Evaluation and Monitoring of Free Running Oscillators Serving as Source of Randomness

Elie Noumon Allini[1], Maciej Skórski[2], Oto Petura[1], Florent Bernard[1], Marek Laban[3] and Viktor Fischer[1]

[1] Hubert Curien Laboratory, University of Lyon, France,
{elie.noumon.allini,florent.bernard,oto.petura,fischer}@univ-st-etienne.fr
[2] Institute of Science and Technology (IST), Klosterneuburg, Austria,
maciej.skorski@gmail.com
[3] Department of Electronics and Multimedia Communications, Technical University of Košice, Slovakia; Micronic A. S., Bratislava, Slovakia, marek.laban@micronic.sk

Abstract. In this paper, we evaluate clock signals generated in ring oscillators and self-timed rings and the way their jitter can be transformed into random numbers. We show that counting the periods of the jittery clock signal produces random numbers of significantly better quality than the methods in which the jittery signal is simply sampled (the case in almost all current methods). Moreover, we use the counter values to characterize and continuously monitor the source of randomness. However, instead of using the widely used statistical variance, we propose to use Allan variance to do so. There are two main advantages: Allan variance is insensitive to low frequency noises such as flicker noise that are known to be autocorrelated and significantly less circuitry is required for its computation than that used to compute commonly used variance. We also show that it is essential to use a differential principle of randomness extraction from the jitter based on the use of two identical oscillators to avoid autocorrelations originating from external and internal global jitter sources and that this fact is valid for both kinds of rings. Last but not least, we propose a method of statistical testing based on high order Markov model to show the reduced dependencies when the proposed randomness extraction is applied.

Keywords: Physical source of randomness · physical RNG · stochastic model · entropy

Introduction

In modern cryptographic systems, security is based on the statistical quality and on the unpredictability of confidential keys. These keys are generated in random number generators (RNGs) using random physical phenomena that occur in the hardware devices in which the system is implemented. A widespread source of randomness in digital devices is the jitter of the clock signal generated inside the device using free running oscillators such as ring oscillators [SMS07, BLMT11, RYDV15], or self-timed rings [CFAF13].

The statistical quality and unpredictability of the generated numbers depend on the size and quality (e.g. the spectrum) of the clock jitter. It is therefore good practice to continuously monitor this jitter using an embedded jitter measurement method. As required in the document AIS-20/31 published by the German Federal Office for Information Security (German acronym BSI) [KS11], the measured jitter parameters should then be used as input parameters in the stochastic model used to estimate entropy, which characterizes the unpredictability of generated numbers.

Generally, many sources of randomness contribute to the overall entropy rate at the output of the RNG based on free running oscillators [VABF10]:

Licensed under Creative Commons License CC-BY 4.0.

IACR Transactions on Cryptographic Hardware and Embedded Systems ISSN 2569-2925,
Vol. 2018, No. 3, pp. 214–242
DOI:10.13154/tches.v2018.i3.214-242

1. *Secure sources* – random sources such as thermal noise, which are considered to be the best sources of randomness, because of their large and almost uniform signal spectrum similar to white noise, they are mutually independent, and unavoidable (i.e. they cannot be manipulated by the attacker);

2. *Security critical sources* – random sources such as low frequency noises that feature some autocorrelation, which reduces the entropy rate at the generator output, while making entropy estimation very complex because of long term dependencies;

3. *Dangerous sources* – environmental, data dependent and correlated sources, which can be random or deterministic. Their contribution to random number generation must be avoided by the design, since they can be manipulated. If the manipulation cannot be avoided, it must at least be detectable through dedicated embedded tests.

In practice, most (and sometimes all) of these sources of randomness coexist. This would not be a big security issue if: 1) only the contribution of secure sources were taken into account when estimating the entropy rate; 2) the generated numbers were impossible to manipulate.

In [SMS07], Sunar *et al.* use an urn stochastic model to estimate the entropy rate at the output of the generator using a huge number of ring oscillators, which the authors claimed were independent. However, the model does not account for possible dependencies between the outputs of the ring oscillators, which can even cause the rings to lock [CBFF12].

In [BLMT11], Baudet *et al.* propose a comprehensive stochastic model for an elementary oscillator based random number generator sampling the jittery clock signal. In their model, the entropy rate at the generator output is estimated from the variance of the random jitter component that originates from the thermal noise.

The output numbers generated by both generators may be biased depending on the duty cycle of the sampled signal(s). Although both generators use the clock signal generated in the rings as a source of randomness, only the model proposed by Baudet *et al.* estimates the entropy rate from the jitter component originating from the thermal noise and consequently avoids overestimating entropy.

Evaluating the contribution of thermal and low frequency noises to the generated randomness is no simple task. In [HBFT14], Haddad *et al.* computed the variance of the jitter for different accumulation times and then computed the jitter component originating from the thermal noise by curve fitting. This method has two disadvantages: 1) its precision depends to a great extent on the precision of the curve fitting algorithm; 2) it is not suitable for monitoring the jitter inside the device.

In [FL14], Fischer and Lubicz proposed a method of evaluation of the variance of the random jitter originating from the thermal noise that can be embedded in logic devices and hence used for online evaluation of the entropy rate at the output of the generator. However, depending on the initial phase of the two clock signals and the jitter accumulation time, the method can produce incorrect results. The error can be corrected by using different accumulation times, but it is not easy to make this correction automatic.

In [KS08], Killmann and Schindler used a pair of noisy diodes as a source of randomness and an operational amplifier, a Schmitt trigger and a counter of edges as a time-to-digital converter transforming the noise into the raw binary signal. Surprisingly, the time-to-digital conversion based on counters had not been previously studied in the context of the use of free running oscillators.

Our contributions: 1) We show that counting the periods of the jittery clock signal, representing a time-to-digital conversion, gives random numbers of significantly better quality than the methods based on sampling the jittery signals. What is more, the counter values can be used to characterize and to continuously monitor the source of randomness. 2) We propose to use Allan variance of counter values instead of the commonly used statistical variance to evaluate the jitter, since it is not sensitive to low frequency components of the jitter originating from low frequency noises, such as flicker noise, which are known to be autocorrelated. The proportion of thermal noise in the total jitter can thus be more easily measured inside the device with no error or overestimation. 3) We demonstrate that by using two identical rings instead of one ring and one quartz oscillator, the impact of not only external, but also of internal global jitter sources can be significantly reduced and render the generator much more robust. 4) We propose to use a statistical method based on a high order Markov model and show how efficient it is in detecting dependencies and correlations in low quality generators.

The paper is organized as follows: in Section 1, we provide the theoretical background and analyze state-of-the-art methods related to our approach. In Section 2, we describe the experimental setup and analyze the impact of the type of the oscillator on the commonly used statistical variance and on Allan variance. In Section 3, we present the results of implementation of variance computation circuitries in hardware and discuss the impact of the measurement circuitry and of the additional logic represented by an AES cipher on the source of randomness in Section 3.2. In Section 4, we discuss the main results. We present our conclusions in Section 5.

1 Theoretical background

In purely digital devices, which are currently used to implement cryptographic systems, analog noise signals such as thermal noise cannot be directly exploited. Instead, the designer can use the fact that electrical noises are transformed in free running oscillators into uncertainties in timings of generated digital clock signals, which can be observed as a jitter in the time domain and as a phase noise in the frequency domain [DDS18].

In logic devices, the most frequently used free running oscillators are ring oscillators (ROs) and self-timed rings (STRs), because both are easy to implement using standard logic gates. ROs are usually composed of an odd number of inverters as shown in the top panel of Fig. 1 (a) or a NAND gate and a sufficient number of non-inverting buffers, as shown in the bottom panel of Fig. 1 (a). In ROs, which are also called single-event ring oscillators [Nis16], only one event (the rising or falling edge of the clock signal) propagates at any given time in the ring. Its propagation time is impacted by noises that modify the slope of the rising and falling edges and the reference voltage of inverters (or buffers).

In STRs, also called multi-event oscillators without signal collision, several events can propagate over the ring at the same time. The STR is composed of L stages, each consisting of a Müller gate and an inverter (see Fig. 1 (b)) [Nis16]. F_i is the forward input of the i-th stage, R_i is the reverse input of the same stage, and C_i is the output of the stage. If the forward and reverse input values differ, the forward input value is written to the stage output. Otherwise, the previous output value is maintained.

Ring oscillators are simpler and hence less expensive than STRs, so many rings can be used to increase entropy [SMS07]. STRs are more complex, but multiple outputs of the same ring can be used to increase entropy [CFFA13].

The randomness originating from electrical noises, which is transformed in the free running oscillators into a clock jitter, can be further transformed into random numbers obtained as a chain of 1-bit or n-bit random values by: 1) sampling the jittery clock signal(s) after a sufficiently long time interval required for entropy accumulation as shown in Fig. 2 (a) [SMS07], [BLMT11]; 2) by counting the periods of the jittery clock signal during the time interval as shown in Fig. 2 (b) [KG04].

While the first method based on sampling may be preferred because of its simplicity, it is very sensitive to dependencies between the clock signals and also to the duty cycle of the sampled clock signal, which can cause a significant bias in the generated numbers [BLMT11].

Although the second method based on counting the periods of the jittery clock signal

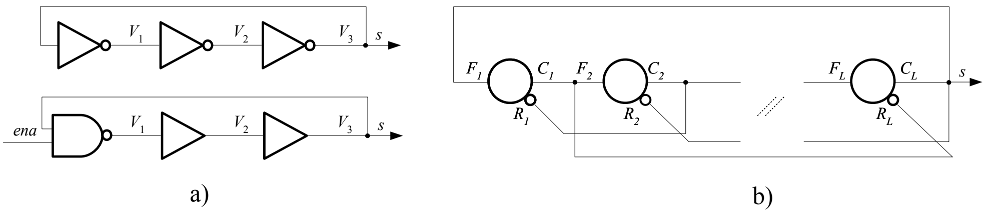

Figure 1: Generation of the jittery clock signal s in free running oscillators: ring oscillators (a) and self-timed rings (b)

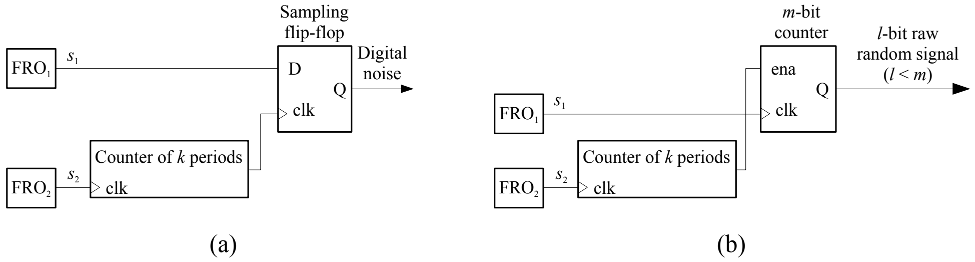

Figure 2: Generation of random numbers from the jittery clock signals s_1 and s_2 using a sampler (a) and a counter (b). Signals s_1 and s_2 are generated in two free running oscillators (FRO) of the same type and topology.

adds some complexity to the RNG design, we will show that it effectively removes dependencies between the clock signals by transforming random events from the time domain to the frequency domain, and even removes the dependence of generated numbers on the duty cycle of the jittery clock signal. We will also show that the counter can be used as a basis for dedicated embedded tests.

