Embedded Evaluation of Randomness in Oscillator Based Elementary TRNG

Viktor Fischer, David Lubicz

Embedded Evaluation of Randomness in Oscillator Based Elementary TRNG

Viktor Fischer[1] and David Lubicz[2,3]

[1] Laboratoire Hubert Curien, Université Jean Monnet, Université de Lyon, F-42000 Saint-Etienne, France
[2] DGA-Maîtrise de l'information, BP 7419, F-35174 Bruz, France
[3] Intitut de Mathématiques de Rennes, Université de Rennes 1, Campus de Beaulieu, F-35042 Rennes, France

Abstract. Jittery clock signals produced in oscillators, particularly in ring oscillators are commonly used as a source of randomness in true random number generators (TRNG). The robustness of the generators, and hence their security, is closely linked to the entropy of the generated bit stream, which depends on the size of the jitter. Known jitter size can be used as an input parameter in a stochastic model for the estimation of entropy. Good entropy management can guarantee the security of the generator. We propose a simple precise method for measuring jitter that can be easily embedded in logic devices. It can be used to calibrate an oscillator based TRNG and/or for assessment of the entropy rate while the TRNG is in operation. The method was thoroughly evaluated in simulations and hardware tests and we show that despite its simplicity and small area requirements, it enables the jitter to be measured with an error of less than 5 %. [1]

Keywords: hardware random number generators, ring oscillators, jitter model, entropy, statistical tests.

1 Introduction

Random numbers play a crucial role in modern cryptography: they are used as confidential keys, initialization vectors, padding values, and also as random masks in side-channel attack countermeasures. Since the era of Kerckhoff, cryptographic algorithms have been designed to be secure so that even if their principle is known by adversaries, useful information cannot be accessed without knowledge of the secret key. The security of modern cryptographic systems using approved cryptographic algorithms is thus based on the confidentiality of the cryptographic keys generated in random number generators. If the secret key is compromised, the whole cryptographic system may be compromised.

This is why random number generators have attracted the attention of researchers, especially in last two decades. Nevertheless, designing a good true random number generator (TRNG) that can be easily implemented in logic devices is still a challenge, mainly

because digital integrated circuits offer only a limited choice of sources of randomness, such as clock jitter [14], metastability [19], oscillatory metastability [18], write collisions in dual-port RAMs [7] or random initialization of a bi-stable circuit [16]. Furthermore, most of these sources are very sensitive to variations in environmental conditions. This makes even a seamlessly good TRNG vulnerable to attacks [12].

Although some published designs were said to be provably secure, it turned out that they cannot resist some active attacks [2]. Instead of relying on the robustness of the proposed principles, designers should thus propose efficient, on-line tests that are capable of rapidly detecting any deviation from normal behavior. Unfortunately, high quality standard statistical tests [13] are too slow and too expensive.

The aim of this paper is to provide a simple efficient way to evaluate the source of randomness directly in the device and to estimate on-line the entropy of the generated signal in a dedicated and consequently efficient and rapid statistical test.

Very few methods of the embedded measurement of the clock jitter as a source of randomness were published up to now. Moreover, they are complex and not aimed for cryptography [20] or they cannot distinguish the jitter coming from the thermal noise from that coming from the flicker noise that is known to be autocorrelated [17].

Our contribution

1. We propose an original, simple, precise method of jitter measurement that can be implemented inside logic devices.
2. We demonstrate that together with a suitable statistical model (e. g. [1]), the measured jitter can be used to estimate entropy at the output of the generator.
3. We show that the proposed entropy estimator can serve as a basis for a rapid on-line dedicated statistical test, that is perfectly adapted to the generator's principle. This approach complies with recent recommendations for evaluation of TRNGs [10].

Organization of the paper: in Section 2, we discuss basic security requirements for random number generators in cryptography. In Section 3, we describe an elementary oscillator-based random number generator and its characteristics. Section 4 is dedicated to the new randomness evaluation method, which is then evaluated by simulations in Section 5. In Section 6 we describe the implementation of the method in hardware. We discuss our results in Section 7 and in Section 8 we draw some conclusions.

2 Security Requirements on RNGs in Cryptography

Security of a TRNG design must be thoroughly evaluated [5]. Namely, two security requirements must be fulfilled:

- *The statistical quality of generated numbers* guarantees that attacks can only succeed by using an exhaustive search for the secret.
- *Unpredictability* means that even knowing the last generator's output, no other output can be predicted with non-negligible probability in a forward or backward direction.

While the statistical quality of the generated numbers is relatively easy to verify, evaluating unpredictability is not straightforward, since it cannot be measured or tested. The entropy (and thus unpredictability) can only be estimated using a *stochastic model*.

A perfect generator should be *robust against environmental fluctuations, aging and attacks*. In practice, perfect and permanent robustness against attacks and manipulations cannot be reached. Even a generator that is robust to all known attacks may be vulnerable to new attacks in the future. The only way to ensure long term resistance against attacks is to *execute permanently dedicated on-line tests* able to detect, quickly and reliably, even temporary reduction of the entropy rate. Embedded tests must be based on existing stochastic model having, as an input parameter, the size of the physical phenomenon that is used as a source of entropy (e. g. the clock jitter).

We can conclude that permanent evaluation of the entropy contents of the raw binary signal, which is the main objective of this paper, will ensure all security requirements are respected.

3 Elementary Oscillator-Based Random Number Generator

In this section, we present a structure called an elementary oscillator-based TRNG (EO TRNG). This structure is useful for several reasons: (1) it is simple enough so that a comprehensive and relatively simple statistical model can be created (see [1]); (2) it can be used as a basic building block for almost an entire class of oscillator-based TRNGs; (3) it can be used as a construction element for a scalable TRNG.

