On-chip jitter measurement for true random number generators

Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens and Ingrid Verbauwhede
COSIC, KU Leuven
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
Email: firstname.lastname@esat.kuleuven.be

***Abstract*—Applications of true random number generators (TRNGs) span from art to numerical computing and system security. In cryptographic applications, TRNGs are used for generating new keys, nonces and masks. For this reason, a TRNG is an essential building block and often a point of failure for embedded security systems. One type of primitives that are widely used as source of randomness are ring oscillators. For a ring-oscillator-based TRNG, the true randomness originates from its timing jitter. Therefore, determining the jitter strength is essential to estimate the quality of a TRNG. In this paper, we propose a method to measure the jitter strength of a ring oscillator implemented on an FPGA. The fast tapped delay chain is utilized to perform the on-chip measurement with a high resolution. The proposed method is implemented on both a Xilinx FPGA and an Intel FPGA. Fast carry logic components on different FPGAs are used to implement the fast delay line. This carry logic component is designed to be fast and has dedicated routing, which enables a precise measurement. The differential structure of the delay chain is used to thwart the influence of undesirable noise from the measurement. The proposed methodology can be applied to other FPGA families and ASIC designs.**

I. Introduction

The security of cryptographic algorithms relies on the uniformity and unpredictability of keys, challenges, nonces and initialization vectors. In addition, many techniques for improving resistance against side-channel attacks, require unpredictable, uniformly distributed numbers for generating masks and secret shares. When security primitives are implemented on hardware platforms, the source of randomness has to be implemented as well. Hardware modules called true random number generators (TRNGs) exploit unpredictable physical processes such as timing jitter or thermal noise, to produce a random digital output.

Due to their critical role in security systems, TRNGs are the subject of strict evaluation. Since the attack [1] on the Motorola TRNG [2] in 2003, both industry and academia raised concerns about the security evaluation of TRNG designs. Simply passing the statistical tests is no longer considered sufficient to prove the security (the Motorola TRNG was able to pass the DIEHARD [3] statistical tests for randomness), and a formal evaluation of the security is required. The special publication SP 800-90B [4] by the National Institute for Standards and Technology (NIST) presents requirements for the design and evaluation of TRNGs. One of the requirements is the theoretical rationale of the randomness generation process. The AIS-31 standard [5] of the German Federal Office for Information Security (BSI) proposes even more strict evaluation criteria based on the stochastic model of the entropy source. Both evaluation standards require the TRNG designer to provide the entropy claim – i.e. the estimation of the lower bound on the entropy produced by the generator (NIST requires the estimation of the min-entropy, and AIS-31 requires the estimation of the Shannon entropy). In order to estimate this lower bound, the designer needs to measure the parameters of the underlying physical process used for randomness generation. We refer to these as *platform parameters*.

Many TRNGs implemented on FPGA platforms rely on the timing jitter in free-running ring oscillators (ROs) as a source of randomness [6], [7], [8], [9], [10]. The strength of the white noise accumulated in these ring oscillators is the critical platform parameter in these designs. It has to be measured at design time in order to estimate the entropy level of the generated output. Incorrect measurement of this parameter can lead to an overestimation of the generated entropy, which results in a security claim that is higher than the actual security provided by the system.

In this paper we propose a methodology for measuring the timing jitter of free-running ring oscillators on FPGA platforms. This methodology is designed to measure the strength of the white noise while filtering out contributions from other noise sources such as flicker noise and power supply variations.

In the remainder of this paper, Section II provides background on TRNG evaluation requirements. In Section III, we describe our carry-chain based methodology for jitter measurement on FPGA platforms. The proposed methodology was applied on two implementation platforms, namely Xilinx Spartan-6 [11] and Intel Cyclone-IV FPGAs [12], and the experimental results are presented in Section IV. Section V provides comparison to related work. We give conclusions in Section VI.

This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work is supported in part by the Flemish Government through G.0130.13N and FWO G.0876.14N, the Hercules Foundation AKUL/11/19, and through the Horizon 2020 research and innovation programme under grant agreement No 644052 HECTOR and Cathedral ERC Advanced Grant 695305. B. Yang is supported in part by the Scholarship from China Scholarship Council (No.201206210295).

