Testing of PLL-based True Random Number Generator in Changing Working Conditions

Martin ŠIMKA [1], *Miloš DRUTAROVSKÝ* [2], *Viktor FISCHER* [3]

[1] Datel, Drawska 10/17, 02-202 Warsaw, Poland
[2] Dept. of Electronics and Multimedia Comm., Technical Univ. of Košice, Park Komenského 13, 041-20 Košice, Slovakia
[3] Laboratoire Hubert Curien UMR CNRS 5516, Bât. F 18 Rue du Professeur Benoît Lauras, 42000 Saint Etienne, France.

info@martinsimka.com, Milos.Drutarovsky@tuke.sk, fischer@univ-st-etienne.fr

Abstract. *Security of cryptographic systems depends significantly on security of secret keys. Unpredictability of the keys is achieved by their generation by True Random Number Generators (TRNGs). In the paper we analyze behavior of the Phase-Locked Loop (PLL) based TRNG in changing working environment. The frequency of signals synthesized by PLL may be naturally influenced by chip temperature. We show what impact the temperature has on the quality of generated random sequence of the PLL-based TRNG. Thank to analysis of internal signals of the generator we are able to prove dependencies between the PLL parameters, statistical parameters of the generated sequence and temperature. Considering the measured results of experiments we form a new requirement in order to improve the robustness of the designed TRNG.*

Keywords

TRNG, phase-locked loop, cryptographic attacks, tracking jitter, on-line randomness tests, FIPS 140-2.

1. Introduction

Random values play a crucial role in several areas of science, e.g. in simulation methods like Monte Carlo [1], in generation of spreading sequences in spread spectrum communication systems [2], in generation of primes, in several cryptographic algorithms [3], or in gambling industry. Naturally, the requirements for generators and generated random data differ according to the application.

In cryptography, the values produced by randomness extractors or generators are used as cryptographic keys, initialization vectors, padding bits, blinding values and/or masking values in countermeasures against side-channel attacks [4]. Beside good statistical properties, random numbers in cryptography (usually random bitstream) must not be predictable and are often generated using some physical uncontrollable process. These numbers are called true random numbers as the opposite to the pseudo-random numbers that are generated using some deterministic algorithm and are thus guessable.

The entropy of True Random Number Generators (TRNGs) is increased by each generated value. Much attention was paid later to the analysis and mathematical modeling of the randomness extraction process. Several papers (e.g. [5], [6]) give theoretical bounds for entropy and provide statistical estimations of the TRNG behavior in order to characterize its security. However, the difficulty concerning these models and entropy estimators is related to underlying physical assumptions that are often difficult or impossible to validate. Some models and proofs of security like that of Sunar et al. [5] can be thus questionable [7], [8].

When designing a TRNG for cryptography, a designer has to take into account that attackers can try to manipulate the generator in order to be able to guess some patterns in the generated bitstream or the whole key with a non negligible probability. There are two types of attacks that should be considered when designing TRNGs: *passive attacks* and *active attacks*. Passive attacks do not modify in any way the functionality of the security device. In case of TRNGs, an attacker can try to guess the generated bitstream when measuring the power consumption or electro-magnetic emissions. Active attacks require some non-invasive or invasive active intervention of the attacker aimed at modification of TRNG behavior. *Non-invasive active attacks* are based on non-permanent variations of the TRNG environment (e.g. supply voltage, temperature, surrounding electro-magnetic field) while trying to achieve some anomalies (e.g. bias from equiprobable values) at the TRNG output. The authors of [9] were able to reduce the keyspace from 2^{64} to just 3300 thanks to frequency injection attack on ring oscillator-based TRNG. Non-invasive attacks are dangerous for two reasons: 1) they are not tamper-evident in most cases; 2) they do not need expensive equipment.

To obtain results of a real-life attack we have executed a simple active non-invasive attack on Field Programmable Gate Arrays (FPGAs) implementation of a TRNG using Phase-Locked Loops (PLLs) as a source of randomness [10]. Namely, we have tried to introduce some bias detectable by FIPS 140-2 statistical tests to the output of generator

by changing its working temperature. Our aim is to find out what kind of changes in the parameters of generated sequences can be observed. Moreover, we will record the internal signals of the generator and evaluate their dependence on the temperature.

Similar experiments have been described in [11] where the PLL-based TRNG (PLL-TRNG) has been evaluated as problematic, with varying quality of the generated bit sequence. Based on obtained results from the attack realization we will provide additional requirements for the PLL-TRNG design and explain why the configuration chosen by Santoro et al. [11] had difficulties in passing statistical tests. Considering the measured results of experiments and extended analysis of internal TRNG signals we form a new PLL-TRNG design requirement in order to improve the robustness of designed TRNG.

The paper is organized as follows: in Section 2 we introduce PLL-TRNG proposed by Fischer and Drutarovsky in [10] and describe a method of jitter characterization based on coherent sampling. Section 3 describes experimental setup used for jitter measurement and detection of environmental manipulations. Results of the experiment are presented in Section 4. In Section 5 we discuss the obtained results. Finally, Section 6 concludes the paper.

2. PLL-Based TRNG in FPGA

In this section, we introduce PLL-TRNG principle based on randomness extraction from the tracking jitter that is inherent in clock signal generated in analog PLL.