3.1 Definition of the Elementary Oscillator-Based TRNG

An elementary oscillator-based TRNG is composed of two oscillators, Osc_i for $i = 1, 2$. The output of one oscillator is used to determine the instants of sampling the output of the second one in a sampling unit, e. g. a synchronous D flip-flop (see Figure 1). The frequency of the sampling oscillator is divided by K_D. The division factor K_D makes it possible to determine the time interval needed to accumulate the phase jitter to a sufficient extent, to ensure a suitable entropy rate in the TRNG output bit stream. In the rest of the paper, we suppose that Osc_1 is the oscillator generating the sampled signal and that oscillator Osc_2 generates the sampling clock signal.

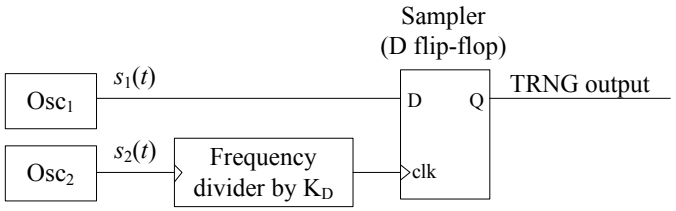


Fig. 1. *Structure of an oscillator-based elementary TRNG*

For $i = 1, 2$, the output signal of Osc_i is given by a periodic function of time t that takes the form

$$s_i(t) = f(\omega_i(t + \xi_i(t))), \tag{1}$$

where f can generally be any real valued function with period 1. In our case, we suppose that we are dealing with TRNG implementation in logic devices and therefore for $\alpha \in [0, 1)$, we define f_α as a specific real valued 1-periodic function such that $f_\alpha(x) = 1$ for all $0 < x < \alpha$ and $f_\alpha(x) = 0$ for $\alpha < x < 1$, and $f_\alpha(0) = f_\alpha(\alpha) = 1/2$. We use f_α as a convenient model for the digital clock signal produced by a clock generator and in particular by a ring oscillator. Note that the clock edge is not necessarily in the middle of the interval $[0, 1)$, since oscillators can often have imbalanced half periods. We do not consider amplitude fluctuations in our model since their contribution to phase jitter is negligible in clock signal generation as explained in [11, p. 134].

In practice, we accept that the frequencies of both signals $s_i(t)$, $i = 1, 2$, fluctuate. Therefore, ω_i is the *mean frequency* of the signal $s_i(t)$, $(\omega_i(t + \xi_i(t)))$ is the *phase* of the oscillator and the function $\xi_i(t)$ represents the *absolute phase drift*. Similarly, $T_i = 1/\omega_i$ is the mean period of $s_i(t)$. The parameter $\zeta = \omega_1/\omega_2$ is the *relative mean frequency* of the elementary TRNG.

As we mainly deal with the relative phase between Osc_1 and Osc_2, we make the simplifying assumption that Osc_1 is a perfectly stable oscillator and that all the phase drift of the elementary TRNG comes from Osc_2, so that we have $\xi_1 = 0$ and we would like to characterize the phase jitter $\xi_2 = \xi$.

As shown in [1], the evolution of the phase can be modeled by an *ergodic stationary Markov process* $\Phi(t)$: for any time t, t_0, such that $t \geq t_0$, the phase $\Phi(t)$ determined by the initial value $\Phi(t_0) = x_0$ follows a probability distribution depending only on $\Delta t = t - t_0$ with mean $\xi(t_0) + \mu(\Delta t)$ and variance $V(\Delta t)$ where V, μ are real valued functions. In the following, we only consider a realization $\xi(t)$ of $\Phi(t)$ and use the stationarity of the process to compute probabilities, which are independent of the time of the realization. For instance, as $\mathbb{P}\{\Phi(t_0+\Delta t)-x_0 \leq x|\Phi(t_0) = x_0\}$ is independent of t_0, this probability can be computed by taking the probability over t_0 of the realization: $\mathbb{P}_{t_0}\{\xi(t_0+\Delta t)-\xi(t_0) \leq x\}$.

As $s_2(t) = f\alpha(\omega_2(t + \xi(t)))$ where ω_2 is the mean frequency of s_2, we deduce that $\mu(\Delta t) = \omega_2 \Delta t$. Thus, if the Markov process is Gaussian (i.e. $\frac{d}{dx}\mathbb{P}\{\Phi(t) \leq x|\Phi(t_0) = x_0\}$ is a Gaussian distribution), it is completely determined by $V(\Delta t)$. The random walk component of the phase jitter is produced by noise sources which affect each transition *independently*. This component is described by a Gaussian probability distribution of variance $\sigma_0^2 \Delta t$.

Other noise sources, such as the $1/f^\beta$ noises, where $0 < \beta < 2$, also contribute to phase jitter. Unfortunately, they are usually autocorrelated. Moreover, because their variance depends quadratically on the jitter accumulation time interval, after longer accumulation, they dominate the jitter coming from the thermal noise. For this reason, the accumulation time should be as short as possible, but long enough to obtain a measurable jitter. In practice, both uncorrelated and correlated noise sources exist and a typical log-log plot of $V(\Delta t)$ versus the measurement delay Δt can be used to separate regions with slope 1 and 2 as explained in [8].

4 Randomness Evaluation Method

In this section, we present a kind of Monte Carlo method to recover the probability density function $\frac{d}{dx}\mathbb{P}\{\Phi(t) \leq x | \Phi(t_0) = x_0\}$ of the jitter accumulated during time interval Δt from knowledge of an output bit sequence of an elementary oscillator-based TRNG depicted in Figure 1 with $K_D = 1$ so that the mean frequency of the sampling signal is ω_2. For $n \in \mathbb{N}^*$, let $(t_j)_{j\in\{1,\ldots,n\}}$ be the time sequence and $(b_j)_{j\in\{1,\ldots,n\}})$ be the output bit sequence corresponding to the rising edges of Osc_2 as depicted in Fig. 2. Recall that the sampled signal is $s_1(t) = f_\alpha(\omega_1 t)$ for $\alpha \in [0,1)$ and that by definition $t_j = jT_2 - \xi(t_j)$.

Next, we introduce a notation of ϵ-uniformity that we use in the remainder of the paper. It uses the modulo operation on real numbers illustrated in Fig. 2: for all $x \in \mathbb{R}$ and $T \in \mathbb{R}$, let $x \mod T = x - \max\{i \in \mathbb{Z} | x - iT \geq 0\}T$.