In the following sections, we will demonstrate and justify the relationship between the measured variance of counter values and that of the jitter present in clock signals s_1 and s_2.

1.1 Characterization of the source of randomness by a statistical variance – a pitfall

Statistical variance characterizes the deviation of a random variable from its mean value. More precisely, if X is a square-integrable random variable, then its statistical variance can be computed as [Žit10, CM98, Eng06]:

$$\mathrm{Var}(X) \;=\; \mathbb{E}\left(X^2\right) - \left(\mathbb{E}(X)\right)^2 \qquad (1)$$

where $\mathbb{E}$ denotes the statistical average. The estimate of this variance on a set of M samples $\{x_i\}_{1\leqslant i\leqslant M}$, is given as [Ril08]:

$$\mathrm{Var}(X) \;=\; \frac{1}{M}\sum_{i=1}^{M} x_i^2 - \left(\frac{1}{M}\sum_{i=1}^{M} x_i\right)^2 . \qquad (2)$$

1.1.1 Limitations of statistical variance in the presence of low frequency noises

We denote the output frequency of the oscillator under study by $\nu(t)$. The fractional frequency of the output is defined as:

$$y(t) \;=\; \frac{\nu(t) - \nu_0}{\nu_0}, \tag{3}$$

where ν_0 is the nominal frequency of the oscillator.

In oscillators, random fluctuations are often characterized by the power law spectrum [AB81]:

$$S_y(f) \;=\; h_\alpha f^\alpha, \tag{4}$$

where f is the Fourier frequency, h_α the intensity of the particular noise process and α a constant that characterizes this process. The typical values of α, with corresponding noise types, that often appear in the literature are $+2$ (white noise phase modulation), $+1$ (flicker noise phase modulation), 0 (white noise frequency modulation), -1 (flicker noise frequency modulation) and -2 (random walk frequency modulation). Knowing that random fluctuations are due to the above mentioned types of noises, the power spectral density of y can be expressed as [Uhr07]:

$$S_y(f) \;=\; \sum_{\alpha=-2}^{2} h_\alpha f^\alpha. \tag{5}$$

Under the assumption that y is a zero-mean stationary random process, its statistics do not change over time. This implies that y is an infinite signal that can only be observed through a time window defined by the function h_τ. The observed signal y_τ can then be considered as the response of a filter, with the impulse response h_τ, to the random input y. The power spectral densities of y and y_τ are therefore related by [CM98]:

$$S_{y_\tau}(f) \;=\; S_y(f)\left|H_\tau(f)\right|^2, \tag{6}$$

where H_τ is the Fourier transform of h_τ. Based on the Wiener-Khinchin theorem, the autocorrelation function of y_τ can then be computed as [CM98]:

$$R_{y_\tau}(\xi) \;=\; \int_{-\infty}^{+\infty} S_{y_\tau}(f) e^{i2\pi\xi f} df \;=\; \int_{-\infty}^{+\infty} S_y(f)\left|H_\tau(f)\right|^2 e^{i2\pi\xi f} df. \tag{7}$$

Because the process has zero mean, the variance is the autocorrelation function, evaluated at 0, hence [Ril08]:

$$\mathrm{Var}(y) \;=\; \mathbb{E}\left(y^2\right) \;=\; \mathbb{E}\left(y_\tau^2\right) \;=\; \int_{-\infty}^{+\infty} S_y(f)\left|H_\tau(f)\right|^2 df. \tag{8}$$

The choice of h_τ reveals how samples of the signal y are used in the variance computation. Since in the case of statistical variance, we are interested in consecutive samples, the corresponding time window has the form depicted in Fig. 3.

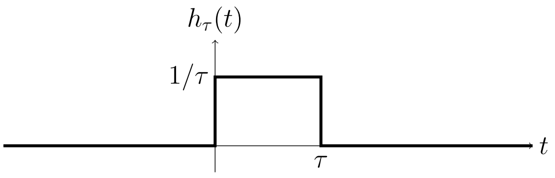

Figure 3: Time window of the statistical variance

The magnitude squared transfer function of the statistical variance is thus [Uhr07]:

$$|H_\tau(f)|^2 = \left(\frac{\sin \pi\tau f}{\pi\tau f}\right)^2. \tag{9}$$

The statistical variance of the signal can then be expressed as:

$$\mathrm{Var}(y) \;=\; \sum_{\alpha=-2}^{2} \frac{h_\alpha}{(\pi\tau)^2} \int_0^{f_h} f^{\alpha-2} \sin^2(\pi\tau f) df, \tag{10}$$

where f_h is the cutoff frequency of the oscillator.

Riemann criterion for improper integrals Given two real numbers b and p such that $b > 0$, the function $t \longmapsto t^p$ is integrable in the improper sense on $(0, b]$ if, and only if, $p > -1$ [CN14, Chapter 10].

In Eq. (10), the integrand is equivalent to $\pi^2\tau^2 f^\alpha$ as $f \to 0$. The Riemann criterion therefore shows that the integral does not converge for $\alpha = -1$ and $\alpha = -2$ corresponding to low frequency noises. Consequently, it is not possible to compute the variance when the data is affected by low frequency noises. In other words, statistical variance should not be used when low frequency noises are not negligible.

For this reason, it is recommended to use other types of variance that converge in the presence of low frequency noises [Ril08, All87]. One example of this type of variance is the Allan variance, which is widely used to study the frequency stability of clocks and oscillators [All66]. Next, we will show that the Allan variance should also be preferred in entropy rate estimation.

1.2 Allan variance

We recall that y denotes the fractional frequency of the oscillator. Thus, the average fractional frequency is defined as:

$$\overline{y}(t) \;=\; \frac{1}{\tau} \int_t^{t+\tau} y(u)du. \tag{11}$$

It corresponds to the average frequency deviation over a time interval of length τ. If the frequency data are acquired periodically with a sampling period of τ, the obtained fractional frequency series is denoted $(\overline{y}_i)$, where $\overline{y}_i$ is the i^{th} acquired sample. The Allan variance of the frequency deviation of y is then defined as [AB81]:

$$\sigma_y^2(\tau) \;=\; \frac{1}{2} \mathbb{E} \left(\overline{y}_{i+1} - \overline{y}_i\right)^2. \tag{12}$$

We denote Avar(y) the Allan variance of y as in [Ril08]. An estimate of this variance in a data set comprised of M average fractional frequency samples, is given as [Ril08]:

$$\mathrm{Avar}(y) = \sigma_y^2(\tau) \;=\; \frac{1}{2(M-1)} \sum_{i=1}^{M-1} \left(\overline{y}_{i+1} - \overline{y}_i\right)^2. \tag{13}$$

1.2.1 Convergence in the presence of low frequency noises

Unlike statistical variance, the Allan variance computes the difference of consecutive samples. This yields the time window presented in Fig. 4, with a magnitude squared transfer function given by:

$$|H_\tau(f)|^2 = 2\left(\frac{\sin \pi\tau f}{\pi\tau f}\right)^2 \sin^2 \pi\tau f. \tag{14}$$

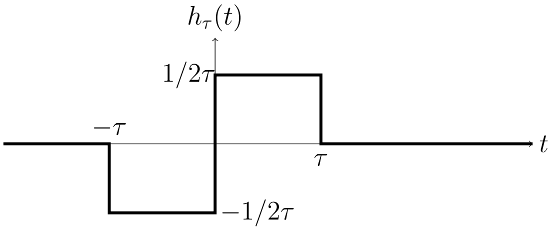

Figure 4: Time window of the Allan variance

This makes it possible to write the signal variance as:

$$\text{Avar}(y) = \sigma_y^2(\tau) \;=\; \sum_{\alpha=-2}^{2} \frac{2h_\alpha}{(\pi\tau)^2} \int_0^{f_h} \sin^4(\pi\tau f) f^{\alpha-2} df. \tag{15}$$

In this new case, the integrand is equivalent to $\pi^4\tau^4 f^{\alpha+2}$ as $f \to 0$. The Riemann criterion for $f \to 0$ ensures that this integral converges when $\alpha > -3$, and thus guarantees the accuracy of the Allan variance, even when the data are affected by low frequency noises ($\alpha = -1$ and $\alpha = -2$).

Next, we present the general properties of the Allan variance. Readers interested in the proofs of these properties should refer to Appendix A.

Theorem 1 (General properties of the Allan variance)**.** *1. The Allan variance coincides with the statistical variance of any stationary and uncorrelated random process.*

2. If λ is a real number and x is a stationary random process, then λx is also a stationary random process and:

$$\text{Avar}(\lambda x) \;=\; \lambda^2 \,\text{Avar}(x). \tag{16}$$

3. If x and y are two independent stationary random processes, the following equation is valid:

$$\text{Avar}(x + y) \;=\; \text{Avar}(x) + \text{Avar}(y). \tag{17}$$

Since the measurement principle of the jitter is based on counter values, the properties of the Allan variance presented here will be used to establish the link between the variance of counter values and the variance of the jitter.

1.3 Link between the variance of counter values and of the jitter

We assume that both s_1 and s_2 contain jitter that causes variations in counter values. Before using the variance of a population of counter values as a measure of quality of the source of randomness, we need to determine and justify the relationship between this variance and the variance of the jitter on both signals s_1 and s_2.

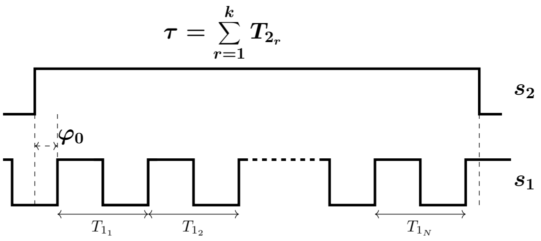

Figure 5: Timings in counting the periods of signal s_1

As mentioned above, the counter values are obtained by counting the number of periods of the measured clock signal s_1 during a time interval τ defined by the reference clock signal s_2. Of course, in practice, both signals feature a jitter, however, to simplify the computation, we include the jitter of signal s_1 in that of signal s_2 as done in [BLMT11]. Consequently, the period T_2 of signal s_2 can be considered as a random variable of standard deviation (see [BLMT11, Appendix C]):

$$\sigma_{eq} \;\simeq\; \sqrt{\sigma_2^2 + \frac{T_2}{T_1}\sigma_1^2}, \tag{18}$$

and the period T_1 of signal s_1 as a constant. The measurement time $\tau = \sum_{r=1}^{k} T_{2_r}$ is thus a random variable. This time defines only the length of the time period, not the position of the initial phase φ_0 of the signal s_1 when the measurement (the counting) starts (see Fig. 5). However, to measure the jitter more accurately, the initial phase φ_0 has to be taken into account. This initial phase is independent of τ, since its value does not depend on the length τ.

