

# Two Methods of the Clock Jitter Measurement Aimed at Embedded TRNG Testing

Oto Petura\*, Marek Laban<sup>†</sup>, Elie Noumon Allini\*, Viktor Fischer\*

\*Hubert Curien Laboratory, UMR 5516 CNRS,  
Jean Monnet University Saint-Etienne  
18, rue Pr. Lauras, 42000 Saint-Etienne, France

<sup>†</sup>Department of Electronics and Multimedia  
Communications  
Technical University of Kosice  
Park Komenského 13, 04120 Kosice, Slovakia

email: (oto.petura, elie.noumon.allini, fischer)@univ-st-etienne.fr, laban@micronic.sk

**Abstract**—In modern cryptographic systems, security is based on quality and unpredictability of confidential keys. These keys are generated in random number generators using random physical phenomena appearing inside the cryptographic system on chip. The most frequently used source of randomness in digital devices is the jitter of clock signals generated inside the device in ring oscillators, self-timed rings, RC oscillators, phase-locked loops (PLLs), etc. The quality and unpredictability of generated numbers depends on the quality and the size of the clock jitter. It is therefore a good practice to monitor this jitter continuously using some embedded jitter measurement method. The measured jitter parameters can be then used as input parameters of the stochastic model used to estimate entropy, which characterizes unpredictability of generated numbers. In this paper, we present and compare two methods of embedded jitter assessment based on the measurement of the variance of counter values, obtained by counting the periods of the jittery clock during a time interval defined by a reference clock generated in the same device. Besides comparing obvious design results such as area, speed, and power consumption, we observe and discuss the impact of the two embedded variance measurement methods on the clock jitter itself, and compare the behavior of the two clock generators used as sources of randomness with and without clock variance measurement circuitry, and with and without additional logic such as an AES cipher, which perturbs the variance computation, as it is the case in most cryptographic embedded systems. This comparison is very important for a good estimation of the low entropy bound from the measurement results.

## I. INTRODUCTION

Security of modern cryptographic systems is based on statistical quality and unpredictability of confidential keys. These keys are generated in random number generators using random physical phenomena, appearing in hardware devices, in which the system is implemented. The most frequently used source of randomness in digital devices is the jitter of clock signal generated inside the device in ring oscillators, self-timed rings, RC oscillators, phase-locked loops (PLLs), etc.

The quality and unpredictability of generated numbers depends on the quality and on the size of the clock jitter. It is therefore a good practice to monitor this jitter continuously using some embedded jitter measurement method. As required in the document AIS-20/31 edited by the German Federal Bureau for Information Security (BSI) [1], the measured jitter parameters are then used as input parameters of the stochastic

model used to estimate entropy characterizing unpredictability of generated numbers.

In [2], Baudet *et al.* proposed a comprehensive stochastic model for an oscillator based random number generator, in which the entropy rate at generator output is estimated from the variance of the random jitter.

In [3], Haddad *et al.* proposed to measure the variance of the random jitter from the variance of counter values obtained by counting the number of periods of the jittery clock signal during a time interval defined using a second clock signal generated in the same device as presented in Fig. 1.

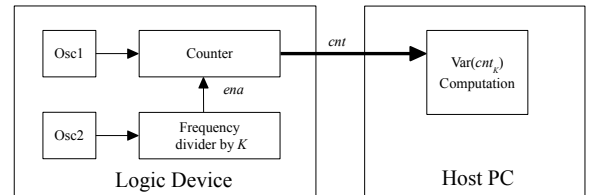


Fig. 1. Jitter variance measurement proposed by Haddad *et al.* in [3]

Instead of computing the variance outside the device, we implement the embedded variance measurement. In this paper, we implement two methods of measurement of the variance of the jitter presented in Section II and evaluate their implementation results in Section III. Next, we study the impact of the measurement circuitry and of the additional logic represented by an AES cipher on the source of randomness in Section IV. In Section V, we present and discuss the jitter measurement results in various implementation conditions. We conclude the paper in Section VI.

## II. THE VARIANCE MEASUREMENT METHODS

The first studied variance measurement method is based on the Koenig-Huygens theorem, from which the variance is expressed as follows:

$$V(X) = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right) - E(X)^2, \quad (1)$$

where  $x_i$  are the counter values and  $E(X)$  is their mean value.

The second method computes the variance from the frequencies  $f_1, f_2, \dots, f_m$  of the possible counter values  $x_1, x_2, \dots, x_m$ :

$$V(X) = \left( \sum_{i=1}^m f_i x_i^2 \right) - \left( \sum_{i=1}^m f_i x_i \right)^2. \quad (2)$$

While both methods should give the same results, it is clear that their implementation and execution in hardware is completely different.

