



**HAL**  
open science

# Beyond Total Locking: Demonstrating and Measuring Mutual Influence on a RO-Based True Random Number Generator on an FPGA

Eloïse Delolme, Viktor Fischer, Florent Bernard, Nathalie Bochard, Maxime Pelcat

► **To cite this version:**

Eloïse Delolme, Viktor Fischer, Florent Bernard, Nathalie Bochard, Maxime Pelcat. Beyond Total Locking: Demonstrating and Measuring Mutual Influence on a RO-Based True Random Number Generator on an FPGA. 37th IEEE International System-on-Chip Conference, Sep 2024, Dresden, Germany. ujm-04649086

**HAL Id: ujm-04649086**

**<https://ujm.hal.science/ujm-04649086v1>**

Submitted on 16 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Beyond Total Locking: Demonstrating and Measuring Mutual Influence on a RO-Based True Random Number Generator on an FPGA

Eloïse Delolme<sup>1</sup>, Viktor Fischer<sup>1</sup>, Florent Bernard<sup>1</sup>, Nathalie Bochard<sup>1</sup>, Maxime Pelcat<sup>2</sup>

<sup>1</sup> Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School  
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France  
{eloise.delolme, fischer, florent.bernard, nathalie.bochard}@univ-st-etienne.fr

<sup>2</sup> Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, F-35000 Rennes, France  
maxime.pelcat@insa-rennes.fr

**Abstract**—Ring oscillator-based true random number generators are of interest because of their well-known and well-characterised conversion of analog noise into random numbers. The main drawback of using ring oscillators as a source of randomness is their tendency to be influenced by their environment. In particular, oscillators may lock to another signal at a frequency close to the nominal frequency of the ring, or two or more rings may even lock to each other. This is particularly dangerous for generators with multiple rings, which require the rings to be independent of each other. Furthermore, to reduce the risk of manipulable global noise sources, the rings should have the same structure and topology, making them even more vulnerable to locking. The metrics commonly used to quantify the degree of locking have limitations that can lead to erroneous conclusions as to whether a ring is locked or not. This is why we prefer to use the term *mutual influence*. In this paper, we propose a clear definition of the mutual influence between ring oscillators used as sources of randomness. One of the advantages of this definition is that it can easily be extended to include the case of the total locking of rings. Based on this definition, we introduce a new metric to quantify the mutual influence, which evaluates a statistical distance between the current distribution of phase differences and uniform distribution. The experimental results of several FPGA implementations of ring oscillators highlighted the suitability of the Kolmogorov-Smirnov test as a metric for detecting mutual influence.

**Index Terms**—True Random Number Generator, Ring oscillator, Locking phenomenon, Mutual influence, Kolmogorov-Smirnov test

## I. INTRODUCTION

True random number generators (TRNGs) are essential cryptographic primitives that are used to generate confidential keys, nonces (numbers used once), padding values and more recently random masks in countermeasures against side-channel attacks. In all these cases, the generator's past and future output values must not be deducible from its current values. The quality of a TRNG cannot be assessed using generic, black box, statistical tests such as NIST SP 800-22 [1]. Modern certification schemes [2] require a stochastic model of the generator whose role is to compute an entropy rate per bit (or at least its lower bound). Ideally, this rate should be as close as possible to 1 to ensure cryptographic use of the generator.

In logic devices such as FPGAs and ASICs, many random number generators are based on free-running oscillators because they are easy to implement and have also proven to be good sources of entropy [3], [4], [5]. Due to the analog noise present in the device, the period of the generated clock signal is unstable and can be observed as timing jitter in the time domain. The authors of [4] describe the elementary ring oscillator (RO)-based TRNG. Since the jitter of the clock signal generated in a ring is usually very small, it has to be accumulated over a long period of time, which significantly reduces the TRNG output bitrate. To overcome this problem, a solution based on multiple ROs (MURO-TRNG) has been proposed [6]. The principle is to sample several ROs with the same reference clock and to XOR the samples to collect more entropy in a shorter time, thereby improving the bitrate. The stochastic model of this generator is more complex and is based on a stronger assumption of the independence of the oscillators. In addition, ROs must have the same topology and be placed close to one another to reduce the impact of manipulable global noise sources [7].

