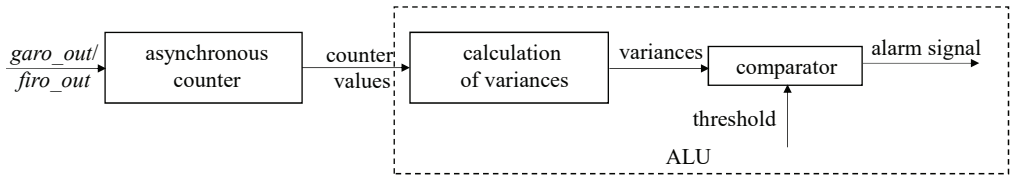


Figure 24: The architecture of detector.

for storage. To further reduce resource consumption, the division operation in variance calculation is implemented by shift operation, thus the window size utilized to calculate variances should be a power of 2. Besides, the variance V calculated and compared in a window with the specific size W is W times of actual variance as shown in Equation 15, which decreases one shift operation with less resource consumption and higher precise due to the reduction of the accuracy loss caused by a shift operation. In Equation 15, c_i is the i-th counter value in a window with the size W. Except for the advantage in resource consumption, our detector design is responsive that it can detect periodic oscillations during the work of FIRO-/GARO-based TRNGs with a latency in the range of $[W, 2W)$, while the existing detector in [Dic15] is not a responsive and online detector.

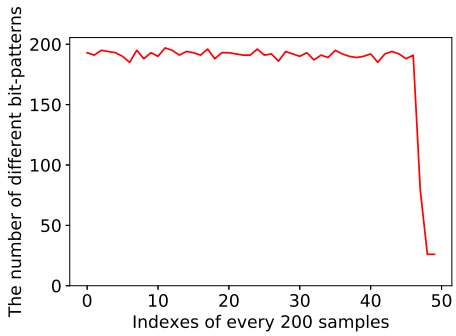
$$V = (E(c_i^2) - E^2(c_i)) \times W = \sum_{i=1}^{W} c_i^2 - \frac{(\sum_{i=1}^{W} c_i)^2}{W} \tag{15}$$

7.2 Parameter design

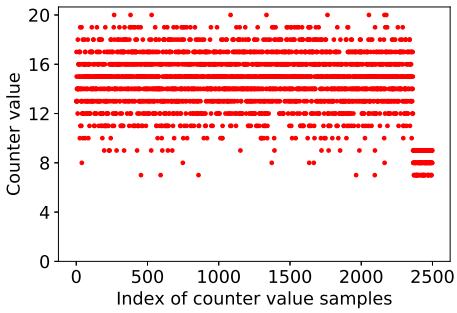
To implement the detector, there are two parameters to be determined, including the window size for variance calculation and the threshold for differentiation between chaotic and periodic oscillations.

The largest window size is determined by the duration of periodic oscillation that the window size should not be larger than the number of counter values corresponding to the duration of periodic oscillation. Otherwise, all the counter values corresponding to periodic oscillation will be mixed with the counter values corresponding to chaotic oscillation in one window, and the large change around the boundary between periodic oscillation and chaotic oscillation will make the variance in the window large, even larger than the variances of chaotic oscillation. Thus, there is no small variance derived from periodic oscillation for detection causing missed detection. The duration of periodic oscillation can be determined by the modified detection method in Subsection 3.2. Taking a GARO with the feedback polynomial $f(x) = x^{15} + x^{14} + x^{11} + x^6 + x^2 + 1$ as an example, the counter values tend to certain values when the GARO gets into periodic oscillation as shown in Figure 25. The tendency will result in reduced variances, which verifies the feasibility of our detector design. According to Figure 25(a), the duration of periodic oscillation is about $200 \times 2 = 400$ sampling periods, and the corresponding number of counter values is about $400/4 = 100$ as shown in Figure 25(b). Thus, the window size should not be larger than 100. Otherwise, the variance corresponding to periodic oscillation will be larger than the variances corresponding to chaotic oscillation causing missed detection as shown in Figure 26(a), where the window size is 128. Thus, the window size for the feedback polynomial $f(x) = x^{15} + x^{14} + x^{11} + x^6 + x^2 + 1$ should be smaller than 128.