Because T_1 is constant, the counter value N is a random variable defined as:

$$N \;:=\; \max\left\{k \in \mathbb{N},\; \varphi_0 + \sum_{r=1}^{k} T_{1_r} \leqslant \tau\right\} \;=\; \max\left\{k \in \mathbb{N},\; \varphi_0 + kT_1 \leqslant \tau\right\}. \tag{19}$$

The value N thus satisfies the inequality:

$$\varphi_0 + N \times T_1 \;\leqslant\; \tau \;<\; \varphi_0 + (N+1) \times T_1, \tag{20}$$

which is equivalent to:

$$N \;\leqslant\; \frac{\tau - \varphi_0}{T_1} \;<\; N + 1. \tag{21}$$

It then follows that N can be written as:

$$N \;=\; \left\lfloor \frac{\tau - \varphi_0}{T_1} \right\rfloor. \tag{22}$$

It thus exists $0 \leqslant \varepsilon < 1$ such that:

$$N \;=\; \frac{\tau - \varphi_0}{T_1} - \varepsilon \tag{23}$$

According to Sheppard's correction [She97], ε is a random variable that is uniformly distributed over $[0, 1)$. Since it is independent of $\frac{\tau-\varphi_0}{T_1}$, using Eq. (16) and (17) from Theorem 1, the following equation holds:

$$\mathrm{Avar}(N) \;=\; \mathrm{Avar}\left(\frac{\tau - \varphi_0}{T_1}\right) + \mathrm{Avar}(\varepsilon) \;=\; \frac{\mathrm{Avar}(\tau) + \mathrm{Avar}(\varphi_0)}{T_1^2} + \frac{1}{12}. \tag{24}$$

It is important to note that the Allan variance of counter values always overestimates the Allan variance of the jitter per unit of time (e.g the signal period). The correction must be applied by subtracting $\mathrm{Avar}(\varepsilon) = \frac{1}{12}$ and $\frac{\mathrm{Avar}(\varphi_0)}{T_1^2}$.

As $\frac{\mathrm{Avar}(\varphi_0)}{T_1^2} \in \left[0, \frac{1}{12}\right]$ (the maximum is obtained if φ_0 is uniformly distributed over $[0, T_1)$), according to Eq. (24), $\mathrm{Avar}(\tau) = T_1^2\,\mathrm{Avar}(N) - \frac{T_1^2}{12} - \mathrm{Avar}(\varphi_0) \geq T_1^2\,\mathrm{Avar}(N) - \frac{T_1^2}{12} - \frac{T_1^2}{12}$. As we do not want to overestimate the jitter, a conservative approach is to take the minimum value for $\mathrm{Avar}(\tau)$ that is:

$$\mathrm{Avar}(\tau) \; = \; T_1^2 \cdot \mathrm{Avar}(N) - \frac{T_1^2}{6}. \tag{25}$$

Using Eq. (25), the variance of the accumulated jitter can be computed from the variance of counter values. This justifies using counter values to estimate the jitter.

2 Study and setup of the variance measurement

To study the difference between statistical variance and the Allan variance in different conditions, we first implemented the circuit presented in Fig. 2 (b) in the hardware. Four different hardware configurations were tested in an Intel Cyclone V FPGA:

- **Configuration 1**: Signal s_1 of 127 MHz was generated in an RO and signal s_2 came from a low jitter quartz oscillator generating a stable 125 MHz clock.
- **Configuration 2**: Both signals (s_1 and s_2) were generated in two ROs with the same number of elements, oscillating at a frequency of 125 and 127 MHz, respectively.
- **Configuration 3**: Signal s_1 of 128 MHz was generated in an STR and signal s_2 came from a low jitter quartz oscillator generating a stable 125 MHz clock.
- **Configuration 4**: Both signals (s_1 and s_2) were generated in two STRs with the same number of elements and oscillating at a frequency of 130 and 128 MHz, respectively.

The counter values were sent to a PC via a simple serial interface and evaluated in the software. The jitter accumulation time τ was set up from the PC using the serial link.

To obtain meaningful and reliable embedded measurements, we first needed to establish the right operating parameters. These parameters are k – the number of periods of signal s_2, which determines the accumulation time τ and M – the number of samples from which the variance will be computed.

We performed a series of variance measurements for different values of M in order to find an acceptable compromise between the measurement time and precision. We used $k = 30\,000$ for this study. Measurement results are shown in Fig. 6.

Figure 6 clearly shows the advantage of the Allan variance: it changes very slightly and only for low values of M, while the statistical variance increases with M and its values fluctuate. This fluctuation occurs because low frequency noises affect the signal periods. We selected $M = 4096$ as a compromise between the number of statistical data (which impacts the precision of the measurement) and the measurement time. To obtain coherent results, the same values of M were used when measuring variance and Allan variance.

We next studied the impact of the accumulation time $\tau = \sum_{r=1}^{k} T_{2_r}$ on the measured variance. We observed the variances and Allan variances of counter values from two ring oscillators as well as two self-timed rings with k ranging from 300 to several million. The results are presented in Fig. 7 and Fig. 8.

We observed that ROs and STRs behave similarly in terms of variance dependence on k. This means that the jitter accumulates in both structures in a very similar way.

We also observed that for low values of k ($k < 1\,000$), the computed variances varied probably because of the quantization noise, rather than random noises. Indeed, for these low values of k, the counter values varied only very slightly.

Last but not least, we observed that for sufficiently high values of k ($k > 10\,000$), the Allan variance was always lower than the statistical variance. This proves that statistical variance overestimates the proportion of uncorrelated noise in the total accumulated jitter.

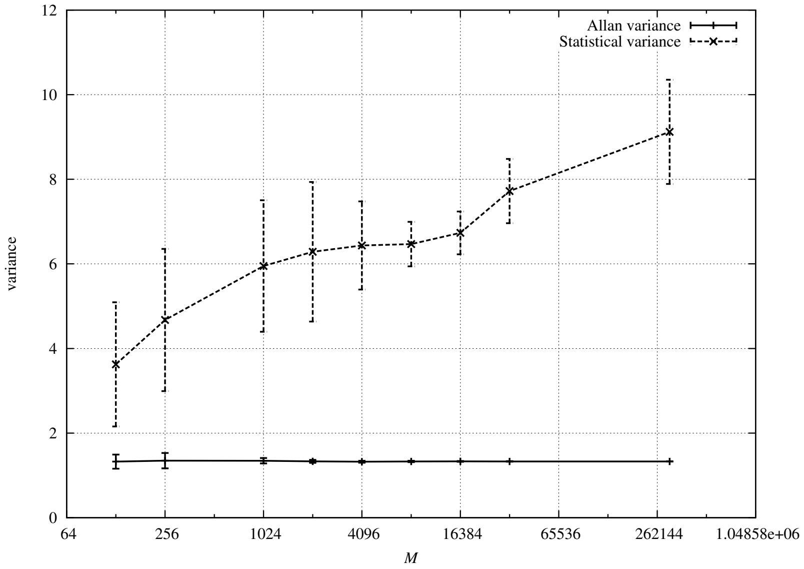

Figure 6: Dependence of the statistical variance and Allan variance on the number of measurements M

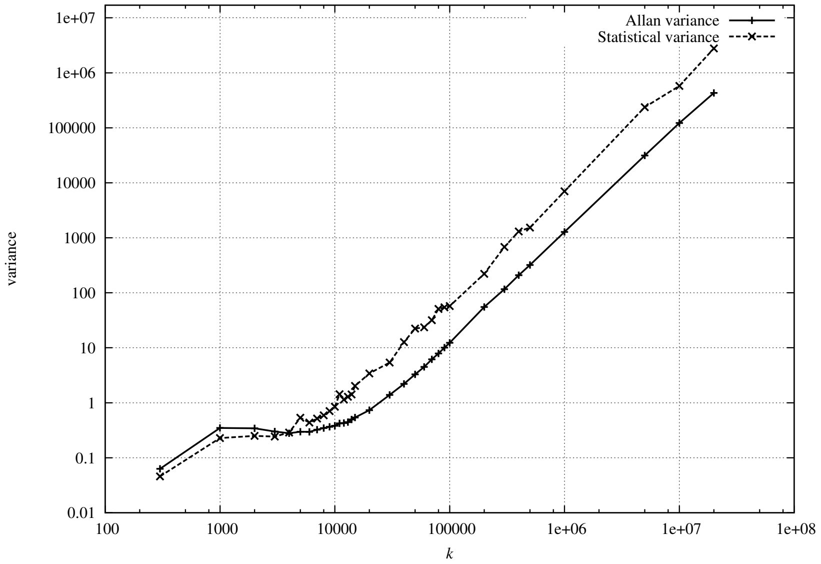

Figure 7: Variance and Allan variance of counter values depending on the measurement interval k, with two ROs as sources of the jittery clock signals

2.1 Accuracy of Allan variance estimation

The sample estimate given in Eq. (13) (time-average) approximates the true value[1] from Eq. (12) (average over process randomness) well, provided that the series is wide-sense stationary and has rapidly decreasing correlations[2] [LG08]. One can think of small correlations as a *short memory* of the process: substantial new information is gained with every sample, so that the estimate becomes increasingly accurate.

[1] This is formally defined as ergodicity in mean.

[2] We discuss basic facts about process correlations in Appendix B.