The circuitry corresponding to implementation of Eq. (1) in hardware is depicted in Fig. 2. As can be seen in this figure,

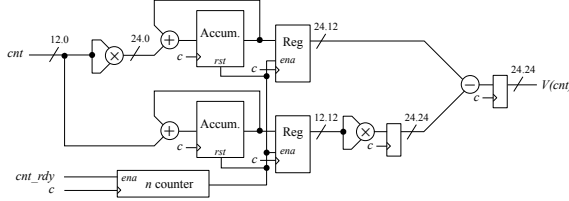


Fig. 2. Counter variance measurement circuitry based on Eq. (1)

all the computations are made in a fixed-point arithmetic. Numbers before and after the radix point indicate number of bits before and after this radix point. Two multipliers (one of 12 bits and the second one of 24 bits) are used to square data. Two adders and associated registers (one of 24 bits and the second one 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 circuitry corresponding to implementation of Eq. (2) in hardware is depicted in Fig. 3. Again, all the computations are

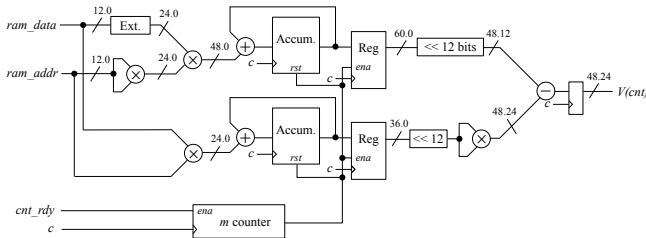


Fig. 3. Counter variance measurement circuitry based on Eq. (2)

made in a fixed-point arithmetic. Note, that the figure does not depict the memory in which the histogram of counter values is saved (accumulated data are obtained using *ram\_data* and *ram\_addr* signals depicted in this figure). Two multipliers (one of 12 bits and the second one of 36 bits) are used to square data, two other multipliers are used to multiply 24-bit data and 12-bit data, respectively. Two adders and associated registers (one of 24 bits and the second one 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.

### III. IMPLEMENTATION RESULTS AND DISCUSSION

First, we evaluate common design parameters like area, speed and power/energy consumption for the two variance measurement methods. We implemented both of the variance measurements on Intel Cyclone V FPGA in order to obtain area requirements and timing analyzer speed estimation. Power consumption was measured using HECTOR evaluation platform [4]. The results are presented in Table I.

TABLE I  
SUMMARY OF IMPLEMENTATION RESULTS OF THE TWO VARIANCE MEASUREMENT METHODS IMPLEMENTED IN THE HECTOR DAUGHTER BOARD FEATURING INTEL CYCLONE V FPGA DEVICE 5CEBA4F17C8N

Method	Area			$f_{max}$ [MHz]	Power [mW]
	ALM/Regs	Mem. blocks	DSPs		
Eq. (1)	119/163	0	2	178.3	6-7
Eq. (2)	156/311	6	6	67.84	7-8

The accumulation memory needed to compute the variance according to Eq. (2) is included in Tab. I. Although the first method clearly wins before the second one in terms of area and speed (only 119 Adaptive Logic Modules (ALM) needed by the first method vs. 156 needed by the second method, 163 registers vs. 311 registers in the second circuitry, no memory block needed vs. 6 block used in the second case), it is important to note that all the computations of the first method are made continuously and on the fly. They are thus impacted by the measured counter values.

On the other hand, the second method computes the variance in two steps. In the first step, which is running continuously, only simple computations are made (accumulation of samples in the accumulation memory). The second step is made independently from data acquisition – only the accumulation memory contents is addressed and used. We will study this difference in the next section.

The power consumption of both methods is comparable, even though the second one consumes always slightly more power.

### IV. STUDY OF THE IMPACT OF THE MEASUREMENT CIRCUITRY ON THE SOURCE OF RANDOMNESS

Next, we propose a rigorous approach to study 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 four steps:

- 1) **Project 1** – Only two ring oscillators, which will be 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. 4).
- 2) **Project 2** – The two ring oscillators and the variance measurement circuitry are implemented in the device as presented in Fig. 5. The generated clock signals are output using LVDS pins and measured using oscilloscope as previously and their variance is measured also

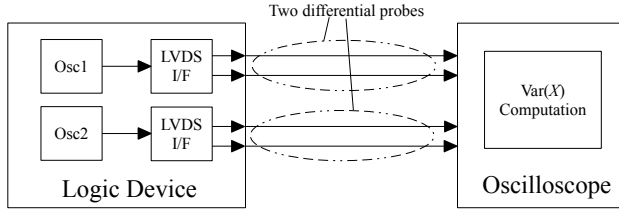


Fig. 4. External jitter measurement method using oscilloscope and differential probes

inside the device using the two presented methods. The variances measured externally (using the oscilloscope) and internally (using the two methods) are compared.