As a consequence, this raises the question of the mutual influence between the oscillators in a TRNG. Mutual influence can be passive: influence between rings, or with the surrounding logic. It can also be active: electromagnetic attacks [8], fault injection [9]. Whatever the case, this influence can have an impact on the operating frequencies of the ROs and consequently on the entropy harvesting mechanism. Furthermore, the assumption of RO independence made at the model level is questionable and may lead to overestimation of the entropy rate of the generator. Worse still, too much influence between the ROs, known as *total locking*, can result in a stable generator output value. This worst case is very easy to detect as the entropy disappears, but the same is not true of mutual influence, which can be partial and/or transient. Mutual influence therefore has a negative impact on the security of the generator and must be detected.

In [10], Mureddu *et al.* study the effect of locking phenomena on several types of oscillating rings and propose four metrics to measure locking. Two of the metrics are based on

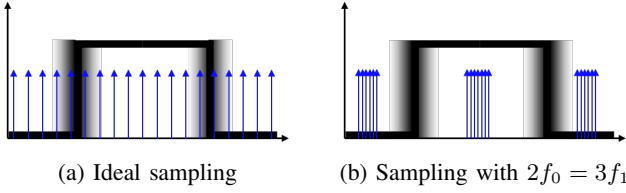


Fig. 1: The representation of sampling events in blue over a nominal period in black.

the signal period, while the other two use the phase shift of the oscillator output signal. When two signals are locked, they have a constant phase difference. One metric is to consider the standard deviation of the phase differences. However, even though the proposed metrics, especially the standard deviation of phase differences, can detect cases of total locking they can not precisely evaluate mutual influence.

This article provides a more precise definition of the mutual influence between ROs in the context of true random number generation, together with a new metric for measuring it. This definition presented in Section II encompasses and generalises the notions of locking described in literature. The proposed metric is tested on four hardware configurations ranging from the one that favours mutual influence the most to one that prevents it the most, as described in Section III. In Section IV, we provide experimental proofs demonstrating that the metrics used in the state-of-the-art are unable to detect some existing mutual influence as accurately as our proposal.

## II. FROM LOCKING TO MUTUAL INFLUENCE

The worst case of influence, total locking, is already reported in the state-of-the-art. However, between the ideal case in which the generator maintains a very high entropy per bit rate and the worst case in which the rings are totally locked, there is a whole range of mutual influence which represents a risk for the generator.

### A. Definition of mutual influence

To guarantee correct operation of a multiple ring oscillators (MURO) TRNG, no entropy drop, even a short one, should occur. If there is a risk to the security of the TRNG, the system should detect it and emit a warning. Let  $s_0$  be the output signal of the sampling oscillator  $RO_0$  of frequency  $f_0$  and  $s_1$  the output signal of the sampled oscillator  $RO_1$  of frequency  $f_1$ . To achieve a high level of security, i.e. when the entropy per bit rate should be very close to 1, the sampling of  $RO_1$  over several periods of  $RO_0$  must cover the nominal period as illustrated in Figure 1a. This can be characterised using phase differences. Indeed, a phase difference value is associated with each sample. Thus, in the ideal case, phase differences should be uniformly distributed over the nominal period.

However, one possible cause of a drop in entropy is a loss of uniformity in the distribution of phase differences. This loss of uniformity leads to the creation of clusters during sampling. The clusters are created when there is a specific rational ratio between  $f_0$  and  $f_1$ . For particular frequency

ratios, for example  $f_0/f_1 = 2/3$ , sampling resembles that in Figure 1b. This configuration is particularly detrimental to the security of the generator, as the sampling events are far from the jittered edges. Those particular frequency ratios can be approximated by a rational ratio  $\frac{p}{q}$ , where  $p$  and  $q$  are sufficiently small to create clusters in the distribution phase differences. The larger the values of  $p$  and  $q$ , the greater the number of clusters and the closer the distribution of phase differences to a uniform distance. This makes it possible to define the mutual influence between two ROs according to Equation 1.