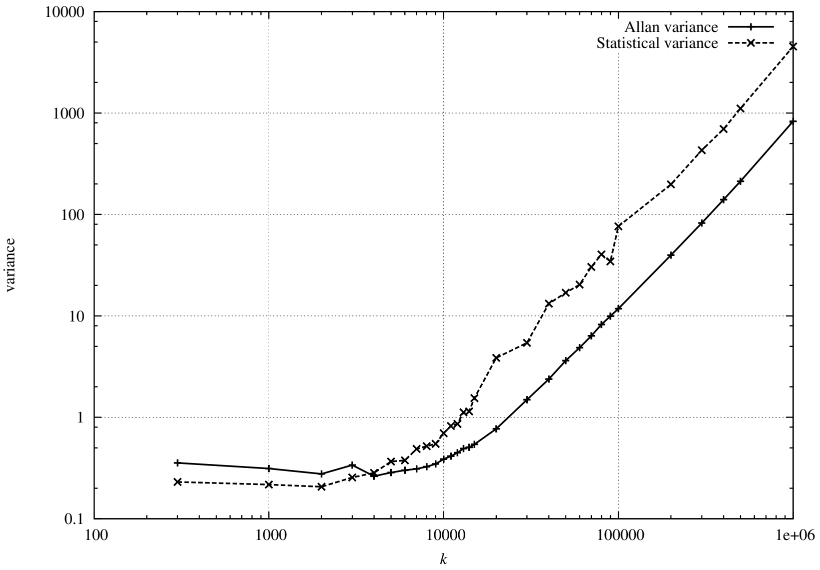

Figure 8: Variance and Allan variance of counter values depending on the measurement interval k, with two STRs as sources of the jittery clock signals

To check the quality of our estimate, we examined the process of counter values (used to compute the variance) and their first differences (used to compute the Allan variance) in more detail. We collected the data from four different hardware configurations, *i.e.* Configurations 1 to 4 described earlier in this section. All projects used $k = 30\,000$ periods of s_2 to set the counting time. The behavior across all experiments is summarized at high level in Table 1.

Table 1: Comparison of raw counter values and their first differences.

counter data	memory	autocorrelations
raw values	long	strong
first differences	short	weak

The autocorrelations were considerably reduced in first differences, as shown in Fig. 9. This confirms that differencing subsequent counter values is a good way of eliminating low frequency components and reducing correlations. Frequency noise in the ring oscillators is

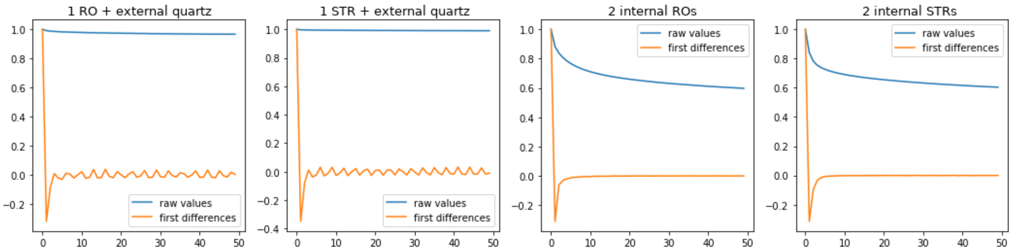

Figure 9: Autocorrelations of raw counter values and their first differences in the four hardware configurations. At lag 0, the correlation is by definition always equal to 1.

modeled by a process with stationary first differences in the theoretical literature[3] [Gre83]; this is consistent with our experiments. More experiments are presented in Appendix B.

Based on our empirical evidence, we assume that the correlation is zero for sufficiently large lags. Under the mild assumption that the difference process is correlated Gaussian, we can bound estimation errors in the Allan variance computations (quantifying convergence rate). More precisely, we assume

> **Model assumption:** The process of counter differences is stationary normal, with zero correlations for lags larger than p.

The technical result and the corrolary regarding Allan variance are given below.

Lemma 1. *Let $Z = \{Z_i\}$ be a zero-mean stationary normal process with autocorrelation function zero for lags larger than p. Then for large M we have*

$$\operatorname{Var}\left(\frac{1}{M}\sum_{i=1}^{M} Z_i^2\right) = O(p/M),$$

in particular

$$\frac{1}{M}\sum_{i=1}^{M} Z_i^2 \longrightarrow \mathbb{E}(Z^2) \quad \textit{in probability.}$$

By applying Lemma 1 to the differences of counter values $\Delta\bar{y}_i = \bar{y}_i - \bar{y}_{i-1}$ we obtain

Corollary 1 (Consistency of Allan variance estimation[4])**.** *Under the chosen model, the Allan variance estimate $\frac{1}{2(M-1)}\sum_{i=1}^{M}(\Delta\bar{y}_i)^2$ in Eq. (13) is asymptotically (for large M) unbiased and consistent, with variance $O(p/M)$.*

2.2 Evaluation of randomness in counter values

We propose to use the least significant bit of the counter values (or of their first difference) as random values. To evaluate the quality of the generated sequence, we model dependencies between subsequent bits by higher-order Markov chains. First, we recall some basics on Markov chains in order to introduce the theorem used to compute the min-entropy rate, which is more conservative than the Shannon entropy rate. We then empirically (based on data generated under different hardware configurations) compare our evaluation technique with the entropy estimation methods in AIS31 and NIST 800-90B.

2.2.1 Theoretical Background

Model The Markov chain model of order d assumes a sequence of random variables $\{U_i\}_i$ over the common state space S ($S = \{0,1\}$ in our case), such that:

- the next state distribution depends only on previous d states (short memory)

$$\Pr[U_i|U_{i-1}, U_{i-2}, \ldots, U_0] = \Pr[U_i|U_{i-1}, U_{i-2}, \ldots, U_{i-d}],$$

- the next state distribution is a function of state values (that are time homogeneous)

$$\Pr[U_i = u_i|U_{i-1} = u_{i-1}, U_{i-2} = u_{i-2}, \ldots, U_{i-d+1} = u_{i-d+1}] = f(u_i; u_{i-1}, \ldots, u_{i-d+1}).$$

[3] Equivalently, phase noise is modeled by a process with second stationary differences.

[4] See proof in Appendix C.

Reduction of a chain of order d to a first order chain A Markov chain of order d can be reduced to a chain of order 1 by introducing the *sliding window of length d*. More precisely, if $\{U_i\}_i$ is a Markov chain of order d with the set of possible states S then the blocks $W_i = (U_i, U_{i-1}, \ldots, U_{i-d+1})$ form a Markov chain of order 1 with states $S^d = \underbrace{S \times \ldots \times S}_{d}$.

The transition matrix can be then estimated by counting the transitions between subsequent states (blocks); namely $P_{v,w} = \frac{\#\{i:W_i=w \text{ and } W_{i-1}=v\}}{\#\{i:W_{i-1}=v\}}$ is the transition probability from v to w. In our case U_i are bits and the assumed order is 8, thus the transformed chain W_i has states $\{0,1\}^8$ and the transition matrix has the size $2^8 \times 2^8$.

Min-entropy rate Entropy rates (understood as the entropy per bit in long sequences) can generally be computed from the transition matrix. However, computation of the min-entropy rate is more complicated than that of the Shannon entropy and *does not have a closed-form formula*. We refer the reader to [KV16] for a detailed discussion of how different definitions of entropy (Shannon entropy, Renyi entropy, min-entropy) can be computed using Markov chains; below we state the result for min-entropy.

A sequence of states $s_1, \ldots, s_{\ell+1}$ is called a *loop* if $s_1 \neq s_2 \neq \ldots \neq s_\ell$ and $s_1 = s_{\ell+1}$, where ℓ is the length of the loop. The min-entropy rate is then determined as follows.

Theorem 2 (Min-entropy rate of Markov chains [KV16])**.** *Let P be the transition matrix of an irreducible and aperiodic Markov chain with the state space S. Then*

$$H_\infty(P) = \min_{\ell} \min_{(s_1,\ldots,s_{\ell+1})\in\mathcal{C}_\ell} \frac{1}{\ell}\sum_{k=1}^{\ell} \log \frac{1}{P_{s_k,s_{k+1}}} \tag{26}$$

where $\mathcal{C}_\ell$ denotes the set of all loops of length ℓ and $P_{s_k,s_{k+1}}$ the probability of the transition from state s_k to state s_{k+1}.

2.2.2 Implementation

Language We implemented the procedure to estimate the min-entropy rate of a Markov chain in Python; to increase the speed, parts of the code were compiled to C by the Cython module. For computation, we used the Numpy library with double precision (64 bits).

Algorithmic issues Computation of the transition matrix requires one pass on the data file. The value in Eq. (26) is found by *dynamic programming*; namely, for every ℓ and every pair of states s', s'' we compute $r(s', s'', \ell) = \min_{s_1=s',s_2,\ldots,s_{\ell-1},s_\ell=s''} \sum_{i=1}^{k} \log \frac{1}{P_{s_k,s_{k+1}}}$ where $s_1, \ldots, s_{\ell-1}$ are different; dynamic programming is used to update values of r when changing from ℓ to $\ell+1$. Once we have these numbers, we can determine the value in Eq. (26). Because the formula assumes different states $s_1, \ldots, s_\ell$ we have that $\ell \leqslant |S|$. Computing all $r(s', s'', \ell)$ for $s' \in S, s'' \in S$ and $\ell \leqslant |S|$ requires a memory size of about $|S|^3$ multiplied by the size of the float placeholders.

Parameters Based on the dependencies indicated by the results of the autocorrelations, we decided to use $d = 8$. We therefore study transitions between blocks of consecutive $d = 8$ bits, and the size of the transition matrix is $2^8 \times 2^8$.

Table 2: Summary of implementation of the Markov chain model for entropy estimation.

Order (window)	States	Trans. matrix	Memory used	Dyn. program. loops
8	$\{0,1\}^8$	$2^8 \times 2^8$	128MB	2^8

2.2.3 Experiments on the generation of random bit streams from free running oscillators

In our experiments on the generation of random numbers using free running oscillators, we wanted to compare randomness extraction using the two methods presented in Fig. 2. We analyzed the output values of the sampler from Fig. 2 (a) and the least significant bit of the counter values from Fig. 2 (b) (and their first differences). We thus analyzed the outputs of four projects.

- The first two projects used the method of entropy extraction based on sampling the jittery clock signal (according to Fig. 2(a)) with two kinds of oscillators used as a source of randomness:
  - signals s_1 and s_2 were generated by two ROs oscillating at 125 and 127 MHz,
  - signals s_1 and s_2 were generated by two STRs oscillating at 130 and 128 MHz.
- The other two projects used the counter method of entropy extraction (according to Fig. 2(b)), while using the same oscillators as the first pair of projects.

For the method of extraction based on sampling of the jittery clock signal, we generated random bit streams for k ranging from 10 000 to 100 000. For the counter method of extraction we generated sequences for k ranging from 2 000 to 100 000. Two kinds of files were generated in this case – one containing the least significant bits of the counter values and the other containing the least significant bit of the first differences of counter values.

We tested all the generated sequences using the AIS31 Procedure B (tests T6 – T8) and NIST 800-90B test suite, from which we also obtained Shannon entropy and min-entropy estimates respectively. The min-entropy was computed for every sequence according to Eq. (26), i.e. the computation was based on high order Markov chains while taking correlations between output bits into account. The results are presented in Appendix D, Table 5 to 10.

Three very important results stand out in the tables presented in Appendix D. First, the method of randomness extraction based on sampling of the jittery clock signal always gives lower entropy rates than those obtained by the method based on counting the jittery clock signal periods. Second, the method of min-entropy estimation based on high order Markov chains gives very consistent results even in the interval of values of k, for which Procedure B of AIS31 revealed no differences in Shannon entropy estimates. Third, the entropy rates are practically the same when the least significant bit of the counter values or that of their differences is used. This is valid independently of the type of free running oscillator (RO or STR).