The smallest window size is determined by chaotic oscillation. If the window size is too small, the variances corresponding to chaotic oscillation may be close to or even smaller than the variances corresponding to periodic oscillation, which makes the variance

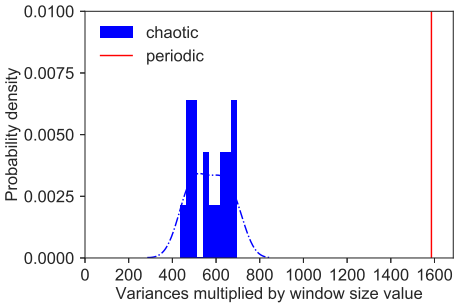


(a) The number of different bit-patterns in every 200 samples.

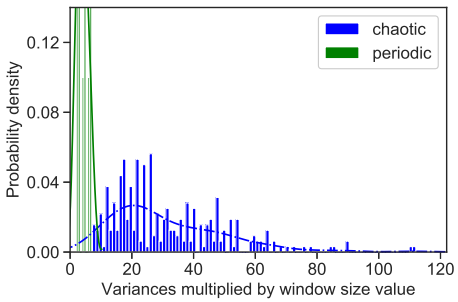


(b) The counter values in one clock cycle.

Figure 25: The measured sampling results of bit-patterns and the counter values.



(a) The variance distribution with $W = 128$.

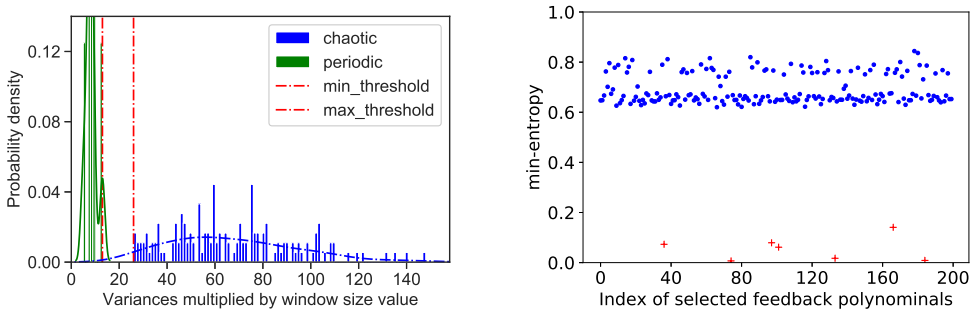


(b) The variance distributions with $W = 8$.

Figure 26: The variance distributions with the window size $W = 128$ and $W = 8$.

distributions of periodic oscillation and chaotic oscillation overlap with each other. Thus, the periodic oscillation can not be distinguished from chaotic oscillation with a threshold, causing missed detection or false alarm. Also taking the feedback polynomial $f(x) = x^{15} + x^{14} + x^{11} + x^6 + x^2 + 1$ as an example, the variance distribution of periodic oscillation overlaps with the variance distribution of chaotic oscillation as shown in Figure 26(b) when the window size is 8, which will cause chaotic oscillation and periodic oscillation inseparable leading to missed detection or false alarm. Thus, the window size for the feedback polynomial $f(x) = x^{15} + x^{14} + x^{11} + x^6 + x^2 + 1$ should be larger than 8.

After determining the suitable range of window size, the smallest window size in the range is adopted for the detector to decrease the reaction time. Once determining the window size, we can determine the range of threshold which should be smaller than the smallest variance of chaotic oscillation and larger than the largest variance of periodic oscillation. The median in the range of threshold is adopted for detection to reduce the influence of noise on the detection accuracy. For example, the window size for the feedback polynomial $f(x) = x^{15} + x^{14} + x^{11} + x^6 + x^2 + 1$ should be smaller than 128 and larger than 8, thus we select 16 as the window size. The corresponding variance distributions with window size 16 are shown in Figure 27(a), where the smallest variance of chaotic oscillation is 26 and the largest variance of periodic oscillation is 13. Thus, the optional range of threshold is from 13 to 26. We can adopt the median 19 or 20 in the range as the threshold.



(a) The variance distribution with $W = 16$.

(b) The test results of detector.

Figure 27: The variance distribution with $W = 16$ and the test results of the proposed detector.