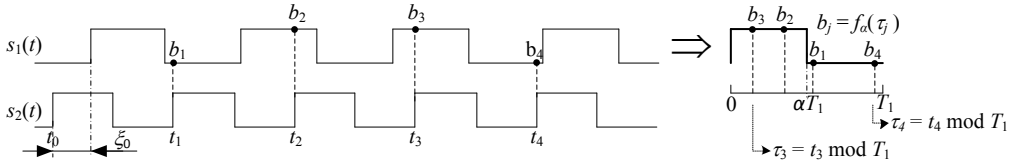


Fig. 2. *Relation between the sampling process and function* $f_\alpha(\cdot)$

Let J be a subset of $\{1,\ldots,n\}$ and $\epsilon > 0$, we say that the distribution of samples $\{(jT_2 - \xi(t_j)) \mod T_1\}_{j\in J}$ is ϵ-uniform if for all $[a,b] \subset [0,T_1]$, we have:

$$\left|\frac{\#\{j \in J | (jT_2 - \xi(t_j)) \mod T_1 \in [a,b]\}}{\#J} - \frac{b-a}{T_1}\right| < \epsilon.$$

In other words, the number of samples in interval $[a,b]$ inside the translated period T_1, over the number of samples in subset J is ϵ-close to the size of interval $[a,b]$ over period T_1. With this definition, we can state the following fact:

Fact 1 *Let $N \in \mathbb{N}$ and for $i \in \{1,\ldots,n-N+1\}$, we set $S_i = \{i,\ldots,i+N-1\}$. Let $\epsilon > 0$ be such that for all $i \in \{1,\ldots,n-N+1\}$ the distribution of samples $\{(jT_2 - \xi(t_j)) \mod T_1\}_{j\in S_i}$ is ϵ-uniform. Let $N \in \mathbb{N}$ be small enough so that the differences between successive values $\delta(j) = \xi(t_{j+M}) - \xi(t_j)$ are negligible (in other words, the value of $\delta(j)$ is almost constant, but sufficiently big) when j runs across all the elements of S_i for a fixed $i \in \{1,\ldots,n-N-M+1\}$. For $i_0 \in \{1,\ldots,n-N-M+1\}$, we define*

$$\mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\} = \frac{\#\{j \in S_{i_0} | b_j \neq b_{j+M}\}}{\#S_{i_0}}.$$

We see that if $(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M})) \mod T_1 \leq \min(\alpha T_1, (1-\alpha)T_1)$, then

$$\left|\mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\} - \left(\frac{2(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))}{T_1} \mod 1\right)\right| < \epsilon,$$

if $(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))) \mod T_1 \geq \max(\alpha T_1, (1-\alpha)T_1)$, *then*

$$\left| \mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\} + \left(\frac{2(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))}{T_1} \mod 1 \right) \right| < \epsilon,$$

otherwise

$$\left| \mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\} - 2\min(\alpha, 1-\alpha) \right| < \epsilon.$$

Proof of Fact 1 is given in Appendix A. It can be observed that for given values M, T_1, and T_2, the variance of the phase difference between samples at distance M (of the accumulated jitter we want to measure) is proportional to the variance of number of different samples in the given set of samples over the total number of samples in this set.

In the following, we present a very interesting application of Fact 1 that is able to recover the distribution of the phase jitter accumulated over a given number M of periods of Osc_2. We make M big enough so that the jitter accumulated during MT_2 is not negligible and N small enough so that the phase jitter can be considered as almost constant in the time period NT_2. Then Fact 1 signifies that it is possible to recover a good approximation of $(2(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))/T_1) \mod 1$ or $-(2(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))/T_1) \mod 1$ by computing $\mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\}$. More precisely, if we denote $\mathscr{C}$ the set of convergents of the continued fraction decomposition of T_2/T_1 (see [9] for the definition of the convergents of continued fraction decomposition) a careful analysis shows that in Fact 1, we can take $\epsilon = 1/\kappa$ where $\kappa = max\{q < N | p/q \in \mathscr{C}\}$. In practice, we have $\epsilon \approx 1/N$. If we make M small enough so that the standard deviation of the distribution of the jitter accumulated during MT_2 is small compared to $\min(\alpha T_1, (1-\alpha)T_1)$, the values of samples $(-MT_2 - \xi(t_{i_0}) + \xi(t_{i_0+M}))/T_1 \mod 1/2$ or $(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))/T_1 \mod 1/2$ follow the probability density function $\frac{d}{dx}\mathbb{P}\{\Phi(MT_2) \leq x | \Phi(0) = x_0\}$ up to a translation. If we denote $V(t)$ the variance of the probability distribution $\mathbb{P}\{\Phi(t) \leq x | \Phi(0) = x_0\}$, we obtain Algorithm 1 to compute $V(MT_2)$.

input : The output sequence $[b_1, \ldots, b_n]$ of an elementary TRNG with $K_D = 1$, K, M and N integers.
output: $V_0 = 4V/T_1^2$ where V is the variance of the jitter accumulated during MT_2.

for $i = 0, \ldots, K$ **do**
  $S_i \leftarrow [Ni + 1, \ldots, Ni + N]$;
  $c[i] = \mathbb{P}_S(b_j \neq b_{j+M})$;
end
$V_0 \leftarrow \frac{1}{K}\sum_{i=0}^{K} c[i]^2 - \left(\frac{1}{K}\sum_{i=0}^{K} c[i]\right)^2$;
return V_0;

Algorithm 1: *Algorithm for computing the variance V of the jitter*

It can be seen that Altorithm 1 is very simple: for computing the variance, it is necessary to count K-times, in successive N couples of bits, the number of couples having different bit values. The distance between the two bits in each couple is M. In practice, $K \sim 10000$, $N \sim 100$ and $M > N$, we let M vary between 200 and 1600.