II. BACKGROUND

A. TRNG evaluation

Designing a TRNG is different from conventional digital circuit design because a mathematical model of the TRNG is required to evaluate the security of the generated output. This mathematical model is used to estimate the lower bound on the entropy for given values of the design parameters and the platform parameters. Design parameters are all variables that are controlled by the designer, e.g. the sampling frequency, the length of the ring oscillator or the number of ring oscillators in the design. Platform parameters are the variables that affect the entropy of the TRNG but that are not controlled by the designer, for example, noise strength or delays of the logic elements. These parameters depend only on the implementation platform, so they have to be measured at design time. The description of the measurement methodology is an obligatory part of the TRNG certification procedure.

B. Jitter in Ring Oscillators

Free-running ring oscillators are implemented by connecting any number of delay elements (including an odd number of inverting elements) into a loop configuration. The delay of each logic element in a ring oscillator consists of an average component and a variable component. Several types of noise could affect the variable component:

- Global noise from the supply voltage variations. This noise affects all components on the platform.
- Environment noise such as slow changes of the operating conditions (e.g. temperature changes).
- Correlated noise such as flicker noise and telegraph noise.
- White noise which is also called Gaussian noise.

Mathematical models of TRNG designs usually rely only on the Gaussian noise because other noise sources are either difficult to model and analyze or can be manipulated by an attacker. These noise sources are usually treated as deterministic.

The period of the ring oscillator is given by:

$$T_i = T_0 + T_{Global} + T_E + T_{Corr} + T_{Gauss}\,, \qquad (1)$$

where T_{Gauss} is a random variable with standard deviation σ_G, T_0 is the average period and T_{Global}, T_E and T_{Corr} are the contributions of the global noise, environment noise and the correlated noise sources, respectively.

For a RO that is running for time t_m, the following relation holds:

$$t_m = n \cdot T_0 + t_D + t_G\,, \qquad (2)$$

where n is the number of full oscillations made during the $(0, t_m)$ time interval, t_D is the deterministic noise accumulated during this interval and t_G is the accumulated Gaussian noise. Let σ_m^2 denote the variance of t_G. Then the following holds:

$$\sigma_m^2 = \sigma^2(t_G) = \frac{\sigma^2(T_{Gauss})}{T_0} \cdot t_m\,. \qquad (3)$$

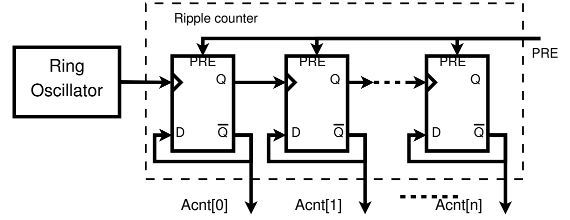

Fig. 1. Ring oscillator period measurement setup.

The parameter σ_m^2/t_m characterizes the rate of the Gaussian jitter accumulation. This quantity is the most important platform parameter for RO-based TRNGs because it is the only parameter that characterizes the randomness generation process.

III. METHODOLOGY

A procedure for accurate and precise characterization of the white noise should satisfy the following requirements:

- All measurements should be performed on-chip – only the digital data should be sent to the output for post-processing.
- Digitization should be performed using a small quantization step to reduce the quantization noise.
- All measurements should be differential in order to reduce the influence of the global noise sources.
- The jitter accumulation time should be as short as possible in order to minimize the effect of the low frequency noise sources.

In this paper, we propose a jitter measurement methodology. This method is based on time-to-digital conversion, which is implemented using the carry chain primitives. Carry chain primitives are available on most commercial FPGAs and they are designed to constitute high-speed adders and multipliers. They can also be configured to operate as high-resolution time-to-digital converters.

The procedure of the proposed measurement methodology consists of three steps:

1) T_0 measurement.
2) Delay line characterization.
3) Differential TDC measurement.

In continuation of this section we describe these three steps in more detail.

A. T_0 measurement

For measuring the average period T_0 of a ring oscillator, the architecture shown in Figure 1 is proposed. The output signal of the ring oscillator under evaluation connects to an n-bit ripple counter to count the number of $0 \rightarrow 1$ transitions. The ripple counter consists of n positive-edge-triggered D flip-flops.