**Mutual influence** between ROs appears when there are two **small** integers  $p, q$  such that

$$pf_0 \approx qf_1 \quad (1)$$

This general definition includes the particular case of total locking, which occurs when the two parts are exactly equal:  $pf_0 = qf_1$ . In this case, the output bitstream of the generator will be deterministic. This definition allows us to draw a parallel between the locking phenomenon and underlying physical phenomena. Even if the physical causes of mutual influence between ROs in logic devices are not yet fully understood, the definition given here implies that signals  $s_0$  and  $s_1$  share harmonics. The Fourier spectrum of a signal describes the energy distribution of a signal in a frequency domain. Energy is maximum on small harmonics in a square signal. The probability of inducing electromagnetic coupling with small values of  $p$  and  $q$  is thus greater.

This type of mutual influence compromises the security of the TRNG and must thus be avoided. Systems need to be able to detect and quantify interactions in order to assess the influence, from the least to total locking.

### B. Detecting and quantifying mutual influence

The distribution of phase differences can be used to quantify potential mutual influence between RO signals. The proposed metric should be able to determine whether the distribution of phase differences is uniform or not. This is the purpose of statistical significance tests. These tests are based on statistical values to determine whether a null hypothesis is rejected with sufficient confidence. In our case, the null hypothesis is «  $H_0$  : the phase differences are uniformly distributed ». If the null hypothesis is rejected, i.e. the test fails, then the phase differences can not be considered to be uniformly distributed. Further, statistical adequacy tests not only provide a binary result, whether uniform or not, but also quantify the result by calculating the test statistic.

The most commonly used adequacy test for uniformity is the  $\chi^2$  test. Applied to phase differences, this test requires dividing the nominal period into a number of classes. The  $\chi^2$  test then evaluates whether the samples are uniformly distributed across these classes or not. However, in our case, this test is irrelevant. Indeed, the  $\chi^2$  test is employed to assess the uniformity of the distribution of discrete random variables, whereas the phase differences are continuous.

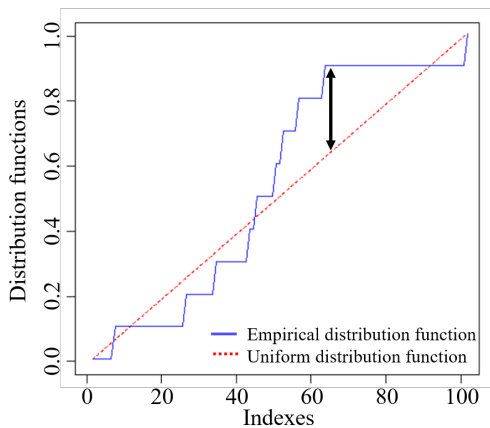


Fig. 2: The Kolmogorov-Smirnov statistic of test  $K_N$  corresponds to the maximum distance between empirical and theoretical distribution functions.

The Kolmogorov-Smirnov test, on the other hand, is suitable for continuous samples [11]. The Kolmogorov-Smirnov test is a statistical goodness-of-fit test that determines whether an empirical distribution follows a theoretical distribution. This test is based on a comparison between empirical and theoretical distribution functions. The first step of this test consists of computing the empirical distribution function that corresponds to

$$F_N(x) = \frac{\#samples < x}{N}, \quad (2)$$

where  $N$  corresponds to the total number of samples on which the Kolmogorov-Smirnov test is performed and the sample values correspond to phase differences. In our case, the theoretical distribution function is the uniform distribution function defined by:

$$F(x) = \begin{cases} 0 & \text{when } x \leq 0 \\ x & \text{when } 0 < x < 1 \\ 1 & \text{when } x \geq 1 \end{cases} \quad (3)$$

The statistical value of the Kolmogorov-Smirnov test corresponds to the maximum distance between the two distribution functions as described by the black arrow in Figure 2. Distances are expressed as :

$$\begin{aligned} K_N^+ &= \sqrt{N} \max_{-\infty < x < +\infty} (F_N(x) - F(x)) \\ K_N^- &= \sqrt{N} \max_{-\infty < x < +\infty} (F(x) - F_N(x)) \end{aligned} \quad (4)$$

The final test statistic value is the greatest of the two distances:  $K_N = \max(K_N^+, K_N^-)$ .  $K_N$  follows a known asymptotic law whose quantiles  $d_{N,1-\alpha}$  are given in a table [12]. The threshold  $d_{N,1-\alpha}$  depends on the number of samples  $N$  and the confidence level  $\alpha$  which is usually set to 5%. According to this value  $K_N$ , the null hypothesis will be rejected or not depending on the following comparisons:

- if  $K_N \geq d_{N,1-\alpha}$ , the null hypothesis is rejected as false with a probability  $\alpha$ : phase difference distribution function is not identical to uniform distribution function.
- if  $K_N < d_{N,1-\alpha}$ , there is not enough evidence to reject  $H_0$  so we consider the phase differences distribution to be uniform.