2.1 PLL as a Source of Randomness

The PLL circuitry in FPGAs is aimed at the on-chip synthesis of clock signals derived from an external clock generator, e.g. quartz oscillator. It provides a set of signals with phase and frequency ratio, which can be set by FPGA static or dynamic configuration. PLLs are embedded in most recent FPGA families. Contrary to other TRNG principles implemented in FPGAs, the randomness source in PLL-TRNGs can be well separated from the rest of the device, because PLLs use physically separated power supply and ground pins. This is the main advantage of their use in security aware TRNGs as the power supply separation reduces the possibilities of attack.

Typical analog PLL block in Altera [12], [13], Xilinx [14], [15] and Actel [16], [17] devices (see Fig. 1) can provide at least one synthesized clock signal with frequency F_{OUT}:

$$F_{OUT} = \frac{F_{VCO}}{k} = F_{REF}\frac{m}{k} = F_{IN}\frac{m}{n \times k} \qquad (1)$$

where F_{IN} is the frequency of input clock signal, F_{REF} is the reference frequency derived from input signal and F_{VCO} is the frequency of the Voltage Controlled Oscillator (VCO) that is used to generate the feedback clock F_{FB}. Reference-, feedback- and post-divider values n, m, and k can vary from one to several hundreds in FPGAs [18], or to several thousands in ASICs [19]. The set of dividers values, loop filter bandwidth and VCO parameters determine the range of input and output frequencies, but also the jitter characteristics.

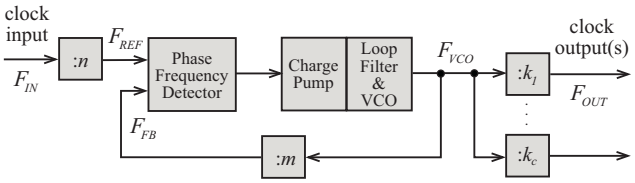

Fig. 1. Block diagram of analog PLL circuitry for clock signal synthesis in Altera FPGA [18].

Tab. 1 presents basic parameters of the PLL circuitry embedded in Altera Cyclone and Stratix FPGA families.

family	# of PLLs	dividers range		
		m	n	k
Cyclone II [20]	2, 4	1-32	1-4	1-32
Cyclone III [21], Cyclone IV [13]	4, 8	1-512	1-512	1-512
Stratix [22]	4, 8×FPLL*	1-32	1-32	1-32
	2, 4×EPLL	1-512	1-512	1-1024
Stratix II [23]	4, 8×FPLL	1-32	1-4	1-32
	2, 4×EPLL	1-512	1-512	1-512
Stratix V [12]	22, 28×FrPLL**	1-512	1-512	1-512

* *EPLL and FPLL stand for Enhanced and Fast PLL, respectively.*

** *FrPLL stands for Fractional PLL with 18 output counters.*

Tab. 1. Parameters of PLL embedded in Altera FPGAs.

When considering PLL as a source of randomness, three parameters have to be taken into account: 1) the size of generated clock jitter; 2) available ranges of frequency dividers n, m and k together with the VCO maximum and minimum frequency; 3) bandwidth of the PLL loop filter. Since the timing jitter is one of factors determining the reliability of fast synchronous logic systems, it is permanently decreased by FPGA vendors. This fact, which was proved also by our experiments, makes the PLL-based randomness generation more difficult. Fortunately, the range of dividers is increased in high-density FPGA devices for achieving more precise frequency synthesis. When a small clock jitter is available, bigger values of dividers can be successfully used.

2.2 Clock Jitter and its Measurements

The clock jitter can be defined as a short-term variation of an event from its ideal position. In general, it is the variation in time of the zero crossing (rising or falling edge) of the clock signal. The most common jitter measurements used by FPGA vendors are period jitter and cycle-to-cycle jitter. The period jitter is defined as the difference between the n-th clock period and the mean clock period. On the other side, the cycle-to-cycle jitter is defined as the difference between two successive clock periods. The jitter can

be considered as a random variable. If it has a normal distribution, it is characterized usually by a 1-sigma value (σ), where σ is the standard deviation of a jitter process and σ^2 is its variance. If the jitter is composed of both random and deterministic components, its size is usually given in a peak-to-peak value. Typical values of the clock jitter depend on the technology and on the configuration of the PLL. They can range from 3.5 ps to 10 ps for ASICs [19], or up to 100 ps for FPGAs [23], [18].

FPGA user cannot modify technology-related parameters of the jitter. He can only modify the PLL output clock jitter by modifying frequency dividers (m,n,k) or, optionally, loop filter bandwidth (only in Altera technology). Note, that the low-bandwidth PLL cannot follow fast changes on input clock and it thus filters out the jitter, but it needs longer time for locking.

Oscillators (including VCOs) are dependent on temperature variations by their nature. We can therefore expect that the behavior of the PLL will change as the temperature varies. We will evaluate the impact of the temperature on the PLL-TRNG in our experiments.

2.3 Randomness Extraction

Next, we will explain the principle of randomness extraction in the PLL-TRNG published in [10]. It uses the *tracking jitter* as a source of randomness. The tracking jitter can be defined as the difference between input and output clock signals of the PLL. The tracking jitter is related to the capability of the PLL circuitry to track changes in input clock signal. The PLL input can be driven by an external clock generator or internal signal including other PLL or RC oscillator embedded in the same FPGA device [24].