7.3 The implementation results

7.3.1 FPGA implementation

Based on the above parameter design method, we can design not only a detector for one feedback polynomial, but also a common detector whose parameters are suitable for most of the feedback polynomials, even for all the feedback polynomials. To verify the effectiveness of our detector design, we remove the full-state sampling circuit assisting the detector design, and adopt 64 as the window size W and 67 as the threshold to conduct online test for a GARO with 15 cascaded inverters on Xilinx Artix-7 FPGA. We select 200 feedback polynomials randomly applied for the test and the results are shown in Figure 27(b), where the min-entropy results for the detected periodic oscillations are highlighted with red color. The test results show that all the periodic oscillations with reduced entropy are correctly detected by the detector, which verifies the high detection rate of our detector design.

In addition to the high detection rate, our detector design is more responsive than the existing detectors, such as the detector described in Subsection 3.2, repetition count test and total failure test. The latency of our detector design is determined by the window size W. For example, the latency is in the range of $[16, 32)$ sampling periods for the feedback polynomial $f(x) = x^{15} + x^{14} + x^{11} + x^6 + x^2 + 1$ with $W = 16$ as described in Subsection 7.2. For the common detector with $W = 64$, the latency is in the range of $[64, 128)$ sampling periods. However, the detector described in Subsection 3.2 can only detect periodic oscillations after finishing sampling. Statistical tests such as repetition count typically require a lot more than 128 samples to reliably detect faulty behavior without raising false alarms [sta] and can only detect catastrophic failures. In addition to the high detection rate and low latency, the implemented detector only consumes 241 LUTs, 149 registers and one DSP, which is lightweight compared with the implementation of statistical tests in [YRM+15] and the detector with a large number of registers or large memory to store all the samples in [Dic15]. The FPGA implementation results illustrate the effectiveness of our detector design.

7.3.2 ASIC implementation

Except for the FPGA implementation to verify the effectiveness of the proposed detector, the same detector circuit is also implemented in ASIC with the same tools and technology as explained in Subsection 6.3 to evaluate area, power consumption and frequency. The area and power consumption of the implemented detector are $2033.6\mu m^2(4035GE)$ and $0.042mW$, which is more lightweight than the ASIC implementations of statistical tests in [YRM+15]. Besides, the detector can work at a frequency higher than $2.8GHz$.

8 Conclusion

Periodic oscillations will compromise FIRO-/GARO-based TRNGs. We take GARO as the example to analyse periodic oscillations which is similar for FIROs. The influence clarification, intrinsic reason analysis and corresponding solutions for periodic oscillations missing in existing researches are realized in this paper. We are the first to use Lyapunov exponent and bifurcation diagram to analyse the chaotic property of FIRO-/GARO-based TRNGs.

The influence of periodic oscillations on FIRO-/GARO-based TRNGs is quantified using NIST SP 800-90B test suite and Lyapunov exponents. Most periodic oscillations of FIRO-/GARO-based TRNGs will cause entropy loss and negative Lyapunov exponents with a lack of chaos. The negative Lyapunov exponents and entropy loss of periodic oscillations compared with chaotic oscillations reflect that periodic oscillation is a main failure of FIRO-/GARO-based TRNGs.

To analyse the intrinsic reasons for periodic oscillations based on the measured results from FPGA implementations, a gate-level model is built and simulated with SPICE, using gate-level implementations and TSMC $28nm$ as the technology. The simulated technology is consistent with the technology of utilized FPGAs. Based on the model, several gate-level implementation guidelines are proposed, including one deduction rule and three decision rules. The guidelines demonstrate the regularity observed from real FPGA implementations and reveal the relationship between delay differences and periodic oscillations. The corresponding bifurcation diagrams also illustrate a strong link between delays and oscillatory behaviors. The gate-level implementation guidelines are verified using the full-state sampling.

Based on the gate-level implementation guidelines, we propose a design methodology to implement a reliable TRNG based on GAROs, including:

1. the evaluation of feedback polynomials to select a suitable feedback polynomial without periodic oscillation, where the proposed algorithm is realized using a Python script.

2. the delay adjustment to eliminate periodic oscillations caused by environmental conditions, aging, attacks and so on, to build a robust TRNG.

The delay adjustment is lightweight and its effectiveness is verified through the comparisons of entropy and Lyapunov exponents before and after the adjustment in experiments.