### III. EXPERIMENTAL SETUP

The use of the Kolmogorov-Smirnov test needs to be evaluated to judge its ability to detect and quantify mutual influence between ROs. For this purpose, we first need to control their influence. Indeed, with two ROs implemented, spontaneous locking could occur and we would not be able to control it. To overcome this difficulty, we simulate the behaviour of an RO using a Delay Line (DL) composed of delay elements that are not looped and whose frequency is controlled externally by a function generator. An RO is implemented close to this delay line. The test bench we used is inspired by the one described in [10].

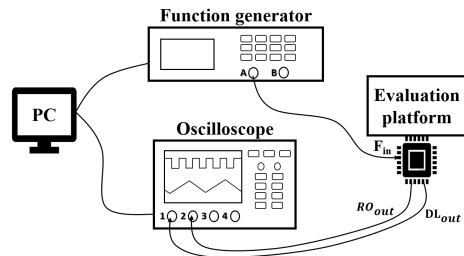


Fig. 3: Experimental setup.

As depicted in Figure 3, the test bench is composed of four main elements. A dedicated hardware evaluation platform with several available field programmable gate array (FPGA) families allows the new metric to be evaluated under the same hardware conditions but on two FPGA families: Intel Cyclone V and Xilinx Spartan 6. The design implemented is simple: only one RO and a disturbing signal on a delay line. No sampling is performed inside the logic device. Our focus is on the output signals of the oscillators not on the output bit stream of the generator. The DL is composed of the same number of delay elements as the RO. A PC sends a command to the function generator to sweep a range of frequencies:  $\pm 6$  MHz around RO nominal frequency  $f_{RO}$  in steps of 50 kHz. This range of frequencies covers possible others ROs frequencies when the topology of ROs of the TRNG are required to be the same. The function generator is an Agilent N81160, which provides a sine wave interference signal  $F_{in}$  of 2.5V peak to peak, because unlike a square wave, a sine wave has only one fundamental frequency. The disturbing signal  $F_{in}$  is sent to the logic device through a differential input connected directly to the delay line. The ring output  $RO_{out}$  and the DL output  $DL_{out}$  are connected to an oscilloscope that measures the phase difference between the two signals. We used LeCroy WL-PBus differential probes because differential probes are more suitable than single-ended probes thanks to their limited sensitivity to noise. The oscilloscope we used is a LeCroy WavePro 404HD with a 4 GHz bandwidth and 20 GS/s time resolution. Acquisition time of the oscilloscope is set to save one million continuous phase difference values for each swept frequency. These values are automatically saved on the PC. The phase differences recovery process is handled by a Python script. The Kolmogorov-Smirnov test is performed

asynchronously and is applied on phase differences once all frequencies have been swept. The target metric must be able to detect total locking as well as mutual influence between oscillators. Four configurations of the design were implemented to challenge the metric on two FPGA families: Intel Cyclone V and Xilinx Spartan 6. In all four configurations, both the RO and the DL comprise the same number of delay elements. On Intel Cyclone V, the ring contains 13 delay elements, resulting in a nominal frequency  $f_{RO}$  close to 150.4 MHz, comparable to the frequency of the RO used in TRNG applications. On Xilinx Spartan 6, the ring is composed of 7 delay elements resulting in a nominal frequency  $f_{RO}$  nearly equal to 139.1 MHz to achieve comparable frequencies on the two families. We now present four ways of placing the two oscillators in relation to each other, from the one with the highest risk of locking to the one with the lowest risk of influence:

**Configuration 1:** all delay elements of the RO and the DL are intertwined in the same logic array block (LAB) of the FPGA. This placement should maximize interaction between the delay elements of both DL and RO. Thus, locking should be amplified [10].