3 Implementation of the variance measurements in hardware and monitoring of the source of randomness

According to AIS 31 recommendations, the source of randomness should be monitored continuously using dedicated embedded test(s). In our case, this monitoring process would be represented by an online check that the differences of counter values fall within the permitted interval.

To compare the parameters of the proposed randomness monitoring process, we implemented tests based on counter differences and two other state-of-the-art tests (proposed in [HBFT14] and [FL14]) in the same device – Intel Cyclone V FPGA.

The circuitry corresponding to implementation of the Allan variance according to Eq. (13) in hardware is shown in Fig. 10.

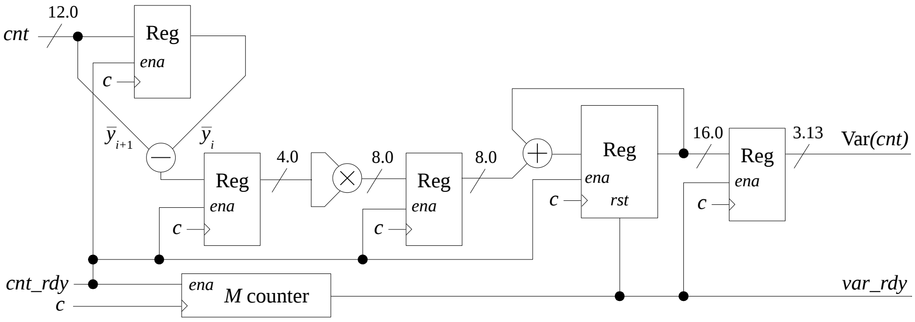

Figure 10: Allan variance measurement circuitry based on Eq. (13)

All the computations are in fixed point arithmetic. This method only requires one multiplier to square data. One subtractor is used to compute the difference of the consecutive samples and one adder with associated register is used as accumulator.

The circuitry corresponding to hardware implementation of the variance computation used by Haddad *et al.* in [HBFT14] and for the one corresponding to Eq. (2) is depicted in Fig. 11.

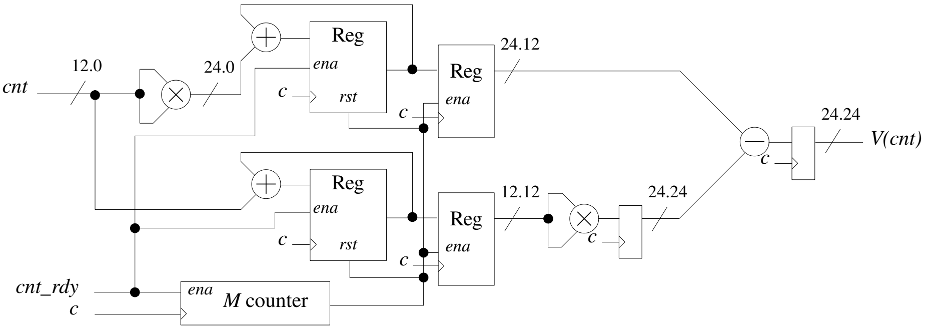

Figure 11: Implementation of the counter variance measurement circuitry for the method proposed by Haddad *et al.* in [HBFT14] and for that corresponding to Eq. (2)

Again, all the computations are in fixed point arithmetic. Numbers before and after the radix point indicate the number of bits of the integer and fractional part of the given value, respectively. Two multipliers (one of 12 bits and the other of 24 bits) are used to square data. Two adders and associated registers (one of 24 bits and the other of 12 bits) are used to implement accumulators. One subtractor is used before the output of the block. Four additional data registers are used to store intermediate data.

The third test we implemented in the hardware has the same architecture as that presented in [FL14], Fig. 6. In the following section, we compare the three implementations.

3.1 Implementation results

First, we evaluate design parameters like area, speed and power consumption of the three methods of variance measurement described above. Area and speed values were obtained from Quartus software. Power consumption was measured using a dedicated hardware evaluation platform [LDFV16]. The results are presented in Table 3.

Table 3: Summary of implementation results of the variance measurement method based on counter differences compared to other state-of-the-art methods implemented in the dedicated evaluation board featuring Intel Cyclone V FPGA device 5CEBA4F17C8N

Method	**Area**		f_{max}	**Power**
	ALM/Regs	DSPs	[MHz]	[mW]
Haddad *et al.* [HBFT14], Eq. (2)	119/160	2	178.3	6-7
Fischer and Lubicz [FL14]	169/200	4	187.7	7-8
Proposed method, Eq. (13)	49/117	1	238.5	4-5

We observe that the Allan variance measurement circuitry based on Eq. (13) is smaller, faster and consumes slightly less power than the circuitry required by the other two methods. This is because the implementation of the Allan variance measurement is simple (only one subtractor and one adder needed, only one DSP block used instead of two or four, respectively).

3.2 Study of the impact of the measurement circuitry on the source of randomness

Next, we propose a rigorous approach to assess the impact of the embedded jitter measurement on the measured jitter itself. The impact of the jitter measurement on the jitter itself is evaluated in the following steps:

- **Project 1** – Only two free running oscillators, used as sources of randomness, are implemented in the selected logic device. The generated clock signals are output using low voltage differential signaling (LVDS) outputs and measured externally using high end oscilloscope and differential probes (see Fig. 12).

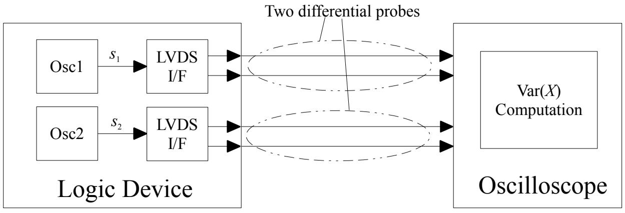

Figure 12: External jitter measurement method using an oscilloscope and differential probes

- **Project 2** – A complete TRNG, embedded variance measurement and an AES cipher are implemented in the FPGA to mimic the behavior of the real crypto SoC as shown in Fig. 13. Signal s_2 is generated using an external quartz oscillator. The variance is measured both internally, and externally.

- **Project 3** – A complete TRNG, embedded variance measurement and an AES cipher are implemented in the FPGA to mimic the behavior of the real crypto SoC as shown in Fig. 13. Signal s_2 is generated using a free running oscillator. The variance is measured both internally, and externally.

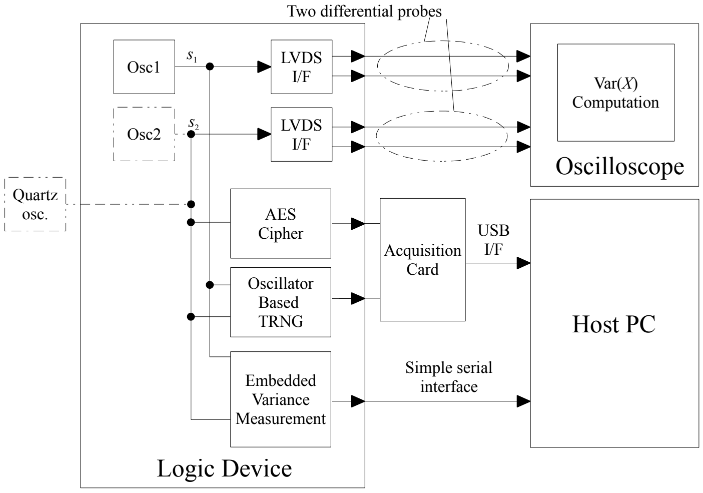

Figure 13: External jitter measurement method using an oscilloscope and differential probes combined with an internal jitter measurement method while the TRNG and the AES cipher are running (only one generator of signal s_2 is present in each of the two projects: Quartz oscillator in Project 2 and $Osc2$ in Project 3)

To ensure the measurement results are consistent, it is important to guarantee the same placement and routing of *Osc1* and *Osc2* in all projects. We generated the Exported Partition file (.qxp), which is the Quartus II software option used to export post-fitting netlists. The exported netlist was then used in all the projects.

We decided to implement only ROs as *Osc1* and *Osc2* because, as shown in Section 2, they are simpler to implement than STRs and the jitter behavior is very similar in both STRs and ROs. Oscillators *Osc1* and *Osc2* had the same number of elements and the same topology. They oscillated at respective frequencies of 124.5 ± 0.3 MHz and 126.3 ± 0.2 MHz. The difference in frequency in the three projects was thus less than 1 %, which was important to ensure the results were comparable.

We measured the jitter of both oscillators as well as the normalized counter value externally using a LeCroy WavePro 735i oscilloscope (4 GHz bandwidth, 40 GS/s) and two D420 WaveLink 4 GHz differential probes. Counter values cannot be obtained directly from an oscilloscope since the value of k cannot be set up like in hardware but can only be deduced from the oscilloscope time base, which, in our case, was set to 5 μs per division. We measured the number of periods of both clocks in this time interval. Finally, to make the comparison of values obtained using the external and embedded measurements more consistent, we measured the number of cycles of both clocks at the same time interval and normalized the resulting data according to the following equation:

$$cnt = \frac{n_1}{n_2} \cdot k, \tag{27}$$

where n_1 represents the number of clock periods of s_1 and n_2 the number of clock periods of s_2 that appear during the same time interval determined by oscilloscope's time base. In our case, we used $k = 30\,000$ to normalize oscilloscope measurements.

Table 4 shows the results of external and internal measurements. The jitters of Osc_1 and Osc_2 were both measured by the oscilloscope. To compare these values with those obtained using the Allan variance according to Eq.(25), we saved the counter values in a file for processing.

Table 4: Results of external and internal measurements of oscillator jitters in three selected projects: Columns 2 and 3 list the jitters σ_1, σ_2 measured using the oscilloscope. Column 4 lists the equivalent jitter σ_{eq} computed from Eq. (18). Column 5 lists the normalized variance of counter values computed from the oscilloscope using Eq. (27). Column 6 lists the Allan variance estimate computed inside the device using Eq. (13).

Project	σ_1	σ_2	σ_{eq} [ps]	$Var(cnt)$	Avar(N)	Avar(τ) [**s**^{2}]
	[ps]	[ps]	from Eq.(18)	from Eq.(27)	from Eq.(13)	from Eq.(25)
Project 1	3.9	3.3	5.1	14.01	2.79	2.0425e-16
Project 2	9.7	7.3	11.8	26.94	4.33	3.2438e-16
Project 3	10.6	10.0	14.5	14.72	2.76	2.0216e-16

We can see that putting the whole cryptosystem including the AES cipher in an FPGA more than doubles the jitter of both oscillators, but the variance of counter values remains almost the same if only internal oscillators are used. In Project 2, in which the signal s_2 is generated by an external quartz oscillator, there was a significant increase in the variance of counter values, which confirms that using identically implemented oscillators and implementing them both inside the FPGA (differential principle of randomness extraction) helps prevent negative effects of the surrounding logic on the measured jitter.