An online test module as an essential block of TRNGs is proposed to improve the robustness of FIRO-/GARO-based TRNGs against potential active attacks and the delay inconstancy of implementations, by adjusting feedback polynomial or delays to eliminate the detected periodic oscillations. The online test module is composed of a detector to detect periodic oscillations based on the regularity of the outputs of GAROs. The detector has much better performances than the existing detector and statistical tests, which is lightweight and responsive with high detection accuracy.

Except for the FPGA implementations for flexible analysis and verification, we also implement a GARO, the full-state sampling circuit and the detector in ASIC to evaluate the areas, power consumptions and frequencies, with TSMC $28nm$ as the technology. The areas evaluated under Design Compiler for the implemented GARO, sampling circuit and detector are $80.3\mu m^2(160GE)$, $58.3\mu m^2(116GE)$ and $2033.6\mu m^2(4035GE)$ respectively. The power consumptions obtained by PrimeTime PX are $0.063mW$, $0.025mW$ and $0.042mW$ respectively. Besides, the sampling circuit and detector can work at a frequency higher than $2.8GHz$.

Our research has deepened the understanding of FIRO-/GARO-based TRNGs, and the implementation guidelines derived from the analysis of gate-level model advance the establishment of stochastic model in the future.

Future work will concern the establishment of stochastic model for FIRO-/GARO-based TRNGs and the evaluation of the FIGARO-based TRNGs composed of the improved FIROs and GAROs with our solutions.

Acknowledgements

This work is supported in part by the National Key R&D Program of China (Grant No. 2018YFB2202101), and in part by the National Natural Science Foundation of China (Grant No. 62104129, No. 62274102). The authors thank the editors and reviewers for their thoughtful comments. Vladimir Rožić contributed to this paper during his research visit to Tsinghua University July-December 2019. The authors also thank Shuying Yin and Wei Guo, who are coworkers at Tsinghua University and Wuxi Micro Innovation Integrated Circuit Design Co., Ltd., for troubleshooting with the use of ASIC design tools and SPICE simulation tools.

References

[CD13] Ihsan Cicek and Günhan Dündar. A chaos based integrated jitter booster circuit for true random number generators. In *21st European Conference on Circuit Theory and Design, ECCTD 2013, Dresden, Germany, September 8-12, 2013*, pages 1–4. IEEE, 2013.

[DG07] Markus Dichtl and Jovan Dj. Golic. High-speed true random number generation with logic gates only. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 45–62. Springer, 2007.

[Dic03] Markus Dichtl. How to predict the output of a hardware random number generator. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 181–188. Springer, 2003.

[Dic15] Markus Dichtl. Fibonacci ring oscillators as true random number generators - A security risk. *IACR Cryptol. ePrint Arch.*, 2015:270, 2015.

[DSFC04] Milos Drutarovský, Martin Simka, Viktor Fischer, and Frederic Celle. A simple pll-based true random number generator for embedded digital systems. *Comput. Artif. Intell.*, 23(5):501–515, 2004.

[FLL+18] Lingyan Fan, Yongping Long, Jianjun Luo, Liangliang Zhu, and Hailuan Liu. A true random number generator based on meta-stable state. *IEICE Electron. Express*, 15(1):20171122, 2018.

[Gol06] J.D.J. Golic. New methods for digital generation and postprocessing of random data. *Computers, IEEE Transactions on*, 55:1217– 1229, 11 2006.

[HKVC22a] Serhii Haliuk, Oleh Krulikovskyi, Dmytro Vovchuk, and Fernando Corinto. Memristive structure-based chaotic system for PRNG. *Symmetry*, 14(1):68, 2022.

[HKVC22b] Serhii Haliuk, Oleh Krulikovskyi, Dmytro Vovchuk, and Fernando Corinto. Memristive structure-based chaotic system for PRNG. *Symmetry*, 14(1):68, 2022.

[HM15] Timothy Hall and Kerry McKay. User's guide to running the draft nist sp 800-90b section 9 entropy estimation tests, 2015-03-10 2015.

[KR11] B. Kliková and Ales Raidl. Reconstruction of phase space of dynamical systems using method of time delay. 2011.

[KS11] W. Killmann and W. Schindler. *A proposal for: Functionality classes for random number generators, version 2.0.* Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn, September 2011. https://www.bsi.bund.de/EN/Home/home node.html.