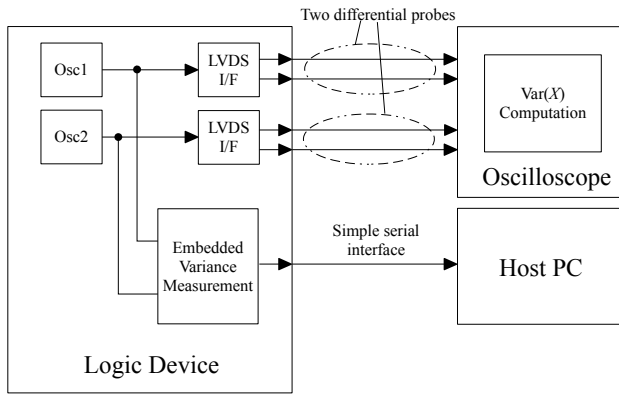


Fig. 5. External jitter measurement method using oscilloscope and differential probes combined with an internal jitter measurement method

- 3) **Project 3** – A complete TRNG using two ring oscillators and the variance measurement circuitry are implemented in the device and at the same time, the differential signal outputs are used to measure the jitter variance externally (see Fig. 6). The TRNG output data are acquired in real time using an external acquisition card featuring 32-MB RAM and transferred to the host PC using a high speed USB connection.
- 4) **Project 4** – An AES cipher is implemented alongside the complete TRNG in order to mimic the behavior of the real crypto SoC as presented in Fig. 7. The variance is measured both internally, and externally.

In Projects 2 to 4, the embedded measurement gives the variance of counter values, which represent number of clock periods of *Osc1* during  $K = 3000$  reference clock periods of *Osc2*.

When performing the measurement using oscilloscope, the value of  $K$  cannot be fixed like it is done in hardware – it can be only deduced from the oscilloscope time base, which was set in our case to  $2 \mu s$  per division. We measured the number of periods of both clocks in a time interval fixed by

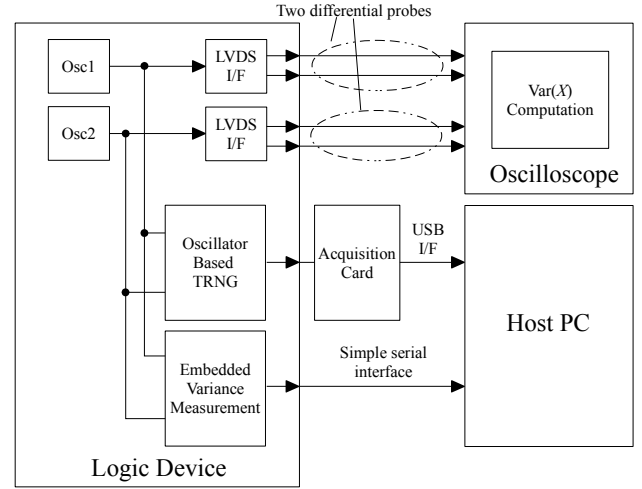


Fig. 6. External jitter measurement method using oscilloscope and differential probes combined with an internal jitter measurement method while the TRNG is running

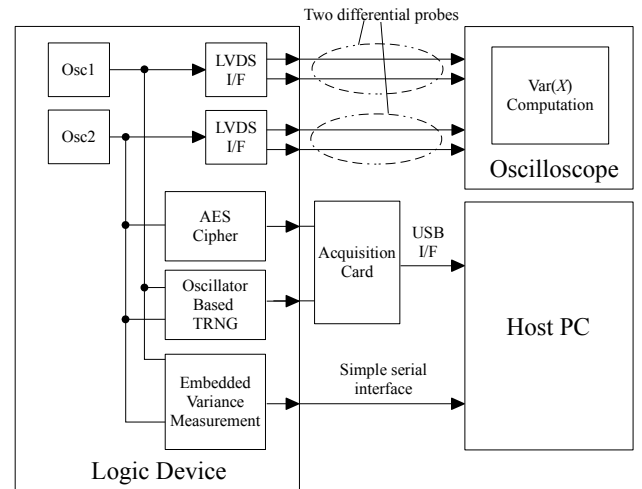


Fig. 7. External jitter measurement method using oscilloscope and differential probes combined with an internal jitter measurement method while the TRNG and the AES cipher are running

the oscilloscope time base. Finally, to make the comparison of values obtained by the external and embedded measurements more consistent, we measure the number of cycles of both clocks at the same time base interval and process the data according to the following equation:

$$cnt = \frac{n_1}{n_2} \cdot K, \quad (3)$$

where  $n_1$  represents the number of clock periods of *Osc1* and  $n_2$  the number of clock periods of *Osc2* appearing during the same time interval determined by oscilloscope's time base.