It should be noted that $\mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\}$ may not return an approximation of $(2(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))/T_1) \mod 1$ or $(-2(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M}))/T_1) \mod 1$ if $(MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M})/T_1 \mod 1 \in [\min(\alpha, 1-\alpha), \max(\alpha, 1-\alpha)]$ but, as in practice $|\alpha - 1/2|$ is always small, these occurrences are rare and easy to detect.

5 Evaluation of the Method by Simulations

We evaluated the principle of the jitter measurement by simulations. In order to maintain coherence with later hardware simulations, we used VHDL package *rng.pkg* [15] for generating jittery clock signals. Using this package, we dynamically modified the timing of the two signals by adding a Gaussian jitter with zero mean and known standard deviation to each generated half period. The obtained clocks were used to generate a bitstream according to Fig. 1. The obtained bitstream file was then used as an input in mathematical evaluations. The objective of the simulations was to recover the jitter size that was indeed introduced to generated clocks, independently from the frequency ratio.

First, the mean clock period of the sampled oscillator Osc_1 was $T_1 = 8\,923$ ps and that of the sampling oscillator Osc_2 was $T_2 = 8\,803$ ps. For $i = 1, 2$, the output clock signal of Osc_i was given by $f_i = f_{1/2}(1/T_i(t + \xi_i(t)))$, where ξ_i is the random walk phase drift such that $\frac{d}{dx}\mathbb{P}\{\xi_i(t + \Delta t) \leq x | \xi_i(t)\}$ follows a Gaussian distribution of mean 0 and variance $\sigma_c^2 \Delta t / T_i$. It is satisfactorily approximated by oscillator Osc_1 with a fixed period and oscillator Osc_2 with a relative jitter $\xi(t)$ such that $\frac{d}{dx}\mathbb{P}\{\xi(t + \Delta t) \leq x | \xi(t)\}$ is a Gaussian distribution $G_{\Delta t}(x)$ with mean 0 and variance $\sigma_{T_2}^2 \Delta t / T_1 \simeq 2\sigma_c^2 \Delta t / T_1$ (see [1, Appendix C] for justification).

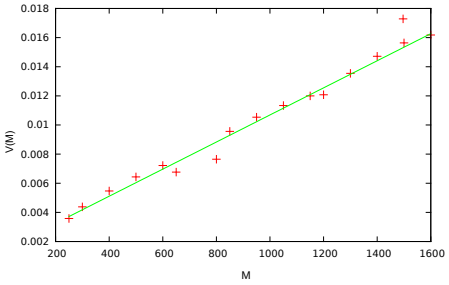
For $\sigma_c = 10$ ps, 15 ps, and 20 ps, we generated EO TRNG output bit sequences using the *rng.pkg* package. Next, using Algorithm 1, we computed the variance $V(M)$ of $G(MT_2)$ as a function of M and we plotted the graphs of $V(M)$ as a function of M for three above mentioned sizes of injected jitter (see left panel in Fig. 3 for $\sigma_c = 10$ ps). Similar results were obtained for different frequency ratios.

The variance was satisfactorily approximated by a linear function with slope a. We then compared the size of the injected jitter (σ_c/T_1) with that obtained from the slope $(\sqrt{a}/2)$. The results presented in the right panel in Fig. 3 show that we were able to recover expected noise parameters with good precision – the error was less than 5 %.

Note that our simulation does not take the $1/f$ noises into account, because there are no generators of such noises generating sufficiently long sequences available right now. Also note, that global noises need not be included: because of the use of the differential measurement principle – two ring oscillators implemented in the same device – impact of the global noise sources is eliminated (see [6] for more details).

6 Hardware Implementation of the Embedded Jitter Measurement

The jitter variance measurement was implemented in hardware according to Algorithm 1. It is presented in two blocks. The first block (see Fig. 4) computes K successive values $c_i = Nc[i]$ by comparing the output values of the first and the last stage of an $(M + 1)$-stage shift register and counting unequal bits during N periods of $s_2(t)$.



Injected jitter	Calculated slope	σ_c/T_1	$\sqrt{a}/2$	Error percentage
σ_c	a			
10 ps	$9.299909\ 10^{-6}$	0.00156	0.00152	2 %
15 ps	$2.03211\ 10^{-5}$	0.00234	0.00225	3 %
20 ps	$2.03211\ 10^{-5}$	0.00312	0.00297	5 %

Fig. 3. *Simulation results, left panel: $V(M)$ as a function of M (jitter with $\sigma_c = 10$ ps was injected); right panel: error percentage for three sizes of the jitter – 10 ps, 15 ps, and 20 ps.*

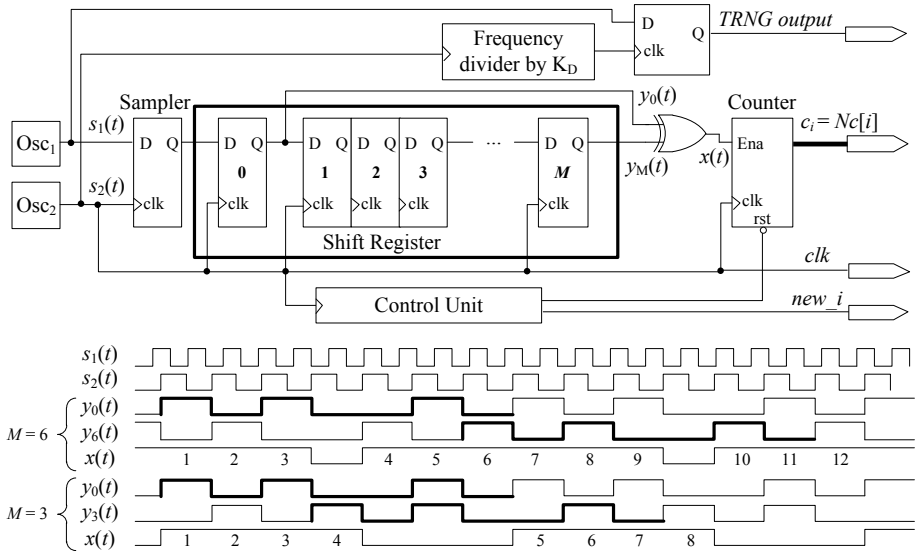


Fig. 4. *Structure of the block aimed at counting successive values $c_i = Nc[i] = N\mathbb{P}_S(b_j \neq b_{j+M})$ and two waveform examples for $M = 6$ (top panel) and $M = 3$ (bottom panel).*