In PLL-TRNG (see Fig. 2), a clock *CLJ* influenced by jitter is sampled using another rationally-related clock signal *CLK* that is used as a reference clock. The relation between the two clocks is fundamental – it guarantees that the generator will generate random bitstream. This relation is assured by the PLL frequency synthesizers.

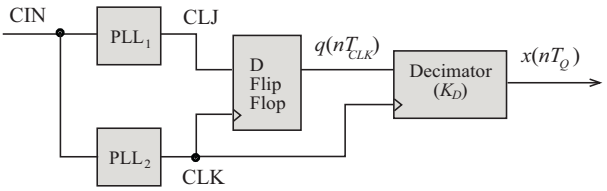

Fig. 2. Block structure of the PLL-TRNG having two PLLs, sampling gate and decimator – corrector of the output sequence.

The two PLLs assure that the frequencies F_{CLJ} and F_{CLK} of the synthesized clock signals *CLK* and *CLJ* are related following the form:

$$\frac{F_{CLJ}}{F_{CLK}} = \frac{K_M}{K_D} = \frac{M_{CLJ}D_{CLK}}{M_{CLK}D_{CLJ}} \tag{2}$$

where K_M and K_D represent multiplication and division factors determined by PLL frequency dividers (D_{CLK}, D_{CLJ}) and multipliers (M_{CLK}, M_{CLJ}). The clock signal *CLJ* is sampled on the rising edges of the clock signal *CLK*. The sampling is illustrated in Fig. 3. As it can be seen, the signal *CLJ* is sampled in K_D discrete positions during the period T_Q, which is given as

$$T_Q = K_D T_{CLK} = K_M T_{CLJ}. \tag{3}$$

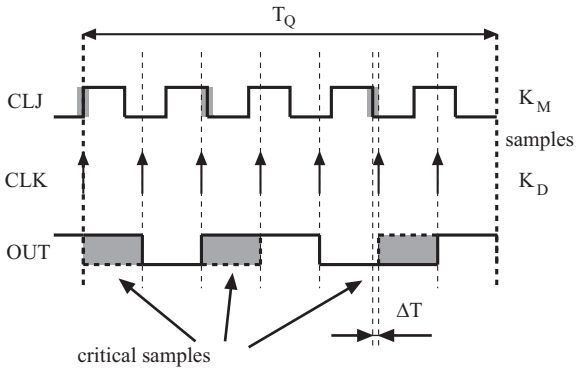

Fig. 3. Sampling of the CLJ clock signal including the tracking jitter on the raising edge of the CLK signal (illustrated for K_M = 5 and K_D = 7).

It was shown in [10] that if K_M and K_D are relatively prime, the set of samples creates an equidistant set of values with the step

$$\begin{aligned} d &= \frac{T_{CLK}}{2K_M}\mathrm{GCD}(2K_M,K_D) = \\ &= \frac{T_{CLJ}}{2K_D}\mathrm{GCD}(2K_M,K_D). \end{aligned} \tag{4}$$

The method offers a possibility to choose the worst-case distance $\mathrm{MAX}(\Delta T_{min}) = d/2$ between two closest edges of the *CLK* and *CLJ* signal during T_Q period:

$$\begin{aligned} \mathrm{MAX}(\Delta T_{min}) &= \frac{T_{CLK}}{4K_M}\mathrm{GCD}(2K_M,K_D) = \\ &= \frac{T_{CLJ}}{4K_D}\mathrm{GCD}(2K_M,K_D). \end{aligned} \tag{5}$$

If the parameters K_M and K_D are chosen so that

$$\mathrm{MAX}(\Delta T_{min}) < \sigma, \tag{6}$$

it is guaranteed with a high probability that during the period T_Q the sampling edge of *CLK* will fall at least once into the edge zone of *CLJ* (here the edge zone means the time interval around the edge with the total width smaller than σ). The K_D samples represented by the output signal $q(nT_{CLK})$ are XOR-ed bit-wise in a decimator [10] for obtaining one random bit during each period T_Q. The generator output bitrate R is thus $R = 1/T_Q$.

2.4 Jitter Measurement

The use of coherent sampling method [6] for randomness extraction enables simple on-line jitter measurement

and characterization inside the logic device. This measurement can be used for realization of embedded statistical test dedicated to the randomness generation principle.

If the clock periods are influenced by a random jitter, the output bits at the decimator output (notice, that the XOR-based decimator [25] computes one output bit per each period T_Q by adding modulo 2 ($\oplus$) of K_D samples $q(.)$ at the sampling frequency F_{CLK}) get the values

$$x(nT_Q) = q(nT_Q) \oplus q(nT_Q - T_{CLK}) \oplus \ldots \\ \ldots \oplus q(nT_Q - (K_D - 1)T_{CLK}). \quad (7)$$

The output bitstream thus represents a nondeterministic signal.