**Configuration 2:** only one delay element of DL is intertwined with one delay element of RO. All the other elements of DL are placed in the LAB right next to it. The intertwined delay elements are those in the centre of each oscillator. As a consequence, this concerns the 7<sup>th</sup> delay element on Intel Cyclone V and the 4<sup>th</sup> delay element on Xilinx Spartan 6. Since only one delay element of each oscillator is intertwined, the mutual influence should be lower.

**Configuration 3:** RO and DL are placed side by side in two different LABs. This configuration is very similar to the previous one, but no delay elements are intertwined. Accordingly there should be even less influence as there is no direct interaction between the delay elements.

**Configuration 4:** in this case, influence between the DL and the RO is minimised. One solution would have been to place both oscillators on either side of the logic device to reduce the effects. However, since the influence between two oscillators placed on the same chip is not well known, RO and DL are instead placed in two FPGAs of the same type in two hardware modules. In this way, the two oscillators share no substrate, no route and no power supply. Consequently there is no physical influence. Each RO and DL is placed in exactly the same position as in Configuration 1, but on two different boards.

The RO is placed and routed in exactly the same way in all four configurations. Only the position of the DL with respect to the RO differs depending on the configuration selected. We try to keep as many parameters as possible unchanged from one configuration to another. Even so, in Intel Cyclone V only, the routing of the RO in Configuration 1 can not be the same as in the other three configurations. The routing tool can not force the routing of Configurations 2 to 4 to be the same as in Configuration 1. Conversely, the routing of Configuration 1 cannot be applied to the other three configurations. As a result, the nominal frequency of the RO in Configuration 1 is

affected. In Configuration 1 in Intel Cyclone V,  $f_{RO}$  is equal to 146.1 MHz whereas in Configurations 2 to 4,  $f_{RO} \approx 150.4$  MHz. This problem did not appear in Xilinx Spartan 6 where RO routing is exactly the same in all four configurations.

#### IV. EXPERIMENTAL RESULTS

In this section, we present the experimental results we obtained on the two FPGAs families. First, the standard deviation of the phase differences seems to be the best locking metric of the four presented in [10]. The standard deviation of a uniform distribution over a range  $[a, b]$  is equal to  $\sigma = \frac{b-a}{\sqrt{12}}$ . Over the range  $0-360^\circ$  (corresponding to a fully covered nominal period), this value is therefore  $103.9^\circ$ . When two signals are totally locked, the phase differences are not distributed over the nominal period, but are always equal to a constant. The values  $a$  and  $b$  are equal and the standard deviation  $\sigma$  of the phase differences drops to  $0^\circ$ . Nevertheless, even though this metric is able to determine when the phase shift distribution is not uniformly distributed over a nominal period, it provides no information about the nature of the distribution when the standard deviation is  $103.9^\circ$ . The left part of Figure 4 shows the standard deviation of the phase differences for each frequency swept in Configuration 1 on Xilinx Spartan 6. When the standard deviation effectively drops to  $0^\circ$  near the nominal frequency of the RO, it is equal to  $103.9^\circ$  from 134 to 137.2 MHz.

However, the right part of Figure 4 represents millions of normalized phase differences values. Each black dot corresponds to a normalized phase difference value, i.e. the phase difference value originally between  $0^\circ$  and  $360^\circ$  is reduced between 0 and 1. White stripes are clearly visible, indicating that some values within the range are never reached. The corresponding distribution is therefore not uniform and the standard deviation does not detect it. This type of distribution reflects an influence that jeopardises the security of the generator and should have been detected by the metric. This justifies the use of the Kolmogorov-Smirnov test as a mutual influence metric.

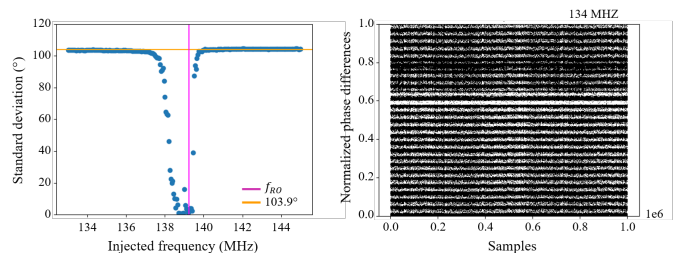


Fig. 4: Standard deviation of phase shift and examples of normalized phase differences for 134 MHz in Configuration 1 on Xilinx Spartan 6 FPGA.