The lower panel in Fig. 4 shows waveforms for the relative mean frequency $\zeta = T_2/T_1 = 10/7$ and given initial phase ξ_0. The sampler output features a repetitive pattern (in bold), depending on ζ and ξ_0. Two cases are depicted: in one, the distance between samples is $M = 6$ and in the other, $M = 3$. Since ζ and ξ_0 are constant, the pattern remains the same, but the XOR gate output differs. In fourteen ($N = 14$) clock periods T_2, we see 12 different bits in the first case and 8 in the second. According to Fact 1, for jitter-free clocks, these values will remain constant in all successive blocks of N bits, but

in the presence of the jitter, their variance will be proportional to the variance of the jitter.

A compromise must be found when determining the distance (M) between samples: for short distances, the accumulated jitter is too small and the precision is thus reduced; for long distances, two phenomena can occur: 1) the proportion of the flicker noise can become dominant or 2) accumulated jitter can become too big.

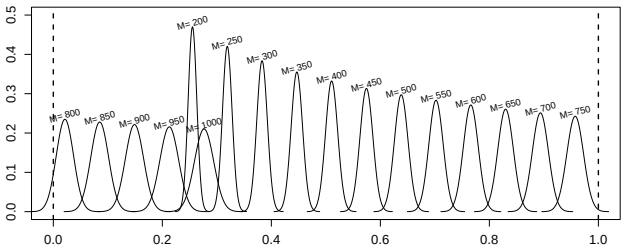


Fig. 5. *Example of distribution of values $c[i]$ between 0 and 1 (dashed vertical lines), for different values of M in steps of 50.*

One important fact must be considered: since the relative mean frequency and phase cannot be controlled (oscillators are free running), the mean number of unequal samples can be any value from interval $[0, N]$, depending on $\zeta = \omega_1/\omega_2$ and distance M. If the mean value is close to the border values of this interval, some measurements may fall outside the interval and cause a measurement error (see curves for M=750 and 800 in Fig. 5). Of course, this error could be corrected by translating the period T_1. However, this would require some additional computations. It is consequently more practical to ensure that the standard deviation of the accumulated jitter is much smaller than period T_1 and the mean values of $c[i]$ are sufficiently far from the interval borders. Distance M, whose values $c[i]$ do not fulfill the last condition should not be used for variance computation. The practical setup of the distance M will be discussed later.

The second block computes the relative variance $4V/T_1^2$ from K values $c[i]$ according to Algorithm 1 (see Fig. 6). The implementation of the block is quite straightforward. It uses two accumulators, two multipliers connected as squaring units and one subtractor. If the K value is chosen so that it is a power of two, division by K and K^2 can be implemented at no cost by shifting the result $\log_2 K$ and $2\log_2 K$ positions to the right, respectively.

Notice also, that this second computing block is used once per N periods T_2 and can thus be easily shared by several EO TRNGs without loss of performance.

Both blocks were implemented in VHDL as parameterized modules depending on parameters {NDE1, NDE2, M, N, and K}. The two oscillators were implemented as NDE1- and NDE2-element ring oscillators. Parameters M, N, and K represent the distance between samples, the length of measurement and the number of measurements, respectively.

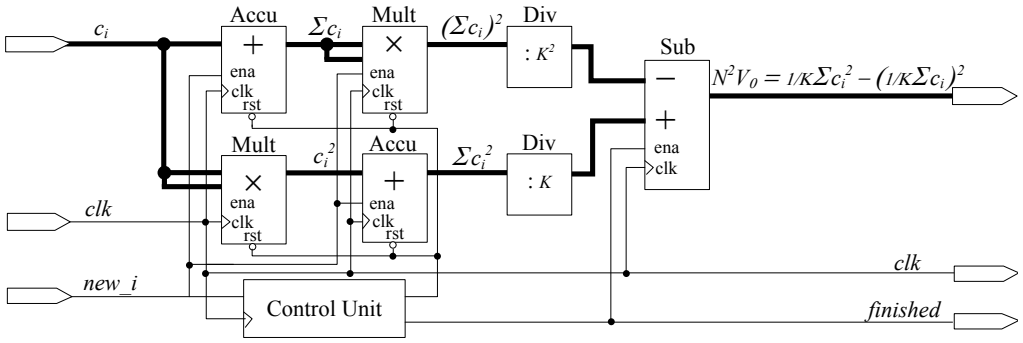


Fig. 6. *Structure of the block aimed at computing variance V_0 using K successive values $c[i] = \mathbb{P}_S(b_j \neq b_{j+M})$.*

6.1 Hardware Implementation Results

We tested the jitter measurement method in two different hardware configurations: 1) EO TRNG, jitter measurement and data interface (USB) were implemented in the same device; 2) the EO TRNG core in Fig. 1 was implemented in one FPGA and the jitter measurement and data interface were implemented in another. The aim of these two implementations was to observe the impact of the jitter measurement circuitry on the generator.

The first hardware configuration was implemented using an evaluation board dedicated to TRNG designs, featuring Altera Cyclone III FPGA and low noise linear power supplies (because of blind review, we will give the reference for the card only in the final version of the paper). As mentioned above, the elementary oscillator based TRNG is negligibly small. Its size is determined essentially by the number of delay elements of the two ring oscillators.