In order to be able to characterize the tracking jitter in situ, let us provide some more details concerning coherent sampling. We assume that during the period T_Q we acquired K_D samples q_i of the *CLJ* signal with order $i = 0, 1, \ldots, K_D - 1$. Next, we need to reorder the samples according to their timing position in the *CLJ* signal. The idea behind this reordering is the same as that used in sampling oscilloscopes, where high-frequency periodic signal (our jittery clock *CLJ*) is sampled using some low-frequency periodic signal (our reference clock *CLK*). The mutual phase of the two clock signals evolves in discrete steps (defined in our case by the distance d). The mutual phase repeats periodically, having a long period (in our case T_Q). In order to reconstruct one period of the signal *CLJ* from K_D samples indexed by i as defined before, one has to reorder these samples using new ordering index j defined by [26]:

$$\widehat{q_j} = q_i \quad (8)$$

where

$$j = iK_M \bmod K_D. \quad (9)$$

The value of the j-th sample $\widehat{q_j}$ $(0 \leq j \leq K_D - 1)$ can be viewed as a binary random variable $X_j \in \{0, 1\}$. Its mean value $E[X_j]$ is equal to the probability $p_j = p(X_j = 1)$, which is related to the value of the jitter in the corresponding sampling instant. In order to calculate probability distribution of all K_D samples, we count number of occurrences of each sample $\widehat{q_j}$ during N periods T_Q and thus estimate their mean values. Moreover, as it will be used in next sections, the values p_j in the vicinity of rising and falling edges of *CLJ* signal can be interpreted as Cumulative Distribution Functions (CDFs) of signal edge positions.

As it will be shown in the next sections, this CDF can be used for evaluation of PLL-TRNG characteristics based on PLL parameters and/or for implementation of embedded statistical tests.

3. Jitter Measurement Setup

First, we wanted to observe the impact of temperature variations on the PLL-TRNG behavior. The temperature T of the FPGA was decreased by application of a freezing spray and measured using temperature sensor placed on the FPGA chip. The lowest achieved temperature was $T = -30°C$. As the FPGA chip produces some heat it warmed up by itself up to $T = +30°C$. During the measurements we tried to keep the temperature in the range of the selected value.

The measurement setup used for the experiments is depicted in Fig. 4.

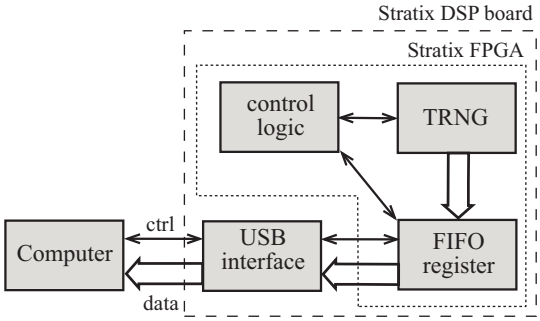

Fig. 4. Block diagram of the PLL-TRNG measurement setup.

Two similar configurations of the PLL-TRNG were chosen as objects under attack. We used Altera Stratix DSP board with EP1S25 device in both cases. The following parameters were chosen or determined by the board: $F_{CLI} = 80$ MHz, $M_{CLK} = 31, D_{CLK} = 10, M_{CLJ} = 36, D_{CLJ} = 7$. Then $F_{CLK} = 248$ MHz, $F_{CLJ} = 411$ MHz, and $K_M/K_D = 360/217$. The sampling parameters of the generator computed by (3) and (4) are: $T_Q \approx 876$ ns and $d \approx 5.6$ ps.

In order to compare the PLL-TRNG behavior for the two PLL settings, we chose the following configurations that differ in bandwidth of the loop filter:

- *Configuration A* has the filter bandwidth set automatically by the synthesizing tool (Altera Quartus [27]).
- *Configuration B* has the filter bandwidth set to *low*.

The lower is the bandwidth the better input jitter rejection can be achieved at the expense of longer locking time of the PLL. The synthesizing tool chooses the optimal bandwidth for selected signal frequencies, achieving acceptable locking time and level of input jitter filtering.

4. Measurement Results

For evaluation of the TRNG dependence on the temperature, we collected for each temperature the generator output random bitstream, as well as the internal values of samples $\widehat{q_j}$. By reordering the samples using (8) it is possible to reconstruct the waveform of the sampled clock signal CLJ and track the probabilities p_j of the samples influenced by jitter. The waveforms sampled by the generator are depicted in Fig. 5 and Fig. 6. For each sample the number of ones is counted during $N = 1000$ periods T_Q. The samples in stable

regions end up with 0 or 1000 hits, what gives the probability $p_T(j) = 0$ or $p_T(j) = 1$, where index T represents actually used temperature. The samples in edge areas (rising and falling edge), influenced by jitter, reach probability values $p_T(j)$ between these boundaries.

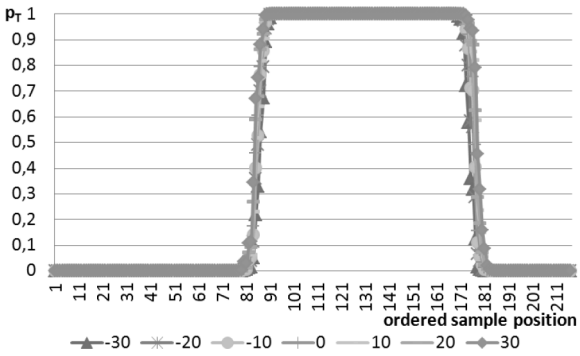

Fig. 5. Probability chart for samples of a jitter influenced clock signal *CLJ* accumulated during $N = 1000$ of T_Q periods for PLL-TRNG with PLL Configuration A with $K_D = 217$ for temperatures in range $T = (-30°C, +30°C)$.