The Kolmogorov-Smirnov test is applied to  $N = 1,000,000$  phase difference samples.

Statistical tests give a binary final result, i.e. they fail if the test statistic is greater than the test threshold  $d_{N,1-\alpha}$  and otherwise pass. When the number of samples  $N$  is greater

than 100, the threshold value is observed to converge rapidly [12]. With  $\alpha = 5\%$ , the threshold  $d_{N,1-\alpha}$  is equal to 1.3581. The test detects either when there is mutual influence between rings, or when the rings are totally locked. This distinction is made using  $K_N$  values which provide information about the strength of mutual influence between ROs. For each frequency, Figure 5 shows whether the test fails or not during the sweep in the four configurations in Intel Cyclone V. The red stripes represent a failed test and mean that the phase differences are not uniformly distributed over the nominal period. As can be seen in the figure, the results complied with our expectations with respect to relative placement of RO and DL. The more we intertwine delay elements, the more the test fails.

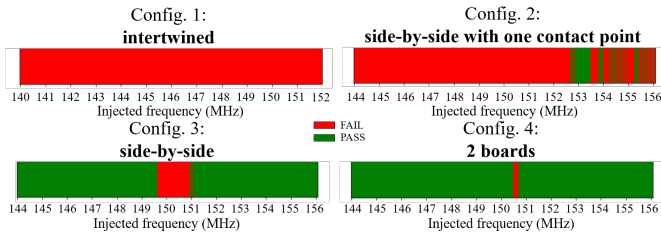


Fig. 5: Results of the Kolmogorov-Smirnov test on Intel Cyclone V FPGA for each configuration implemented.

In addition, the mutual influence and hence the risk of total locking is stronger as we intertwine the delay elements. This observation is shown in Figure 6. The maximum value of  $K_N$  with  $N = 1,000,000$  is  $\sqrt{N} = 1,000$ . When RO and DL are implemented on two different boards in Intel Cyclone V, the highest  $K_N$  value obtained is 24.94 which corresponds to a distance between the empirical and theoretical distribution functions of 0.0249. This value is very low compared to the maximum value of  $K_N$ . For the sake of comparison, when the delay elements are intertwined the maximum value of  $K_N$  is 888.39, which corresponds to a strong influence. These results highlight the importance of focusing not only on the test result, but also on the test statistic  $K_N$  and its meaning.

The result of Configuration 4 can be challenged because RO and DL are implemented on two different boards. However, the Kolmogorov-Smirnov test fails at four frequencies around the nominal frequency  $f_{RO}$ . This point confirms our definition and reinforces the choice of the Kolmogorov-Smirnov test to detect the loss of uniformity in the distribution of the phase differences. In fact, the Kolmogorov-Smirnov test makes it possible to detect particular frequency ratios that occur for two different reasons. Firstly, particular frequency ratios may result from the implementation itself which favours the creation of clusters during sampling, as is the case in Configuration 4. Secondly, the implementation may be ideal for sampling, but a passive or active physical phenomenon may damage the signals and produce particular frequency ratios.

In Configuration 2, failed tests between 144 MHz and 152.7 MHz have to be distinguished by the test statistic  $K_N$ . Figure 7 depicts the  $K_N$  values of the Kolmogorov-Smirnov test for Configuration 2 in Intel Cyclone V more precisely.

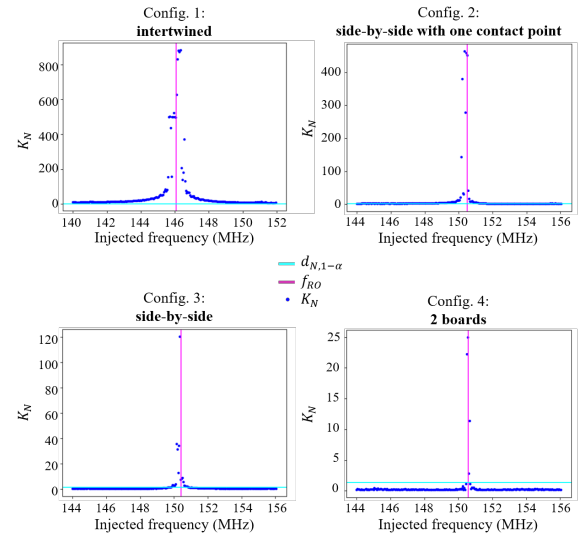


Fig. 6: Kolmogorov-Smirnov test statistic  $K_N$  on Intel Cyclone V FPGA for each configuration implemented.