The size of the jitter measurement circuitry is determined by parameters M, N, and K. Practical experiments showed that the shift register should have between 200 and 500 stages (we recall that the depth of the shift register is linked to parameter M, which determines the jitter accumulation time). For less than 200 stages, the accumulated jitter variance only differed by a few bits and the precision was not sufficient (see Fig. 7). For bigger register sizes, the unwanted jitter coming from the correlated flicker noise became non negligible. According to Fact 1 and the simulation results presented in Section 5, to increase the precision of the measurement, the value of parameter N (number of samples used for computing mean values $c[i]$ from Algorithm 1) should be less than that of M. For this reason, we selected the N value to be around 150 and M between 250 and 450. For easy division by K, its value was set at 8192. The value $V_0 = 4V/T_1^2$ was then computed according to Algorithm 1 using 32-bit arithmetic operations and sent to PC via USB interface for further analysis. In the given configuration, the EO TRNG including jitter measurement circuitry occupied 301 logic cells (LEs), maximum 450 memory bits, plus one DSP block 9x9 and four DSP blocks 18x18.

Results of the jitter measurement in the first hardware configuration implemented in Altera Cyclone III FPGA for M varying between 250 and 1200, $N \sim 120$ and $K = 8192$ are depicted in Fig. 7. The left panel of the figure shows, that the variance increases linearly for $250 < M < 450$. This interval corresponds to accumulation times, during which the thermal noise dominates. The right graph in Fig. 7 is a zoom on this zone. From the dependence of the variance on M (the slope) and the period $T_1 = 7.81$ ns, we were able to compute the jitter size $\sigma = 5.01$ ps per period T_1.

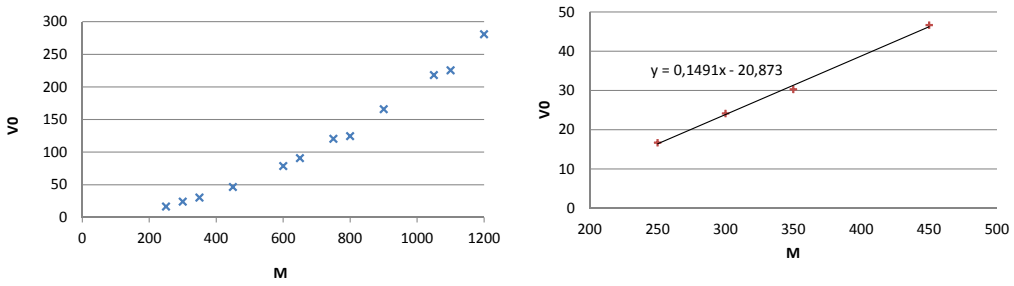


Fig. 7. *Results of the jitter measurement in hardware.*

The same measurement method was applied in the second hardware configuration, in which EO TRNG was implemented in a separate FPGA and the jitter measurement circuitry and data interface were implemented in the same evaluation board as the first configuration described above (Cyclone III FPGA). Both FPGAs were interconnected via the LVDS (low voltage differential signaling) interface for the transmission of two signals: the reference clock and the EO TRNG sampler output signal.

It is important to underline that because the TRNG signal was output after the sampler, the FPGA input/output circuitry did not have any impact on the jitter measurement, as is the case when standard jitter measurement methods are used to measure the jitter of outputs of the two rings using external equipment (e.g. oscilloscope).

The result of this second experiment was that the jitter standard deviation was $\sigma = 4.9$ ps per period $T_1 = 7.69$ ns. This is a negligible change from the jitter of 5.01 ps in the previous experiment. This means that the jitter measurement can be embedded in the same device as the EO TRNG.

7 Discussion on Entropy Management Using Embedded Jitter Measurement

During the jitter evaluation described in the previous section, we calculated jitter from the slope of the variance depending on M. This method was useful to determine the interval in which variance depends linearly on the accumulation time. However, for implementation inside the device, this would require additional circuitry (to compute the slope and variance from the slope) to be implemented inside the device. Fortunately,

knowing that the dependence in the selected interval is linear, it is sufficient to permanently measure just one point of the curve, i.e. just one value $V_0 = 4V/T_1^2$. We measured the jitter at $M = 300$. The measured standard deviation was $\sigma_0 = 2\sqrt{V}/T_1 = 5.01$ ps.

As explained in Sec. 6, for practical reasons, the variance should not be computed for values M, whose mean values $c[i]$ are close to zero or one. These values are not known in advance since oscillators are free running. If the jitter is sufficiently small compared to the T_1 period, which is always true for small accumulation times, these cases are rare, but unavoidable. For this reason, the shift register has several outputs around stage 300 and we selected one of the outputs, for which the computed values $c[i]$ were close to 0.5. This means the computation of their variance is free of errors.

Knowing the size of the jitter, we were able to manage the EO TRNG entropy: by entering the known jitter size in the model presented in [1], we computed the value of frequency divider K_D, to ensure that the entropy per bit is higher than $H_{min} = 0.997$, as required by AIS 31 [10]. The formula is derived from [1] and it gives K_D as an expression of σ_c, T_1, T_2 and H_{min}.

$$K_D = \frac{-\ln\left(\frac{\pi}{2}\sqrt{(1-H_{min})\ln(2)}\right)}{2\pi^2 \frac{T_2}{T_1}\frac{\sigma_c^2}{T_1^2}} \tag{2}$$

For $T_1 = 8.9$ ns, $T_2 = 8.7$ ns, $\sigma_c = 5.01$ ps and $H_{min} = 0.997$, we got $K_D \approx 430\,000$.

In this context, the role of the proposed jitter measurement circuitry is different: the continuous jitter measurement can be used as an on-line test, which should guarantee that the jitter never falls under the value that was used for entropy estimation and management (in our case, $\sigma = 5.01$ ps per period T_1 and $K_D = 430\,000$).

As mentioned above, the jitter measurement circuitry we proposed can be used in conjunction with a suitable stochastic model as a dedicated statistical test. In comparison with standard statistical tests, this test is performed closer to the source of randomness and can thus more accurately and more rapidly detect incorrect behavior of the generator.

For example, the tests FIPS 140-1 included in the AIS 31 RNG evaluation methodology require 20 000 input bits. Note that in our case, to obtain 20 000 bits at the generator output, we would need $K_D = 430\,000$ times more bits at the sampler output, i.e. at least $8.6 \cdot 10^9$ bits. However, in order to perform our dedicated test, which is better adapted to the detection of specific TRNG weaknesses (reduction in the jitter from the thermal noise or locking of the rings [3]), we only need $N \cdot K$ bits (around $1 \cdot 10^6$ sampler output bits). The dedicated test is thus more than 8 600 times faster and still very efficient. Our experiments showed that FIPS 140-1 tests were far less restrictive – the RNG output passed these tests for K_D as low as 100 000, probably because of the flicker noise.