From the charts on Fig. 5 and Fig. 6, we can see that the position of critical samples (defined as ones with $p_T(j) \notin \{0,1\}$) does not change across the range of temperatures in both configurations. The configuration A (Fig. 5) includes lower number of critical samples than the configuration B (Fig. 6) what implies lower variance (σ^2) of the jitter. We can conclude that lower bandwidth of the feedback PLL loop in the configuration B causes higher number of the critical samples. This can be explained by the fact that lower bandwidth decreases PLL output jitter and thus increases the tracking jitter. Note that the proposed method permits to measure the tracking jitter, i.e. the jitter that is used for randomness generation used by PLL-TRNG. The main advantage of this measurement is that it can be realized inside the FPGA device.

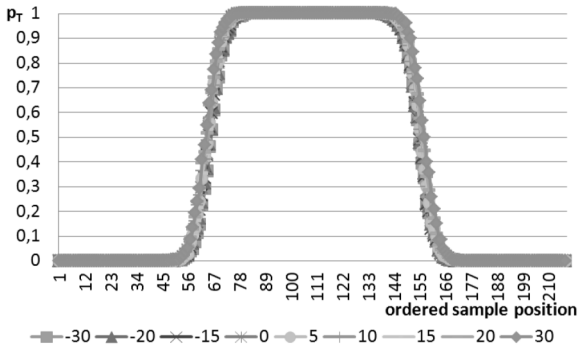

Fig. 6. Probability chart for samples of a jitter influenced clock signal *CLJ* accumulated during $N = 1000$ of T_Q periods for PLL-TRNG with PLL Configuration B with $K_D = 217$ for temperatures in range $T = (-30°C, +30°C)$.

The random sequences produced by the two generator configurations were tested by simple statistical tests defined in older version of the FIPS 140-2 standard [28]. This test suite can reveal a bias or unbalanced distribution of zeros and ones in generated sequence by application of 4 basic tests (monobit test, poker test, runs and long runs tests). If at least one test from the set does not pass, the result is denoted as *FAILED*, otherwise we put *OK* mark.

In Tab. 2 we summarize the results of statistical tests at different chip temperatures. It can be seen that while the configuration A has produced by some temperatures the sequences that did not pass the statistical tests suite FIPS 140-2, the configuration B is reliable in the whole range of temperatures. The columns with number of critical samples show the number of samples influenced by jitter. It can be observed that in the case of the configuration B, (with a low bandwidth of the loop filter), the number of influenced samples is significantly higher.

	Conf A		Conf B	
temperature T	FIPS	critical	FIPS	critical
in $°C$	tests	samples #	tests	samples #
-30	OK	26	OK	66
-20	FAILED	25	OK	64
-10	OK	24	OK	62
0	OK	24	OK	63
+10	OK	24	OK	68
+20	FAILED	22	OK	61
+30	FAILED	25	OK	60

Tab. 2. Results of FIPS 140-2 statistical tests of PLL-TRNG output and number of random samples influenced by the jitter at different chip temperatures for configurations A (default configuration) and B (low filter bandwidth).

We further investigate the number and position of critical samples for both configurations in dependency on the chip temperature. From the stochastic model of the PLL-TRNG [26] we observe that the output bias of the generated sequence is dominated by the samples with probability around the value $p = 0.5$. If we take into account only "significant" samples, i.e. samples having more than $0.1N = 100$ out of $N = 1000$ unstable values acquired during N periods T_Q (i.e. samples with probabilities $p_T > 0.1$ and $p_T < 0.9$), there will be 4–6 and 12–13 *highly critical samples* per edge for configuration A and B, respectively.

Fig. 7 and Fig. 8 show in details the region around the rising edge of the sampled waveform. We can observe how the probabilities p_T of the critical samples changes in relation to different chip temperature. In configuration A, probabilities of individual samples are spread to bigger extent. In configuration B the subsequent samples have very similar probabilities, meaning that the probability values are less sensible to temperature changes. Note that $p_T(j)$ values in the vicinity of edge shown on Fig. 7 and Fig. 8 can be interpreted also as CDF of edge position influenced by jitter. Such interpretation can be used for direct estimation of variance of edge position from acquired $p_T(j)$ values. Probability Density Function (PDF) required for variance computation can be easily computed form CDF by using numerical derivation.

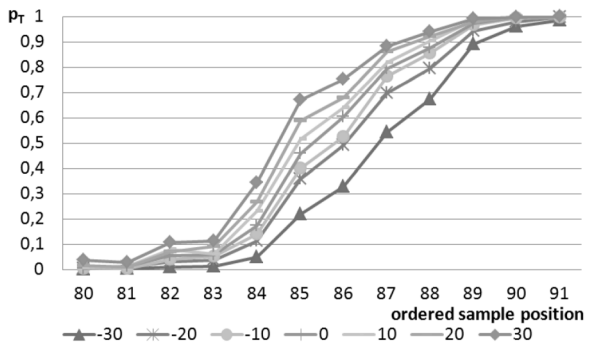

Fig. 7. Probabilities of critical samples in the area of rising edge of the sampled *CLJ* signal for PLL-TRNG with configuration *A* sampled during $N = 1000$ periods T_Q, comparison for different temperatures in range $T = (-30°C, +30°C)$ interpreted as CDF of rising edge position.