Compared with the binary test result (PASS or FAIL),  $K_N$  values allow us to distinguish the range of minor influences from the range of major influences. Between 144 MHz and 149.05 MHz, the Kolmogorov-Smirnov test fails but the  $K_N$  values are just above the test threshold  $d_{N,1-\alpha}$ :  $K_N \leq 4$  whereas the threshold equals 1.3581. Then, between 149.05 MHz and 152.7 MHz, the values of  $K_N$  increase significantly, which means that the distribution of phase differences for these frequencies is far from uniform. The mutual influence is stronger up to  $K_N = 464.15$ .

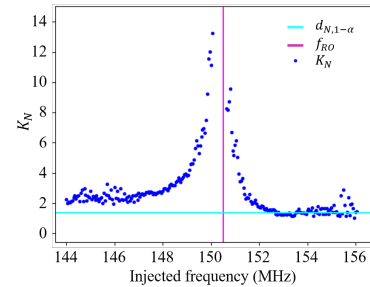


Fig. 7: Zoom on Kolmogorov-Smirnov test statistic  $K_N$  for Configuration 2 on Intel Cyclone V FPGA.

On Xilinx Spartan 6, the binary results of the Kolmogorov-Smirnov test are not as clear as on Intel Cyclone V, as can be seen in Figure 8. The first three configurations fail for all frequencies of the sweep. Again, the binary results of the Kolmogorov-Smirnov test are not sufficient and need to be reinforced by the strength of the influence.

In Figure 9, the  $K_N$  values reveal differences between the two FPGA families. First, the mutual influence is generally stronger on Xilinx Spartan 6 than on Intel Cyclone V. When RO and DL are juxtaposed, the test statistic is always above the threshold with  $K_N \leq 5$ . This order of magnitude is the same as that in Configuration 2 on Intel Cyclone V. On the

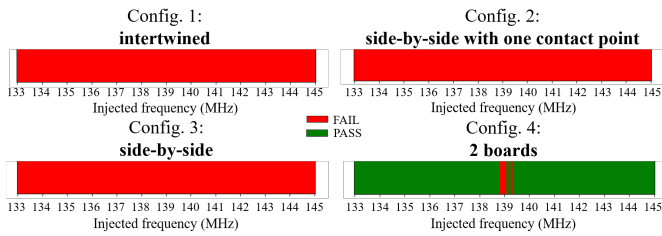


Fig. 8: Results of the Kolmogorov-Smirnov test on Xilinx Spartan 6 FPGA for each configuration implemented.

other hand, on Intel Cyclone V, Configuration 1 shows the influence is twice as strong as in Configuration 2 whereas on Xilinx Spartan 6, the strength of the mutual influence is the same in the two configurations. In addition, for ranges of relatively small influences, i.e. far from the nominal frequency of the RO  $f_{RO}$ , the strength of influence of Configurations 1 and 2 is of the same order of magnitude. This proposed metric will be useful in future work.

To sum up, the differences observed between the two families of FPGAs allow us to affirm that this new Kolmogorov-Smirnov metric can detect both the physical influences that appear when we intertwine the elements, as well as certain frequency ratios that result in the creation of clusters in the sampling. In both cases, the Kolmogorov-Smirnov test indicates security risk for the TRNG.

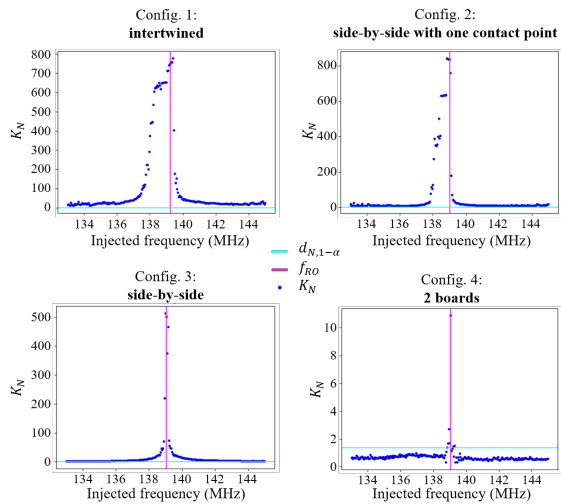


Fig. 9: Kolmogorov-Smirnov test statistic  $K_N$  on Xilinx Spartan 6 FPGA for each configuration implemented.