To maintain the measurement results consistent, it is impor-

tant to guarantee the same placement and routing of *Osc1* and *Osc2* in Project 1 to 4. 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 projects. We then used the same strategy to export the netlist of the variance measurement block from Project 2 (to Project 3 and 4) and the complete TRNG including the variance measurement method from Project 3 to Project 4.

Ring oscillators *Osc1* and *Osc2* had the same number of elements and the same topology. They oscillated at respective frequencies of  $275.7 \pm 1.17$  MHz and  $267.6 \pm 0.8$  MHz. The difference in their frequency in all projects was thus smaller than 1 %, which was important to get comparable results.

The variance measurement circuitry from Fig. 2 and 3 was clocked at 25 MHz (clock signal *c* in these figures). This clock signal was obtained from the low-jitter quartz oscillator oscillating at the frequency of 125 MHz using an embedded PLL, which divided its frequency by 5.

## V. MEASUREMENT RESULTS AND DISCUSSION

In this section, we evaluate the two variance measurement methods implemented in the selected logic device as a part of Project 1 to 4 presented in the previous section. Namely, we are interested in comparing the two measurement methods in terms of their noise susceptibility. Table II shows the results of different variance measurement methods.

TABLE II  
VARIANCE MEASUREMENT RESULTS

Project, method	Variance measured by	
	oscilloscope	hardware
Project 1	0.449	N/A
Project 2, Eq. (1)	0.450	0.398
Project 2, Eq. (2)	0.492	0.415
Project 3, Eq. (1)	0.439	0.391
Project 3, Eq. (2)	0.454	0.400
Project 4, Eq. (1)	0.458	0.396
Project 4, Eq. (2)	0.531	0.403

Table II shows, that the variance measured externally has always higher values than that measured in hardware. This fact can be caused by the external measurement equipment (FPGA outputs, probes, oscilloscope, etc.), which adds some additional jitter and consequently, lowers measurement precision.

This observation is very important for the TRNG security. Indeed, the most dangerous error in the TRNG security evaluation is entropy rate overestimation due to incorrect quantification of the source of entropy, i.e. the counter values variations in our case. (we recall that a sufficient entropy rate guarantees unpredictability of generated random values and thus security).

Another important result that can be observed in Table II is that the variance values obtained using the first method based on Eq. 1 are always smaller than those obtained using the second method based on Eq. 2. This is consistent in all projects (Project 2 to 4).

It is also interesting to note that the variance measured using the oscilloscope has the same behavior: the variance of counter values measured outside the device is always smaller when Method 1 is implemented in the device (independently of the type of the project) comparing to the case when Method 2 is used. This can be caused by the noise generated by the measurement circuitry, which is more complex for Method 2 (it includes embedded memory and more multipliers and registers).

## VI. CONCLUSIONS

The two measurement methods can be more or less suitable for different data format and for different kind of applications. However, in our testing scenarios, the first method is better in terms of area, speed, and power consumption.

Both embedded methods provide consistently lower variance values than the external measurement used by most designers. This means, that the jitter of internal clock signals is lower than the jitter observed outside the device, which avoids entropy rate overestimation.

Another positive fact is that we did not observe any significant impact of the surrounding logic (of the TRNG and AES cipher circuitry implemented in the same device) on the embedded measurement results. This is important to ensure that the generator output will not be manipulable.

However, it should be taken into account that the variance measurement method itself can still have some negative impact on the size of the counter values. This especially true when measurement Method 2 is applied.

## ACKNOWLEDGMENT

This work has 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.

## REFERENCES

- [1] W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators, version 2.0," 2011. [Online]. Available: [https://www.bsi.bund.de/EN/Home/home\\_node.htm](https://www.bsi.bund.de/EN/Home/home_node.htm)
- [2] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, "On the security of oscillator-based random number generators," *Journal of Cryptology*, vol. 24, no. 2, pp. 398–425, 2011.
- [3] P. Haddad, F. Berdnard, 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.
- [4] 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, Nov. 2016.