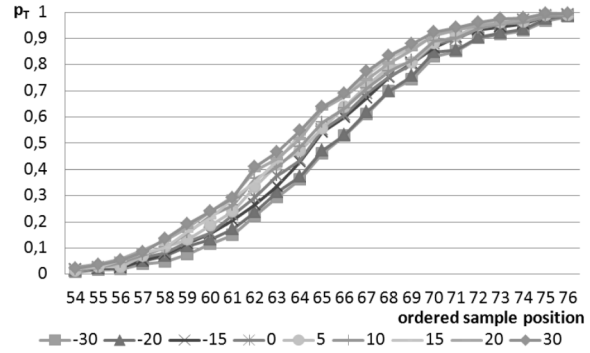

Fig. 8. Probabilities of critical samples in the area of rising edge of the sampled *CLJ* signal for PLL-TRNG with configuration *B* sampled during $N = 1000$ periods T_Q, comparison for different temperatures in range $T = (-30°C, +30°C)$ interpreted as CDF of rising edge position.

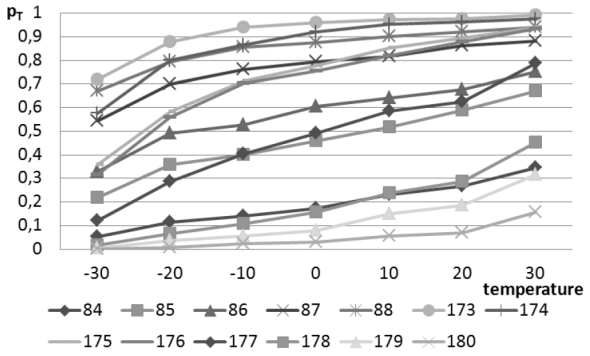

Fig. 9. Detail of probabilities of critical samples in the area of rising edge of CLJ signal for PLL-TRNG with configuration *A* sampled during $N = 1000$ periods T_Q showed for temperatures from range $T = (-30°C, +30°C)$.

In order to better visualize the changes in sampled signals in dependency on temperature we provide Fig. 9 and Fig. 10 which show in detail the dynamics of probabilities for most critical samples.

In configuration *A*, we can observe significant dependence of probability on the chip temperature. For example at position number *86* the difference in probability at minimal and maximal temperature ($p_{-30} - p_{30}$) is almost 0.5. This fact, as well as the low number of critical samples causes the instability of the PLL-TRNG.

Although the jitter is present during the whole range of the temperatures (the number of critical samples does not change), the bias of the samples changes visibly and influences the statistical parameters of the generated sequence. In a moment when all samples are strongly biased (this is the case for the temperature between $T = 20°C$ and $T = 30°C$) the output sequence is also biased and does not pass the FIPS 140-2 statistical tests suite.

The configuration *B* is more stable in changing chip temperature and the density of samples with probability close to 0.5 is much higher when comparing to the case *A*. Thanks to that, the statistical parameters of the generated sequence stay acceptable and pass all required statistical tests. The bias of particular samples is compensated by other samples in the critical area, and the final sequence is kept with very low bias.

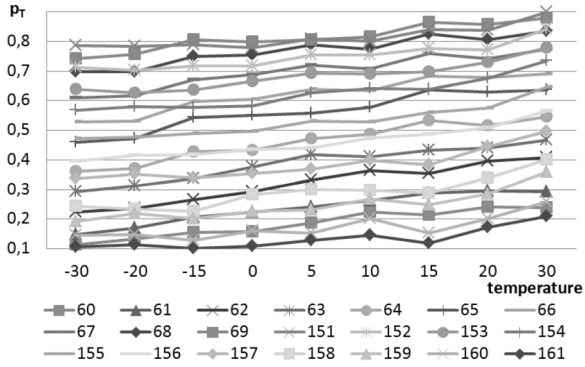

Fig. 10. Detail of probabilities of chosen critical samples in the area of rising edge of CLJ signal for PLL-TRNG with configuration *B* sampled during $N = 1000$ periods T_Q showed for temperatures from range $T = (-30°C, +30°C)$.

5. Discussion

From the observations depicted above we can conclude that variance of the jitter in the sampled signal does not change significantly. The size of deviation can be observed as number of critical samples which remains almost constant in the whole range of tested temperatures. These results have been confirmed by computing variance as presented in Tab. 3. The PDF used for variance computation was computed from CDFs shown on Fig. 7 and Fig. 8 by using standard 3-point numerical approximation of derivation. The presence of jitter represents a fundamental condition for generator proper function. Therefore, a well suited startup test for this kind of generators should include a test of number and position of critical samples.

The on-chip implementation of this test needs to include a memory block and counters which sum up for each edge position of the sampling signal the number of sampled

ones. The critical edge positions with the counter value different from 0 ($p \neq 0$) or not equal to the number of T_Q periods ($p \neq 1$) indicate the presence of the jitter. The number of critical samples must be higher than zero, but low number of samples cannot be accepted either. From empirical experiments described above we can conclude that configurations with more than 10 highly critical samples per edge behave reliably even in changing environment.

temperature T in $^\circ C$	Conf A σ^2	Conf B σ^2
-30	5.4	24.6
+20	4.9	25.4
+30	4.6	25.0

Tab. 3. Variance of rising edge position for both configurations at different temperatures obtained from PDF.

Continuous monitoring of the critical samples number allows to implement an effective online test for the discussed category of PLL-based generators. Each significant change either in position or in probability value of critical samples may have an impact on the parameters of the generated sequence and therefore should initiate an alarm signal inside the PLL-TRNG.