As an example, we demonstrate the efficiency of the proposed test during a temperature attack on real hardware in Appendix B.

8 Conclusion

In this paper, we presented an original, simple and precise method of jitter measurement that can be implemented inside logic devices. We demonstrated that in conjunction with a suitable statistical model, the measured jitter can be used to estimate entropy at the

output of the generator. We also showed that the proposed entropy estimator can be used to build a rapid dedicated on-line statistical test that is perfectly adapted to the generator's principle. This approach complies with recent recommendations for TRNG evaluation [10] and ensures a high level of security by rapidly detecting all deviations from correct behavior.

Since the EO TRNG is the basic construction element of many oscillator based TRNGs including those based on self-timed rings [4], the proposed principle can be widely applied. However, in order to prevent attacks like those described in [12] and [2] (locking of rings), the jitter needs to be evaluated for all ring oscillators exploited in the generator. If necessary, the variance computation circuitry, as well as shift registers and counters of unequal samples, can be shared by all the rings in time.

References

1. M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux. On the security of oscillator-based random number generators. *Journal of Cryptology*, 24:398–425, 2011.
2. 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 W. Schindler and S. A. Huss, editors, *Constructive Side-Channel Analysis and Secure Design – COSADE 2012*, volume 7275 of *LNCS*, pages 151–166. Springer, 2012.
3. N. Bochard, F. Bernard, V. Fischer, and B. Valtchanov. True-Randomness and Pseudorandomness in Ring Oscillator-Based True RandomNumber Generators. *International Journal ofReconfigurable Computing, Article ID 879281*, page 13, 2010.
4. A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert. A Very High Speed True Random Number Generator with Entropy Assessment. In Coron J. S. Bertoni, G., editor, *Cryptographic Hardware and Embedded Systems – CHES 2013*, volume 8086 of *LNCS*, pages 179–196. Springer, 2013.
5. V. Fischer. A Closer Look at Security in Random Number Generators Design. In W. Schindler and S. A. Huss, editors, *Constructive Side-Channel Analysis and Secure Design – COSADE 2012*, volume 7275 of *LNCS*, pages 167–182. Springer, 2012.
6. V. Fischer, F. Bernard, N. Bochard, and M. Varchola. Enhancing Security of Ring Oscillator-based RNG Implemented in FPGA. In *Proceedings of Field Programmable Logic and Applications – FPLA 2008*, 2008.
7. T. Guneysu. True random number generation in block memories of reconfigurable devices. In Kang Zhao Jinian Bian, Qiang Zhou, editor, *Field-Programmable Technology – FPT 2010*, pages 200–207. IEEE Press, 2010.
8. A. Hajimiri, S. Limotyrakis, and T. Lee. Jitter and phase noise in ring oscillators. *IEEE Journal of Solid-State Circuits*, 34(6):790–804, 1999.
9. A. Ya. Khinchin. *Continued fractions*. The University of Chicago Press, Chicago, Ill.-London, 1964.
10. W. Killmann and W. Schindler. A proposal for: Functionality classes for random number generators, version 2.0. Technical report, Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn, September 2011. Accessed: 2014-01-03.
11. W. Maichen. *Digital Timing Measurements: From Scopes and Probes to Timing and Jitter*. Frontiers in Electronic Testing. Springer, 2010.
12. A. T. Markettos and S. W. Moore. The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators. In Gaj K. Clavier, C., editor, *Cryptographic Hardware and Embedded Systems – CHES 2009*, volume 5747 of *LNCS*, pages 317–331. Springer, 2009.

13. NIST SP800-22 rev. 1. A statistical test suite for random and pseudorandom number generators for cryptographic applications, August 2008. Available at http://csrc.nist.gov/CryptoToolkit/tkrng.html.
14. 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*, pages 109–119, 2007.
15. G. Swaminathan. Random number generators (RNG) VHDL package. http://www.ittc.ku.edu/EECS/EECS_546/magic/files/vlsi/vhdl/random.pkg, 1992. Accessed: 2014-01-03.
16. G. Taylor and G. Cox. Behind Intels New Random-Number Generator. http://spectrum.ieee.org/computing/hardware/behind-intels-new-randomnumber-generator/0, 2011. Accessed: 2014-01-03.
17. Boyan Valtchanov, Alain Aubert, Florent Bernard, and Viktor Fischer. Modeling and observing the jitter in ring oscillators implemented in FPGAs. In *11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, (DDECS 2008)*, pages 1–6, 2008.
18. M. Varchola and M. Drutarovsky. New High Entropy Element for FPGA Based True Random Number Generators. In Standaert F.X. Mangard, S., editor, *Cryptographic Hardware and Embedded Systems – CHES 2010*, volume 6225 of *LNCS*, pages 351–365. Springer, 2010.
19. I. Vasyltsov, E. Hambardzumyan, Y.-S. Kim, and B. Karpinskyy. Fast Digital TRNG Based on Metastable Ring Oscillator. In E. Oswald and P. Rohatgi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2008*, volume 5154 of *LNCS*, pages 164–180. Springer, 2008.
20. Wang Xueqing, William R. Eisenstadt, and Robert M. Fox. Embedded jitter measurement of high-speed i/o signals. 2007.

Appendix

A Proof of Fact 1

In this section, we use the following notations: for interval I and $t \in \mathbb{R}$, $I + t$ is the interval $\{x + t | x \in I\}$. If I, J are intervals, $I + J$ is the interval $\cup_{t \in J} I + t$. We consider intervals that are invariant under translation by $T \in \mathbb{R}$. Thus, if $I \subset \mathbb{R}$ is an interval, we let $I_T = \cup_{n \in \mathbb{Z}}(I + nT)$. For instance, $[0, 1)_2 = \cup_{i \in \mathbb{Z}}[2i, 2i + 1)$. If $I = [x, y]$ is an interval, by convention, we set $I = \emptyset$ if $x > y$, and we have the obvious extension for open or semi-open intervals.