To further confirm this claim, we acquired a large sequence of counter values from Project 2 and transferred them to a PC in order to visualize them over time. The acquisition was done with the accumulation period set by $k = 30\,000$. The whole acquisition took approximately 30 minutes. Figure 14 shows the counter values when the signal s_2 was generated by an external quartz oscillator. A strong low frequency signal can be seen to affect the counter values. The frequency of the signal is approximately 1.5 mHz.

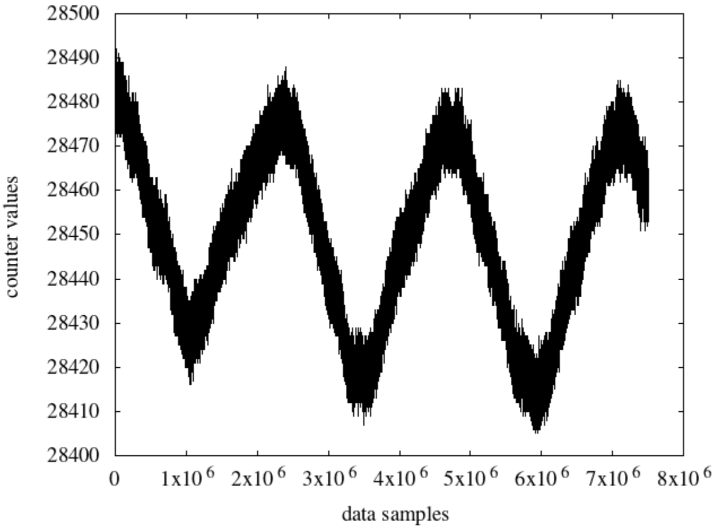

Figure 14: Counter values acquired using an external oscillator for s_2

Figure 15 shows the counter values when s_2 was generated by an internal RO. Even though the low frequency pattern is still slightly visible, its amplitude is significantly reduced.

We discovered that the observed low frequency signal originated from the power line even though the evaluation board was using only low noise linear power supplies. These findings confirm that unwanted global noises are almost always present and are unavoidable. Since this kind of noise can be manipulated, it can be extremely dangerous for the TRNG

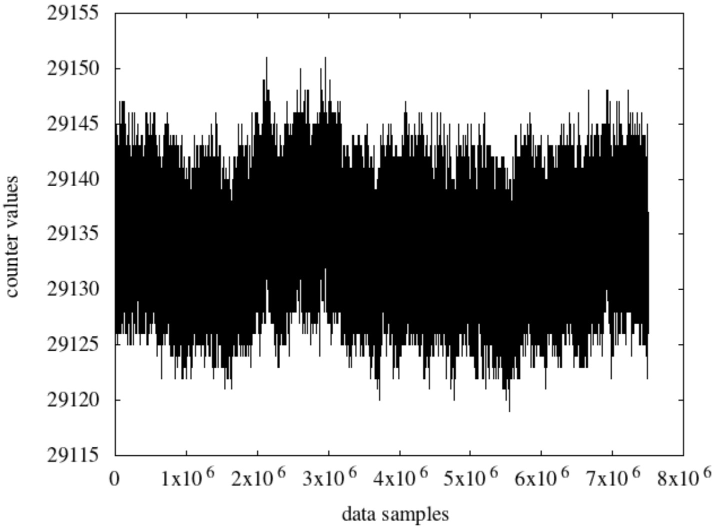

Figure 15: Counter values acquired when both oscillators are implemented in the FPGA

design. Moreover, a low frequency signal such as the one visible in Fig. 14 is usually hard to detect.

4 Discussion

We have very clearly demonstrated several advantages of the Allan variance over statistical variance: it gives stable values independent of low frequency noises even for short data sets. It is thus suitable for the estimation of entropy originating from non-manipulable independent noises such as thermal noise. It can serve as a basis for embedded tests, for which it is particularly suited because of its small area and low latency.

RO and STR behave similarly in terms of variance dependence on jitter accumulation time. Jitter accumulates in both structures in a very similar way. This is a new observation.

Using two identical oscillators reduces autocorrelations in RNG output values. Using the first differences of counter values instead of counter values themselves further reduces autocorrelations.

The method of randomness extraction based on sampling of the jittery clock signal always gives lower quality results than the method based on counting the jittery clock signal periods. The jitter accumulation times can be reduced more than ten times (more than 400 000 periods of the reference clock were needed in [FL14] and fewer than 30 000 if the jittery clock periods are counted). This means significantly higher bit rates at generator output with no loss of entropy.

The method of min-entropy estimation based on high order Markov chains gives very consistent results even in the interval of values of k, for which Procedure B of AIS 31 revealed no differences in Shannon entropy estimates (see Tables 7 to 10).

The studies described here confirm, that using external oscillators jeopardizes the implementation of security critical applications. They also prove, that implementing identical oscillators inside the FPGA and using their relative jitter transformed into counter values or even better into their differences, can efficiently mitigate the negative effects of global noise sources both external to the FPGA and generated internally by the surrounding logic, represented in our case by the AES cipher.

5 Conclusions

We evaluated the jitter of clock signals generated in ring oscillators and self timed rings and the way the jitter is transformed into random numbers. We showed that counting the periods of the jittery clock signal gives random numbers of significantly better quality than the usual methods of sampling jittery clock signals. We used counter values to characterize and to continuously monitor the source of randomness. We showed that using the Allan variance to characterize the clock jitter has at least two advantages: first, it is not sensitive to low frequency noises such as flicker noise, and second, significantly less circuitry is required for its computation than that used in other methods. We also show that a differential principle of randomness extraction from the jitter, based on the use of two identical oscillators is essential to avoid autocorrelations originating from both the external and internal sources of global jitter, independently of the type of ring used. Last but not least, we propose a new method of statistical testing based on a high order Markov model to demonstrate the reduction of dependencies when the proposed randomness extraction is applied. While providing an estimation of min-entropy, the method is very efficient in detecting dependencies between generated numbers.

Acknowledgments

This work received funding from the European Union's Horizon 2020 research and innovation programme in the framework of the project HECTOR (Hardware Enabled Crypto and Randomness) under grant agreement No 644052. Maciej Skorski also acknowledges funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 682815/TOCNeT).

References

[AB81] D. W. Allan and J. A. Barnes. A Modified "Allan Variance" with Increased Oscillator Characterization Ability. In *Proceedings of 35th Annual Frequency Control Symposium*, pages 470–475, 1981.

[All66] D. W. Allan. Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221–230, 1966.

[All87] D. W. Allan. Should the classical variance be used as a basic measure in standards metrology? *IEEE Transactions on Instrumentation and Measurement*, 1001(2):646–654, 1987.

[Bar46] M. S. Bartlett. On the theoretical specification and sampling properties of autocorrelated time-series. *Supplement to the Journal of the Royal Statistical Society*, 8(1):27–41, 1946.

[BLMT11] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux. On the security of oscillator-based random number generators. *Journal of Cryptology*, 24(2):398–425, 2011.

[CBFF12] C. Costea, F. Bernard, V. Fischer, and R. Fouquet. Implementation of ring oscillators based physical unclonable functions with independent bits in the response. *International Journal of Reconfigurable Computing*, ID 168961, 2012.

[CFAF13] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet. A self-timed ring based true random number generator. In *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC 2013)*, pages 99–106, 2013.

[CFFA13] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert. A very high speed true random number generator with entropy assessment. In G. Bertoni and J.-S. Coron, editors, *Cryptographic Hardware and Embedded Systems (CHES 2013)*, volume 8086 of *LNCS*, pages 179–196. Springer, 2013.

[CM98] George R. Cooper and Clare D. McGillem. *Probabilistic Methods of Signal and System Analysis.* Oxford University Press, 3rd edition, 1998.

[CN14] A. D. R. Choudary and Constantin P. Niculescu. *Real Analysis on Intervals: Improper Riemann Integrals.* Springer India, New Delhi, 2014.

[DDS18] N. Da Dalt and A. Sheikholeslami. *Understanding Jitter and Phase Noise.* Cambridge University Press, 2018.

[Eng06] Shlomo Engelberg. *Random signals and noise: a mathematical introduction.* CRC Press, 2006.

[FL14] V. Fischer and D. Lubicz. Embedded evaluation of randomness in oscillator based elementary trng. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems (CHES 2014)*, volume 8731 of *LNCS*, pages 527–543. Springer, 2014.

[Gre83] Charles Greenhall. A Structure Function Representation Theorem with Applications to Frequency Stability Estimation. *IEEE Transactions on Instrumentation and Measurement*, 32(2):364 – 370, 1983.

[HBFT14] P. Haddad, F. Bernard, V. Fischer, and Y. Teglia. On the Assumption of Mutual Independence of Jitter Realizations in P-TRNG Stochastic Models. In *Design, Automation and Test in Europe (DATE 2014), Dresden, Germany.* IEEE, 2014.

[KG04] P. Kohlbrenner and K. Gaj. An embedded true random number generator for FPGAs. In *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, pages 71–78. ACM, 2004.

[KS08] W. Killmann and W. Schindler. A Design for a Physical RNG with Robust Entropy Estimators. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2008*, volume 5154 of *LNCS*, pages 146–163. Springer, 2008.

[KS11] W. Killmann and W. Schindler. A proposal for: Functionality classes for random number generators, version 2.0. [online] Available from https://www.bsi.bund.de/EN/TheBSI/thebsi_node.html, 2011.

[KV16] Sudeep Kamath and Sergio Verdu. Estimation of entropy rate and Renyi entropy rate for Markov chains. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 685–689, 2016.

[LDFV16] M. Laban, M. Drutarovsky, V. Fischer, and M. Varchola. Platform for testing and evaluation of PUF and TRNG implementations in FPGAs. In *TRUDEVICE – 6th Conference on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE 2016), Barcelona, Spain*, November 2016.

[LG08] Alberto Leon-Garcia. *Probability, Statistics, and Random Processes for Electrical Engineering.* Pearson/Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2008.

[Nis16] Y. Nishio. *Oscillator Circuits: Frontiers in Design, Analysis and Applications*. IET, 1st edition, 2016.

[PP01] A Papoulis and Unnikrishna Pillai. *Probability, random variables and stochastic processes*. McGraw-Hill, 4th edition, Nov. 2001.

[Ril08] W. Riley. *Handbook of Frequency Stability Analysis*. NIST, 2008.

[RYDV15] V. Rozic, B. Yang, W. Dehaene, and I. Verbauwhede. Highly efficient entropy extraction for true random number generators on FPGAs. In *Proceedings of the 52nd Annual Design Automation Conference (DAC), San Francisco, USA*, pages 116:1–116:6, 2015.