To perform an experiment, all flip-flops in the ripple counter are preset to 1 before enabling the ring oscillator. Then, the ring oscillator is enabled for N_{acc} system clock cycles. The

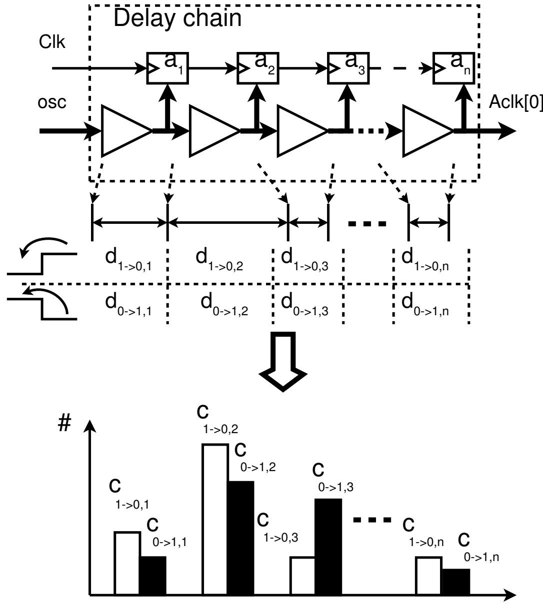

Fig. 2. Monte Carlo method for delay measurement.

system frequency is denoted as f_{system}. After N_{acc} cycles, the ripple counter result $Acnt$ is stored.

This experiment is repeated N_{exp} times. At the beginning of every experiment, the ripple counter is preset, and the value of it is stored as $Acnt_i$. In order to cancel out the influence of low frequency noise over the ring oscillator period, the average of all stored $Acnt_i$ is computed. In the last step, T_0, the average period of the ring oscillator, can be computed as:

$$T_0 = \frac{N_{acc} \cdot N_{exp}}{f_{system} \cdot \sum_{i=1}^{N_{exp}} Acnt_i}. \quad (4)$$

B. Delay line characterization

In our proposed jitter measurement method, two delay line based time-to-digital converters are used. For a precise measurement, the steps of elements in the delay line need to be small and accurate. Therefore, the delay line should be characterized first.

We propose a Monte Carlo experiments based method to evaluate the delay line. As shown in Figure 2, a signal osc generated by a ring oscillator propagates through an n elements delay chain under evaluation. The average period of the signal osc is T_0. The length of the delay line should be able to cover more than one and a half full periods of the signal osc. An independent system clock Clk is used to sample this delay line N_{exp} times. The captured states (bit vectors) of the delay line are denoted as:

$$m_i = (a_{i,0}, a_{i,1} \dots a_{i,n}),\, a_{i,j} \in \{0, 1\}, \quad (5)$$

where $a_{i,0}$ is the most recent bit and $a_{i,n}$ is the earliest bit in the i_{th} sampled sequence.

The bit vector m_i consists of consecutive $'1'$s and $'0'$s, representing the signal osc. If the delays of all n elements are the same, the probabilities of having a transition are equal at every position. However, due to the nonlinearity of the delay line, the probability of having a transition is higher for a delay element with larger delay.

We denote the delay of a $1 \rightarrow 0$ and $0 \rightarrow 1$ transitions for the j_{th} element with $d_{1\rightarrow 0,j}$ and $d_{0\rightarrow 1,j}$, respectively. The goal of characterizing the delay line is to estimate the value of $d_{1\rightarrow 0,j}$ and $d_{0\rightarrow 1,j}$.

$c_{1\rightarrow 0,j}$ and $c_{0\rightarrow 1,j}$ are defined as:

$$\begin{aligned} c_{1\rightarrow 0,j} &= \#(m_i),\, for\, (a_{i,j}, a_{i,j+1}) = (0, 1), \\ c_{0\rightarrow 1,j} &= \#(m_i),\, for\, (a_{i,j}, a_{i,j+1}) = (1, 0). \end{aligned} \quad (6)$$

$c_{1\rightarrow 0,j}$ is the number of samples that have the subvector $(0, 1)$ at position $(j, j + 1)$.

The proportion of any two elements from a union set of $c_{0\rightarrow 1,j}$ and $c_{1\rightarrow 0,j}$ is equal to the proportion of the corresponding delays of a union set of $d_{0\rightarrow 1,j}$ and $d_{1\rightarrow 0,j}$.