It is possible to estimate basic jitter parameters and draw the PDF of deviation of sampled signal edge positions from its ideal one directly from measured data. The PDFs of rising edge deviation have been obtained by numerical derivation of CDFs shown in Fig. 7 and Fig. 8 respectively. In Fig. 11 we compare the PDF histograms of rising edge deviation for PLL-TRNG working in configuration *A* and *B* with different loop filter bandwidth. Ideal edge position for analyzed configuration was estimated as the one with the highest PDF value. In both cases the jitter has a Gaussian-like distribution. As it can be observed, the configuration *A* includes jitter with lower deviation when compared to the configuration *B*.

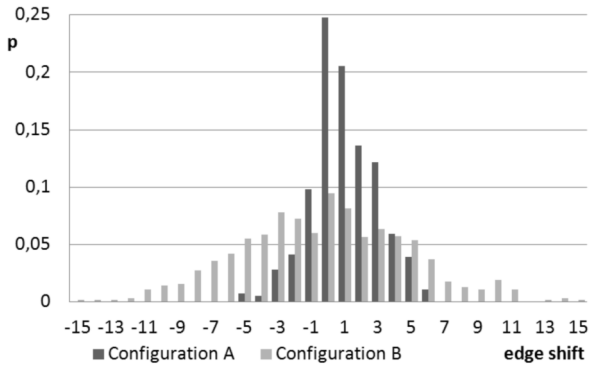

Fig. 11. Estimated PDF of the rising edge shift from its ideal position obtained by numerical derivation of CDF of the clock edge position in sampled *CLJ* signal.

From the obtained results and suggestions for PLL-TRNG design we can conclude that the design tested in [11] with parameters $K_M/K_D = 270/203$ is not suitable for usage in environment with changing temperature. The number of highly critical samples for this configuration has been confirmed by measurements to be 3–4 samples per edge. As we proposed in the suggestions above, the fact that the number of highly critical samples should be higher than 10 is important. This condition is not met in configuration published in [11] and the generator behaves as Configuration *A* during our experiments.

6. Conclusions

We analyzed an influence of changeable chip temperature on behavior of PLL-TRNG implemented in Altera FPGA. We showed what impact has the temperature on the internal PLL-TRNG signals and the quality of generated random sequence. We defined and demonstrated importance of highly critical samples for the proper operation of PLL-TRNG. As a result, we propose additional requirement for the PLL-TRNG design procedure that needs to be met in order to achieve a higher robustness of the design for practical cryptographic applications. We can conclude that PLL configurations with more than 10 highly critical samples per edge behave reliably even in changing environment. Tight probability values of critical samples assure stability of the PLL-TRNG internal environment and stable parameters of generated random values. Our proposed new interpretation of reordered probabilities of internal generator signals as a CDF of edge positions can be used as an additional tool for efficient implementation of on-line statistical test for PLL-TRNG. We will concentrate on efficient implementation of the on-line tests in our future research.

Acknowledgments

This work has been done in the frame of the Slovak scientific project VEGA 1/0045/10 2010-2011 of Slovak Ministry of Education.

References

[1] MACKAY, D. J. C. Introduction to Monte Carlo methods. In JORDAN, M. I. (ed.) *Learning in Graphical Models*. NATO Science Series, p. 175 – 204. Kluwer Academic Press, 1998.

[2] TANG, T., SIEGEL, P. H., MILSTEIN, L. B. A comparison of long versus short spreading sequences in coded asynchronous DS-CDMA systems. *IEEE Journal on Selected Areas in Communications*, 2001, vol. 19, no. 8, p. 1614 – 1624.

[3] MENEZES, A. J., VAN OORSCHOT, P. C., VANSTONE, S. A. *Handbook of Applied Cryptography*. New York: CRC Press, 1996.

[4] STANDAERT, F.-X., ROUVROY, G., QUISQUARTER, J.-J. FPGA implementations of the DES and Triple-DES masked against power analysis attacks. In *International Conference on Field Programmable Logic and Applications – FPL 2006*. Madrid (Spain), 2006.

[5] SUNAR, B., MARTIN, W. J., STINSON, D. R. A provably secure true random number generator with built-in tolerance to active at-

tacks. *IEEE Transaction on Computers*, 2007, vol. 56, no. 1, p. 109 – 119.

[6] BERNARD, F., FISCHER, V., VALTCHANOV, B. Mathematical model of physical RNGs based on coherent sampling. *Tatra Mountains Mathematical Publications*, 2010, vol. 45, p. 1 – 14.

[7] DICHTL, M., MEYER, B., SEUCHEK, H. Spice simulation of a "provably secure" true random number generator. [Online] Cryptology ePrint Archive, Report 2008/403, 2008. Available at: http://eprint.iacr.org/.

[8] BOCHARD, N., BERNARD, F., FISCHER, V. Observing the randomness in RO-based TRNG. In *International Conference on Reconfigurable Computing and FPGAs, ReConFig '09*. Quintana Roo (Mexico), 2009, p. 237 – 242.

[9] MARKETTOS, A. T., MOORE, S. W. The frequency injection attack on ring-oscillator-based true random number generators. In *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems CHES 2009*. Lausanne (Switzerland), 2009, p. 317 – 331.

[10] FISCHER, V., DRUTAROVSKY, M. True random number generator embedded in reconfigurable hardware. In *4th International Workshop on Cryptographic Hardware and Embedded Systems CHES 2002*. Redwood Shores (CA, USA), 2002, p. 415 – 430.