[She97] W. F. Sheppard. On the Calculation of the most Probable Values of Frequency-Constants, for Data arranged according to Equidistant Division of a Scale. *Proceedings of the London Mathematical Society*, s1-29(1):353–380, 1897.

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

[Uhr07] Pierre Uhrich. Stabilité des oscillateurs ultra-stables. *Cours X-ENS*, 2007.

[VABF10] B. Valtchanov, A. Aubert, F. Bernard, and V. Fischer. Characterization of randomness sources in ring oscillator-based true random number generators in FPGAs. *Design and Diagnostics of Electronic Circuits and Systems, 2008. DDECS 2010. 13th IEEE Workshop on*, pages 1–6, 2010.

[Žit10] Gordan Žitković. Introduction to stochastic processes - lecture notes. https://www.ma.utexas.edu/users/gordanz/notes/introduction_to_stochastic_processes.pdf, December 2010. accessed 17/01/2018.

A Proofs of the Allan variance properties

A.1 Allan variance generalizes the statistical variance

If x is a stationary and uncorrelated random process, we know that its statistical variance exists [All87]. If we call μ the expected value of x, then:

$$\begin{aligned}
\mathrm{Avar}(x) &= \frac{1}{2}\,\mathbb{E}\left[(x_{i+1}-x_i)^2\right] \\
&= \frac{1}{2}\,\mathbb{E}\left(\left[(x_{i+1}-\mu)-(x_i-\mu)\right]^2\right) \\
&= \frac{1}{2}\,\mathbb{E}\left[(x_{i+1}-\mu)^2-(x_{i+1}-\mu)(x_i-\mu)+(x_i-\mu)^2\right] \\
&= \frac{1}{2}\,\mathbb{E}\left[(x_{i+1}-\mu)^2\right]-\frac{1}{2}\,\mathbb{E}\left[(x_{i+1}-\mu)(x_i-\mu)\right]+\frac{1}{2}\,\mathbb{E}\left[(x_i-\mu)^2\right] \\
&= \frac{1}{2}\,\mathrm{Var}(x_{i+1})-\frac{1}{2}\,\mathbb{E}\left[(x_{i+1}-\mu)(x_i-\mu)\right]+\frac{1}{2}\,\mathrm{Var}(x_i). \qquad (28)
\end{aligned}$$

Since x is stationary, one has:

$$\mathrm{Var}(x_{i+1}) \;=\; \mathrm{Var}(x_i) \;=\; \mathrm{Var}(x) \qquad (29)$$

and:

$$\mathbb{E}\left(x_{i+1}\right) \;=\; \mathbb{E}\left(x_i\right) \;=\; \mu. \qquad (30)$$

Moreover, the uncorrelatedness of x implies:

$$\mathbb{E}\left(x_{i+1}x_i\right) \;=\; \mathbb{E}\left(x_{i+1}\right)\mathbb{E}\left(x_i\right). \qquad (31)$$

It then follows:

$$\begin{aligned}
\mathbb{E}\left[(x_{i+1}-\mu)(x_i-\mu)\right] &= \mathbb{E}\left(x_{i+1}x_i-\mu x_{i+1}-\mu x_i+\mu^2\right) \\
&= \mathbb{E}\left(x_{i+1}x_i\right)-\mu\,\mathbb{E}\left(x_{i+1}\right)-\mu\,\mathbb{E}\left(x_i\right)+\mu^2 \\
&= 0. \qquad (32)
\end{aligned}$$

Hence:

$$\mathrm{Avar}(x) \;=\; \mathrm{Var}(x). \qquad (33)$$

A.2 Multiplication by a scalar

Given a real number λ and a stationary random process x, then λx is also a stationary random process. Its Allan variance is then:

$$\mathrm{Avar}(\lambda x) \;=\; \frac{1}{2}\,\mathbb{E}\left[(\lambda x_{i+1}-\lambda x_i)^2\right] \;=\; \lambda^2\frac{1}{2}\,\mathbb{E}\left[(x_{i+1}-x_i)^2\right] \;=\; \lambda^2\,\mathrm{Avar}(x). \qquad (34)$$

A.3 Sum of independent random processes

If x and y are two independent stationary random processes, one has:

$$\begin{aligned}
\mathrm{Avar}(x+y) &= \frac{1}{2}\,\mathbb{E}\left[\left(x_{i+1}+y_{i+1}-x_i-y_i\right)^2\right] \\
&= \frac{1}{2}\,\mathbb{E}\left[\left(x_{i+1}-x_i+y_{i+1}-y_i\right)^2\right] \\
&= \frac{1}{2}\,\mathbb{E}\left[\left(x_{i+1}-x_i\right)^2+\left(x_{i+1}-x_i\right)\left(y_{i+1}-y_i\right)+\left(y_{i+1}-y_i\right)^2\right] \\
&= \frac{1}{2}\,\mathbb{E}\left[\left(x_{i+1}-x_i\right)^2\right]+\frac{1}{2}\,\mathbb{E}\left[\left(x_{i+1}-x_i\right)\left(y_{i+1}-y_i\right)\right]+\frac{1}{2}\,\mathbb{E}\left[\left(y_{i+1}-y_i\right)^2\right] \\
&= \mathrm{Avar}(x)+\frac{1}{2}\,\mathbb{E}\left[\left(x_{i+1}-x_i\right)\left(y_{i+1}-y_i\right)\right]+\mathrm{Avar}(y). \qquad (35)
\end{aligned}$$

Because the processes x and y are independent, one has:

$$\mathbb{E}\left(x_i y_j\right) = \mathbb{E}\left(x_i\right)\mathbb{E}\left(y_j\right), \qquad (36)$$

for any $i, j \in \mathbb{N}$. Since they are stationary:

$$\mathbb{E}\left(x_j\right) = \mathbb{E}\left(x_i\right) = \mathbb{E}\left(x\right) \quad \text{and} \quad \mathbb{E}\left(y_j\right) = \mathbb{E}\left(y_i\right) = \mathbb{E}\left(y\right),$$

for any $i, j \in \mathbb{N}$. Hence:

$$\begin{aligned}
\mathbb{E}\left[\left(x_{i+1}-x_i\right)\left(y_{i+1}-y_i\right)\right] &= \mathbb{E}\left[x_{i+1}y_{i+1}-x_{i+1}y_i-x_i y_{i+1}+x_i y_i\right] \\
&= \mathbb{E}\left[x_{i+1}y_{i+1}\right]-\mathbb{E}\left[x_{i+1}y_i\right]-\mathbb{E}\left[x_i y_{i+1}\right]+\mathbb{E}\left[x_i y_i\right] \\
&= \mathbb{E}\left[x_{i+1}\right]\mathbb{E}\left[y_{i+1}\right]-\mathbb{E}\left[x_{i+1}\right]\mathbb{E}\left[y_i\right]-\mathbb{E}\left[x_i\right]\mathbb{E}\left[y_{i+1}\right] \\
&\quad +\mathbb{E}\left[x_i\right]\mathbb{E}\left[y_i\right] \\
&= \mathbb{E}\left[x\right]\mathbb{E}\left[y\right]-\mathbb{E}\left[x\right]\mathbb{E}\left[y\right]-\mathbb{E}\left[x\right]\mathbb{E}\left[y\right]+\mathbb{E}\left[x\right]\mathbb{E}\left[y\right] \\
&= 0. \qquad (37)
\end{aligned}$$

It then follows that:

$$\mathrm{Avar}(x+y) = \mathrm{Avar}(x)+\mathrm{Avar}(y). \qquad (38)$$

B Autocorrelations

B.1 Background

Sample autocorrelation Given a sequence of observations $z_1, \ldots, z_N$ originating from a random process $\{Z_i\}_{i=1}^N$, the *sample autocorrelation* is the function of the time lag τ defined by

$$\hat{\rho}_u(\tau) = \frac{\frac{1}{N-\tau}\sum_{i=1}^{N-\tau}\sum(z_{i+\tau}-\hat{\mu})(z_i-\hat{\mu})}{\hat{\sigma}^2} \qquad (39)$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ are sample mean and variance estimates

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} z_i$$

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(z_i-\hat{\mu})^2$$

At longer lags $\tau \approx N$ there are fewer samples to estimate, so that $\hat{\rho}_u$ becomes unstable; for this reason one often applies the following modification

$$\hat{\rho}_b(\tau) = \frac{\frac{1}{N}\sum_{i=1}^{N-\tau}\sum(z_{i+\tau}-\hat{\mu})(z_i-\hat{\mu})}{\hat{\sigma}^2} \tag{40}$$

which increases the bias but has lower variance (and smaller MSE error as suggested in some empirical studies); however, in our case N is big enough to obtain accurate results of $\hat{\rho}_u(\tau)$ for a wide range of values $0 \ll \tau \ll N$.

Process autocorrelation If the sample $z_1,\ldots,z_N$ comes from a WSS ergodic process $\{Z_i\}_i$ then $\hat{\rho}$ and $\hat{\rho}_b$ estimate the *process autocorrelation function*

$$\rho_Z(\tau) = \frac{\mathbb{E}(Z_{t+\tau}-\mathbb{E}Z_{t+\tau})(Z_i-\mathbb{E}Z_t)}{\sqrt{\mathrm{Var}(Z_{t+\tau})}\sqrt{\mathrm{Var}(Z_t)}}$$

which under the WSS assumption depends only on τ (as the mean $\mathbb{E}(Z_i)=\mu$ and variance $\sigma^2 = \mathrm{Var}(Z_i)$ do not depend on i).

Sample vs. process autocorrelation If the sample $z_1,\ldots,z_N$ comes from a WSS ergodic process $\{Z_i\}_i$ then $\hat{\rho}$ and $\hat{\rho}_b$ estimate the process autocorrelation. This estimate converges provided the autocorrelations decay fast enough (in theoretical literature this is captured by the notion of covariance ergodicity [PP01]). Confidence for these estimates, when necessary, can be obtained using Bartlet's formula [Bar46].

B.2 Examples

Raw counter values We first estimate autocorrelations of counter values. As expected they are very high, particularly for setups with a quartz reference clock. We use both estimators (40) and (39). We compute the sample autocorrelation function by fast Fourier transform.

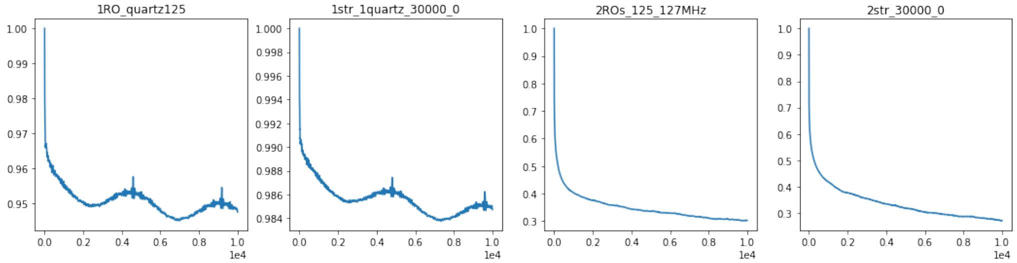

Figure 16: Autocorrelations of counter values, small lags (bias-corrected estimate $\hat{\rho}_u$)

Differences of counter values Next, we estimate autocorrelations for the counter differences. The autocorrelations are significantly reduced.