## V. CONCLUSION

In the RO-based TRNG, jittered samples are obtained correctly if they are uniformly distributed over the sampling period. When setting up the pairs of ROs, it is therefore important to ensure that their nominal frequencies are not related by a ratio of the form  $\frac{p}{q}$ , where  $p$  and  $q$  are small. This avoids the formation of clusters during sampling outside the jittered zones. However, despite these implementation precautions, it is possible that physical phenomena (passive

during operation or active in the case of an EM attack) could disturb the oscillator frequencies, producing a frequency ratio that is dangerous for the generation of random numbers. To investigate this phenomenon, we forced the occurrence of coupling between oscillators and showed that the metric used in the state-of-the-art (standard deviation of the phase differences) is insufficient. In fact, it can be equal to its ideal value, although mutual influence exists between the oscillators.

For this reason, we proposed a new and more precise metric based on a Kolmogorov-Smirnov adequacy test to quantify the mutual influence by checking the uniformity of the samples over the sampled period. On the one hand, this metric can be used to detect initial problems in the implementation of ROs with undesirable frequency ratios without any mutual influence related to physical phenomena. Most importantly, once an acceptable initial RO implementation is achieved, it allows us to detect any degradation caused by underlying physical phenomena.

Our future work will involve using this more precise metric to define RO implementation constraints (placement, routing) to avoid undesirable frequency ratios. In parallel, we will work on implementing this metric in hardware to provide an online embedded test to detect any mutual influence between ROs.

## REFERENCES

- [1] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," April 2010.
- [2] W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators, version 2.0," [online] Available at [https://www.bsi.bund.de/EN/TheBSI/thebsi\\_node.html](https://www.bsi.bund.de/EN/TheBSI/thebsi_node.html), 2011.
- [3] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Trans. Computers*, vol. 56, no. 1, pp. 109–119, 2007.
- [4] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, "On the security of oscillator-based random number generators," Cryptology ePrint Archive, Report 2009/299, 2009, <https://eprint.iacr.org/2009/299>.
- [5] D. Lubicz and V. Fischer, "Entropy computation for oscillator-based physical random number generators," *Journal of Cryptology*, vol. 37, no. 13, pp. 1–33, 2024.
- [6] K. Wold and C. H. Tan, "Analysis and enhancement of random number generator in FPGA based on oscillator rings," *Int. J. Reconfigurable Comput.*, vol. 2009, pp. 501 672:1–501 672:8, 2009.
- [7] N. Bochard, F. Bernard, V. Fischer, and B. Valtchanov, "True-randomness and pseudo-randomness in ring oscillator-based true random number generators," *Int. J. Reconfigurable Comput.*, vol. 2010, pp. 879 281:1–879 281:13, 2010.
- [8] P. Bayon, L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson, and P. Maurine, "Contactless electromagnetic active attack on ring oscillator based true random number generator," in *COSADE 2012*, ser. LNCS, W. Schindler and S. A. Huss, Eds., vol. 7275. Springer, Heidelberg, May 2012, pp. 151–166.
- [9] K. Yamashita, B. Cyr, K. Fu, W. P. Bursleson, and T. Sugawara, "Redshift: Manipulating signal propagation delay via continuous-wave lasers," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2022, no. 4, pp. 463–489, 2022.
- [10] U. Mureddu, N. Bochard, L. Bossuet, and V. Fischer, "Experimental study of locking phenomena on oscillating rings implemented in logic devices," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 66-I, no. 7, pp. 2560–2571, 2019.
- [11] D. E. Knuth, *The art of computer programming, Volume II: Seminumerical Algorithms, 3rd Edition*. Addison-Wesley, 1998.
- [12] R. von Mises, "Mathematical theory of probability and statistics," *Journal of the Royal Statistical Society*, vol. 129, no. 2, pp. 289–291, 1966.