[11] SANTORO, R., SENTIEYS, O., ROY, S. On-line monitoring of random number generators for embedded security. In *IEEE International Symposium on Circuit and Systems ISCAS 2009*. Taipei (Taiwan), 2009, p. 3050 – 3053.

[12] Altera Corporation. *Stratix V Device Handbook, Clock Networks and PLLs in Stratix V Devices*. [Online] 2010. Available at: http://www.altera.com/literature/hb/stratix-v/stx5_51005.pdf

[13] Altera Corporation. *Cyclone IV Device Handbook, Clock Networks and PLLs in Cyclone IV Devices*. [Online] 2010. http://www.altera.com/literature/hb/cyclone-iv/cyiv-51005.pdf

[14] Xilinx Corporation. *Spartan-6 Family Overview*. [Online] 2010. Available at: http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf

[15] Xilinx Corporation. *Virtex-6 Family Overview*. [Online] 2010. Available at: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf

[16] Actel Corporation. *Using ProASICplus Clock Conditioning Circuits, Application Note AC306*. [Online] 2007. Available at: http://www.actel.com/documents/APA_PLL_AN.pdf

[17] Actel Corporation. *ProASIC3E Flash Family FPGAs, Revision 9 (datasheet)*. [Online] 2009. Available at: http://www.actel.com/documents/PA3E_DS.pdf

[18] Altera Corporation. *Using PLLs in Stratix Devices*. [Online] 2002. Available at: http://extras.springer.com/2001/978-0-306-47635-8/an/an200.pdf

[19] AMI Semiconductors Company. *XpressArray High Density 0.18 um Structured ASIC*.

[20] Altera Corporation. *Cyclone II Device Handbook, PLLs in Cyclone II Devices*. [Online] 2007. Available at: http://www.altera.com/literature/hb/cyc2/cyc2_cii51007.pdf

[21] Altera Corporation. *Cyclone III Device Handbook, Clock Networks and PLLs in Cyclone III Device Family*. [Online] 2009. Available at: http://www.altera.com/literature/hb/cyc3/cyc3_ciii51006.pdf

[22] Altera Corporation. *Stratix Device Handbook, General-Purpose PLLs in Stratix & Stratix GX Devices*. [Online] 2005. Available at: http://www.altera.com/literature/hb/stx/ch_1_vol_2.pdf

[23] Altera Corporation. *Stratix II Device Handbook, PLLs in Stratix II & Stratix GX Devices*. [Online] 2009. Available at: http://www.altera.com/literature/hb/stx2gx/stx2_sii52001.pdf

[24] DRUTAROVSKY, M., VARCHOLA, M. Cryptographic system on a chip based on Actel ARM7 soft-core with embedded true random number generator. In *Proceedings of the 11th IEEE Design and Diagnostics of Electronic Circuits and System Workshop DDECS '08*. Bratislava (Slovakia), 2008, p. 164 – 169.

[25] DAVIES, R.B. *Exclusive OR (XOR) and Hardware Random Number Generators*. Technical report. [Online] 2002. Available at: http://www.robertnz.net/pdf/xor2.pdf

[26] ŠIMKA, M., FISCHER, V., DRUTAROVSKY, M., FAYOLLE, J. Model of a true random number generator aimed at cryptographic applications. In *Proceedings of the International Symposium on Circuit and Systems – ISCAS 2006*. Island of Kos (Greece), 2006, p. 5619 – 5623.

[27] Quartus II Software. [Online] Available at: http://www.altera.com/products/software/.

[28] Federal Information Processing Standards, National Institute of Standards and Technology, U.S. Department of Commerce. *Security Requirements for Cryptographic Modules*. 2001. NIST FIPS PUB 140-2.

About Authors...

Martin ŠIMKA was born in 1979 in Košice. He received his MSc degree in electronics and telecommunications in 2002 after defending his Master's Thesis on implementation of an arithmetic coprocessor for modular multiplication. Recently he submitted a PhD thesis at Technical University of Košice on topic of analysis and implementation of selected blocks for public-key cryptographic systems on FPGAs. He currently works for Polish Security Printing Works (Warsaw, Poland) as cryptography specialist and architect of secure IT systems.

Miloš DRUTAROVSKÝ was born in 1965 in Prešov, Slovak Republic. He received the MSc degree in radioelectronics and PhD degree in electronics from Technical University of Košice, Slovak Republic, in 1988 and 1995, respectively. He defended his habilitation work - Digital Signal Processors in Digital Signal Processing in 2000. He is currently working as an Associated Professor at the Department of Electronics and Multimedia Communications, Technical University of Košice. His current research interests include applied cryptography, digital signal processing, and algorithms for embedded cryptographic architectures.

Viktor FISCHER received his MSc and PhD degrees in electronics from Technical University of Košice, Slovak Republic, in 1981 and 1991, respectively. From 1982 to 1991 he was an Assistant Professor at the Department of Electronics, Technical University of Košice. From 1991 to 2006, he was working at the Jean Monnet University of Saint-Etienne, France, as an Invited Professor in electronics and computer science and as an expert in design of secure embedded systems with the MICRONIC company in Slovakia. Since 2006, he is a full-time professor at the Jean Monnet University and he manages the research team Secured Embedded Systems in the Hubert Curien Laboratory, UMR 5516 CNRS/University of Saint-Etienne. His research is oriented towards the design of secured hardware implementations of cryptographic primitives and data security systems and especially of True Random Number Generators.