[Lam18] Dragan Lambic. Security analysis and improvement of the pseudo-random number generator based on quantum chaotic map. *Nonlinear Dynamics*, 94, 10 2018.

[LJZ19] Xinning Liu, Song Jia, and Hanzun Zhang. A novel high-speed fpga-based true random number generator based on chaotic ring oscillator. In *13th IEEE International Conference on ASIC, ASICON 2019, Chongqing, China, October 29 - November 1, 2019*, pages 1–4. IEEE, 2019.

[Meh19] Sina Mehdizadeh. A robust method to estimate the largest lyapunov exponent of noisy signals: A revision to the rosenstein's algorithm. *Journal of Biomechanics*, 85:84–91, 2019.

[RCD93] Michael T. Rosenstein, James J. Collins, and Carlo J. De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1):117–134, 1993.

[Roz16] Vladimir Rozic. *Circuit-Level Optimizations for Cryptography; True Random Number Generators: Design, Evaluation and Testing.* PhD thesis, KU Leuven, 2016.

[SDH17] Martin Schramm, Reiner Dojen, and Michael Heigl. Experimental assessment of FIRO- and GARO-based noise sources for digital TRNG designs on FPGAs. pages 1–6, 09 2017.

[SMS07] Berk Sunar, William J. Martin, and Douglas R. Stinson. A provably secure true random number generator with built-in tolerance to active attacks. *IEEE Trans. Computers*, 56(1):109–119, 2007.

[SSS87] Shinichi Sato, Masaki Sano, and Yasuji Sawada. Practical methods of measuring the generalized dimension and the largest lyapunov exponent in high dimensional chaotic systems. *Progress of theoretical physics*, 77(1):1–5, 1987.

[sta] random: implement the "repetition count" nist sp800-90b health test. https://lkml.org/lkml/2020/9/21/181.

[Sta14] John A. Stankovic. Research directions for the internet of things. *IEEE Internet Things J.*, 1(1):3–9, 2014.

[TBK+18] Meltem Sonmez Turan, Elaine B. Barker, John M. Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. Recommendation for the entropy sources used for random bit generation. *Special Publication (NIST SP) - 800-90B*, 2018.

[VCC09] Angelo Vulpiani, Fabio Cecconi, and Massimo Cencini. *Chaos: from simple models to complex systems*, volume 17. World Scientific, 2009.

[VD10] Michal Varchola and Milos Drutarovský. New high entropy element for FPGA based true random number generators. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2010.

[WYZL20] Ping Wang, Jinjiang Yang, Min Zhu, and Bo Liu. A high-throughput low-area entropy source based on asynchronous feedback unit. In *ICCSP 2020: 4th International Conference on Cryptography, Security and Privacy, Nanjing, China, January 10-12, 2020*, pages 131–135. ACM, 2020.

[Yan18] Bohan Yang. *True Random Number Generators for FPGAs*. PhD thesis, KU Leuven, 2018.

[YJW+17] Yunfan Yang, Song Jia, Yuan Wang, Shaonan Zhang, and Chao Liu. A reliable true random number generator based on novel chaotic ring oscillator. In *IEEE International Symposium on Circuits and Systems, ISCAS 2017, Baltimore, MD, USA, May 28-31, 2017*, pages 1–4. IEEE, 2017.

[YRG+18] Bohan Yang, Vladimir Rozic, Milos Grujic, Nele Mentens, and Ingrid Verbauwhede. ES-TRNG: A high-throughput, low-area true random number generator based on edge sampling. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):267–292, 2018.

[YRM+15] Bohan Yang, Vladimir Rozic, Nele Mentens, Wim Dehaene, and Ingrid Verbauwhede. Embedded HW/SW platform for on-the-fly testing of true random number generators. In Wolfgang Nebel and David Atienza, editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 345–350. ACM, 2015.

[YSV04a] Mustak Yalcin, Johan Suykens, and Joos Vandewalle. True random bit generation from a double-scroll attractor. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51:1395 – 1404, 08 2004.

[YSV04b] Mustak Yalcin, Johan Suykens, and Joos Vandewalle. True random bit generation from a double-scroll attractor. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51:1395 – 1404, 08 2004.