Proof. We suppose that $\alpha \leq (1 - \alpha)$, if necessary by changing f_α by $1 - f_\alpha$. For $j \in \{1, \ldots, n\}$, we let $\tau_j = jT_2 - \xi(t_j) \mod T_1$. By definition, for all $j \in \{1, \ldots, n - M\}$, $b_j = f_\alpha(\omega_1(jT_2 - \xi(t_j)))$ and $b_{j+M} = f_\alpha(\omega_1((j + M)T_2 - \xi(t_{j+M})))$. As f_α is 1-periodic, we have $b_j \neq b_{j+M}$ if and only if the cardinality of the intersection of the interval $[\tau_j, \tau_{j+M}]_{T_1} = [0, (MT_2 + \xi(t_j) - \xi(t_{j+M})) \mod T_1]_{T_1} + ((jT_2 - \xi(t_j)) \mod T_1)$ with the set $\{0, \alpha T_1\}$ is equal to 1 (see Figure 8).

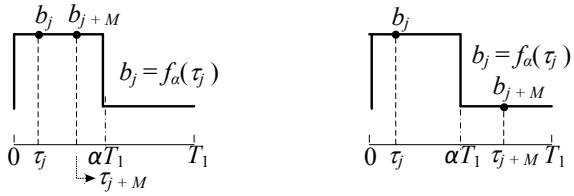


Fig. 8. *Keeping the notations of the proof of Fact 1, we have $b_j = b_{j+M} = 1$ (left) and $b_j = 1 \neq b_{j+M} = 0$ (right).*

Let $i_0 \in \{1, \ldots, n - N - M + 1\}$, using the hypothesis that $\delta(j) = \xi(t_{j+M}) - \xi(t_j)$ is almost a constant equal to $\delta(i_0)$ when j runs across all the values of $\{i_0, \ldots, i_0 + N - 1\}$, we deduce that $\mathbb{P}_{S_{i_0}}\{b_j \neq b_{j+M}\}$ is given by

$$P = \mathbb{P}_X\{\#(([0, (MT_2 + \xi(t_{i_0}) - \xi(t_{i_0+M})) \mod T_1]_{T_1} + X) \cap \{0, \alpha T_1\}) = 1\},$$

where X is a random variable, which follows the same distribution in the interval $[0, T_1]$ as the sample $\{(jT_2 - \xi(t_j)) \mod T_1\}_{j \in S_{i_0}}$. Let $\ell = (MT_2 + \xi(i_0) - \xi(i_0 + M)) \mod T_1$. Suppose that $\ell \leq \alpha T_1$, then the set of $x \in [0, T_1]$ such that $\#([x, x + \ell]_{T_1} \cap \{0, \alpha T_1\}) = 1$ is $([-\ell, 0]_{T_1} \cup [\alpha T_1 - \ell, \alpha T_1]_{T_1}) \cap [0, T_1]$. The size of the last interval is 2ℓ. The case $\ell \geq (1 - \alpha)T_1$ comes down to the preceding case by replacing ℓ by $T_1 - \ell$ and computing the complementary event. We obtain the size of $x \in [0, T_1]$ such that $\#([x, x + \ell]_{T_1} \cap \{0, \alpha T_1\}) = 1$ is $2(T_1 - \ell)$. On the other hand, if $\alpha T_1 \leq \ell \leq (1 - \alpha)T_1$, the set of $x \in [0, T_1]$ such that $\#([x, x + \ell]_{T_1} \cap \{0, \alpha T_1\}) = 1$ is $([-\ell, \alpha T_1 - \ell]_{T_1} \cup [0, \alpha T_1]_{T_1}) \cap [0, T_1]$, the size of which is $2\alpha T_1$.

Finally, by assuming that the distribution of X is ϵ-uniform in the interval $[0, T_1]$, we find that if $\ell \leq \alpha T_1$ then $|P - \frac{2\ell}{T_1}| < \epsilon$, if $\ell \geq (1 - \alpha)T_1$ then $|P - 2 + \frac{2\ell}{T_1}| < \epsilon$, and otherwise $|P - 2\alpha| < \epsilon$. This concludes the proof.

B Experiments on detection of attacks using the proposed dedicated test

The studied elementary oscillator based TRNG can be attacked by reducing the jitter, e. g. by decreasing the temperature and thus the thermal noise causing the jitter. We evaluated reaction of the proposed dedicated test on this attack.

In our experiments, we modified the temperature of the generator and we observed the size of the measured jitter and compared it with the pre-computed threshold in the dedicated test. The temperature was rapidly reduced to $-20\,^{\circ}\mathrm{C}$ and left to rise back to $21\,^{\circ}\mathrm{C}$. We repeated this cycle several times. The results of the jitter measurement in one experiment are depicted in Fig. 9.

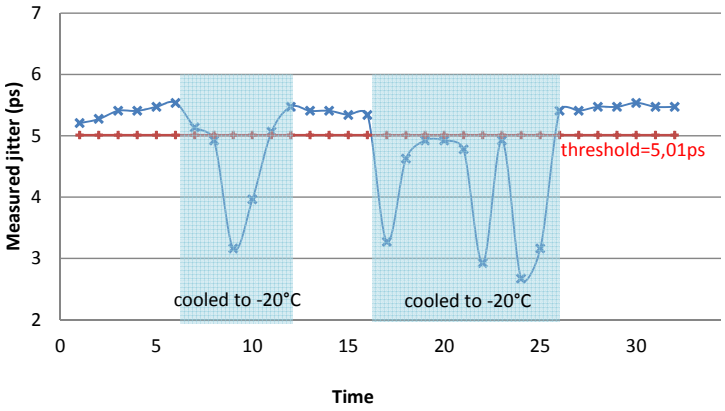


Fig. 9. *Evolution of the temperature attack in time.*

We see that as expected, the test was able to detect the jitter reduction coming from the temperature decrease and activate the alarm.