C Proof of Lemma 1

Since the process is Gaussian and the variance and mean of Z_i do not change over time, higher moments do not change over time either. We therefore have

$$\mathbb{E}\left(Z_i^2 - \mathbb{E}(Z_i^2)\right)^2 = O(1) \tag{41}$$

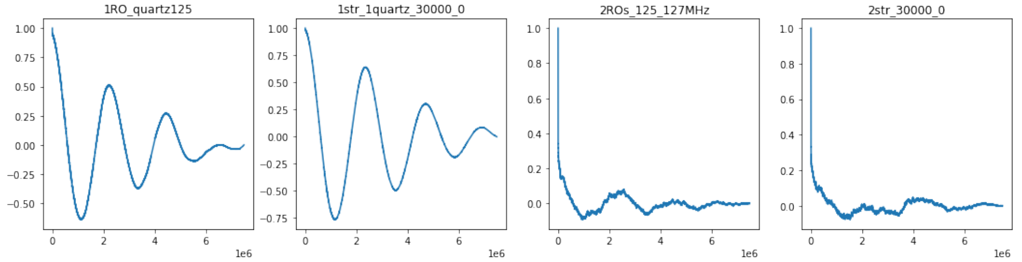

Figure 17: Autocorrelations of counter values, including large lags (variance-stable estimate $\hat{\rho}_b$)

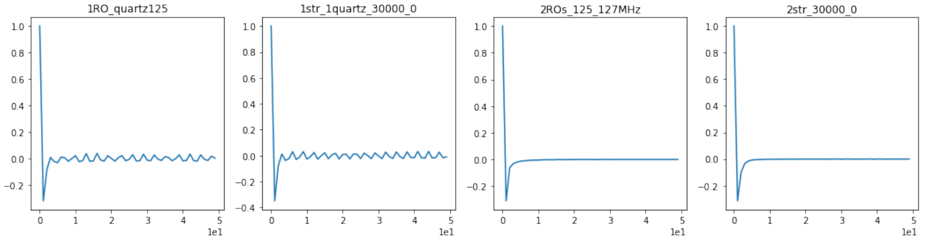

Figure 18: Autocorrelations of counter values differences (bias-corrected estimate $\hat{\rho}_u$)

where the constant does not depend on time i. Now let us consider mixed moments

$$\mathbb{E}\left(Z_i^2-\mathbb{E}(Z_i^2)\right)\left(Z_j^2-\mathbb{E}(Z_j^2)\right)=\mathbb{E}(Z_i^2Z_j^2)-\mathbb{E}(Z_i^2)\mathbb{E}(Z_j^2)$$

which, for the joint Gaussian distribution (Z_i, Z_j) can be simplified[5] as

$$\begin{aligned}\mathbb{E}(Z_i^2Z_j^2)-\mathbb{E}(Z_i^2)\mathbb{E}(Z_j^2)&=2\cdot\mathrm{Cov}(Z_i,Z_j)^2\\&=2\left(\rho(Z_i,Z_j)\mathrm{Var}(Z_i)\mathrm{Var}(Z_j)\right)^2\end{aligned}$$

where $\rho(Z_i, Z_j)$ is the correlation. Again $\mathrm{Var}(Z_i)$ does not change over time; moreover $\rho(Z_i, Z_j)$ depends only on the lag $j-i$ and equals zero when $|i-j| > p$ according to our assumptions. Thus

$$\mathbb{E}\left(Z_i^2-\mathbb{E}(Z_i^2)\right)\left(Z_j^2-\mathbb{E}(Z_j^2)\right)=\left\{\begin{array}{rl}O(1) & |i-j|\leqslant p\\ 0 & |i-j|>p\end{array}\right. \tag{42}$$

where the constant does not depend on i, j. By combining Equations (41) and (42) we obtain

$$\begin{aligned}\mathrm{Var}\left(\frac{1}{M}\sum_{i=1}^{M}Z_i^2\right)&=\frac{1}{M^2}\left(\sum\mathbb{E}\left(Z_i^2-\mathbb{E}(Z_i^2)\right)^2+\sum_{i\neq j}\mathbb{E}\left(Z_i^2-\mathbb{E}(Z_i^2)\right)\left(Z_j^2-\mathbb{E}(Z_j^2)\right)\right)\\&=O(p/M).\end{aligned}\tag{43}$$

[5] We use the well-known formulas for central 4th-order moments of multivariate Gaussians.

D Results of entropy estimation using a Markov chain model and statistical tests required by standards

Entropy was evaluated in six different configurations of the TRNG including two methods of randomness extraction and two types of oscillators, as explained in Sect. 1:

- Both clock signals generated by ROs, randomness extraction by sampling the clock.
- Both clock signals generated by STRs, randomness extraction by sampling the clock.
- Both clock signals generated by ROs, randomness extraction by counting the clock edges (the least significant bit of the counter represented the random bit).
- Both clock signals generated by ROs, randomness extraction by counting the edges.
- Both clock signals generated by STRs, randomness extraction by counting the edges.
- Both clock signals generated by STRs, randomness extraction by counting the edges.

We used two standardized batteries of statistical tests alongside the method proposed in this article to evaluate the output of the TRNGs:

- **Markov chain min-entropy estimate.** This method is explained in detail in Section 2.
- German AIS 20/31 test suite from Procedure B, which is intended to test the output of the TRNG core. Entropy is estimated by the test T8 is the Shannon entropy per random bit.
- American NIST 800-90B test suites for independent and identically distributed data (IID) and non-IID data. If data is detected to be IID, the min-entropy estimate of the IID test track is given. Otherwise, the non-IID entropy estimate is used.

Table 5: Entropy estimation using high order Markov chains, AIS 31 and NIST 800-90B tests, when two internal ROs and the sampling method was used. Dependencies are modeled using 8$^{\text{th}}$ order Markov chains.

τ	Markov chain	AIS 31 Procedure B	AIS 31 T8	NIST 800-90B	NIST 800-90B
periods of s_2	min-entropy per bit		Shannon entropy per bit	IID	min-entropy per bit
10 000	0.8102	failed	0.9844	non-IID	0.648
20 000	0.8105	failed	0.9851	non-IID	0.647
30 000	0.8102	failed	0.9847	non-IID	0.648
50 000	0.9369	failed	0.9992	non-IID	0.673
100 000	0.9012	failed	0.9935	non-IID	0.670

Table 6: Entropy estimation using high order Markov chains, AIS 31 and NIST 800-90B tests, when two internal STRs and the sampling method was used. Dependencies are modeled using 8th order Markov chains.

τ	Markov chain	AIS 31 Procedure B	AIS 31 T8	NIST 800-90B	NIST 800-90B
periods of s_2	min-entropy per bit		Shannon entropy per bit	IID	min-entropy per bit
10 000	0.5440	failed	0.9072	non-IID	0.489
20 000	0.5435	failed	0.9074	non-IID	0.489
30 000	0.5425	failed	0.9021	non-IID	0.489
50 000	0.5432	failed	0.9030	non-IID	0.489
100 000	0.5423	failed	0.9076	non-IID	0.489

Table 7: Entropy estimation using high order Markov chains, AIS 31 and NIST 800-90B tests, when two internal ROs and the least significant bits of counter values were used. Dependencies are modeled using 8th order Markov chains.

τ	Markov chain	AIS 31 Procedure B	AIS 31 T8	NIST 800-90B	NIST 800-90B
periods of s_2	min-entropy per bit		Shannon entropy per bit	IID	min-entropy per bit
2 000	0.2939	failed	0.0910	non-IID	0.621
10 000	0.8089	failed	0.9966	non-IID	0.844
15 000	0.9769	passed	0.9998	non-IID	0.931
15 000	0.9769	passed	0.9998	non-IID	0.931
20 000	0.9865	passed	0.9999	IID	0.999
25 000	0.9907	passed	0.9999	IID	0.998
100 000	0.9910	passed	0.9999	IID	0.998

Table 8: Entropy estimation using high order Markov chains, AIS 31 and NIST 800-90B tests, when two internal ROs and the least significant bits of the first differences of counter values were used. Dependencies are modeled using 8th order Markov chains.

τ	Markov chain	AIS 31 Procedure B	AIS 31 T8	NIST 800-90B	NIST 800-90B
periods of s_2	min-entropy per bit		Shannon entropy per bit	IID	min-entropy per bit
2 000	0.2816	failed	0.0981	non-IID	0.336
10 000	0.8087	failed	0.9865	non-IID	0.661
15 000	0.9783	passed	0.9998	non-IID	0.876
20 000	0.9893	passed	0.9999	IID	0.999
25 000	0.9908	passed	0.9999	IID	0.999
100 000	0.9909	passed	0.9999	IID	0.998

Table 9: Entropy estimation using high order Markov chains, AIS 31 and NIST 800-90B tests, when two internal STRs and the least significant bits of counter values were used. Dependencies are modeled using 8th order Markov chains.

τ	Markov chain	AIS 31 Procedure B	AIS 31 T8	NIST 800-90B	NIST 800-90B
periods of s_2	min-entropy per bit		Shannon entropy per bit	IID	min-entropy per bit
2 000	0.3331	failed	0.0999	non-IID	0.565
10 000	0.8535	failed	0.9966	non-IID	0.849
15 000	0.9871	passed	0.9999	IID	0.998
20 000	0.9788	passed	0.9999	IID	0.999
25 000	0.9915	passed	0.9996	IID	0.999
100 000	0.9807	passed	0.9999	IID	0.998

Table 10: Entropy estimation using high order Markov chains, AIS 31 and NIST 800-90B tests, when two internal STRs and the least significant bits of the first differences of counter values were used. Dependencies are modeled using 8th order Markov chains.

τ	Markov chain	AIS 31 Procedure B	AIS 31 T8	NIST 800-90B	NIST 800-90B
periods of s_2	min-entropy per bit		Shannon entropy per bit	IID	min-entropy per bit
2 000	0.3264	failed	0.0997	non-IID	0.346
10 000	0.8463	failed	0.9979	non-IID	0.672
15 000	0.9883	passed	0.9999	non-IID	0.897
20 000	0.9924	passed	0.9999	IID	0.999
25 000	0.9915	passed	0.9998	IID	0.999
100 000	0.9835	passed	0.9999	IID	0.998