We define the following functions for the summation of consecutive $c_{0\rightarrow 1,j}$ and $c_{1\rightarrow 0,j}$:

$$\begin{aligned} S_{1\rightarrow 0}(i, j) &= \sum_{x=i}^{j} c_{1\rightarrow 0,x} \\ S_{0\rightarrow 1}(i, j) &= \sum_{x=i}^{j} c_{0\rightarrow 1,x}. \end{aligned} \quad (7)$$

We denote $e_{1\rightarrow 0,j}$ and $e_{0\rightarrow 1,j}$ as the edge positions of the j_{th} latest $1 \rightarrow 0$ and $0 \rightarrow 1$ transitions in the snapshot, respectively. Therefore, $(a_{i,e_{1\rightarrow 0,1}}, \dots, a_{i,e_{0\rightarrow 1,1}}, \dots, a_{i,e_{1\rightarrow 0,2}-1})$ represents the first full period of osc captured in this snapshot.

In the next step, the function $W(m_i)$ is defined and used to represent the period of the first captured full period of the signal osc:

$$\begin{aligned} W(m_i) = S_{0\rightarrow 1}(&e_{1\rightarrow 0,1}, e_{0\rightarrow 1,1} - 1) \\ &+ S_{1\rightarrow 0}(e_{0\rightarrow 1,1}, e_{1\rightarrow 0,2} - 1). \end{aligned} \quad (8)$$

And, W_{cnt} is the average $W(m_i)$ value of N_{exp} experiments:

$$W_{cnt} = \frac{\sum_{i=1}^{N_{exp}} W(m_i)}{N_{exp}}. \quad (9)$$

The delays $d_{1\rightarrow 0,j}$ and $d_{0\rightarrow 1,j}$ of every element of the delay line can be estimated by:

$$\begin{aligned} d_{1\rightarrow 0,j} &= \frac{c_{1\rightarrow 0,j}}{W_{cnt}} \cdot T_0 \\ d_{0\rightarrow 1,j} &= \frac{c_{0\rightarrow 1,j}}{W_{cnt}} \cdot T_0. \end{aligned} \quad (10)$$

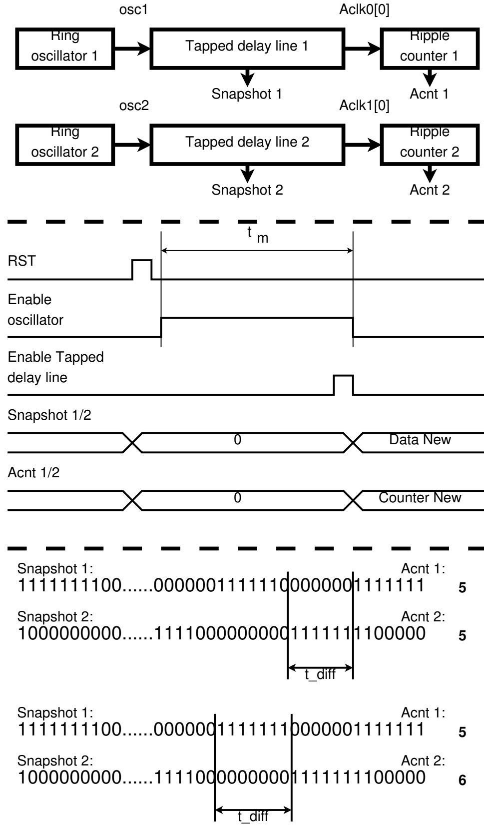

Fig. 3. Setup of the differential jitter measurement.

C. Differential TDC measurement

Figure 3 depicts the architecture proposed for jitter measurement. This architecture consists of two time-to-digital convertors (TDCs), two ring oscillators and two ripple counters. Signals $osc1$ and $osc2$ are the outputs of the corresponding free running oscillators. Then $osc1$ and $osc2$ propagate through two n-stages tapped delay lines. The rising transitions in $osc1$ and $osc2$ are counted by two ripple counters.

To estimate the jitter strength, we first measure the timing phase difference of the corresponding edges in Snapshot 1 and 2 after an accumulation time t_m. This measurement is repeated N_{exp} times. The number of $1 \rightarrow 0$ transitions in Snapshot 1 and 2 are denoted as l_{S1} and l_{S2} respectively. We denote $g_{1\rightarrow 0,i}$ as the position of the i_{th} latest $1 \rightarrow 0$ transition in the bit vector. Vectors m_{S1} and m_{S2} are captured sequences in Snapshot1 and Snapshot2. The timing of the i_{th} $1 \rightarrow 0$ transition in sequence m can be computed by:

$$T(m, i_{th}) = \sum_{i=g_{1\rightarrow 0,i_{th}}}^{n} d_i + T_0 \cdot Acnt. \tag{11}$$

The phase difference of the corresponding edges for j_{th} experiment is computed by:

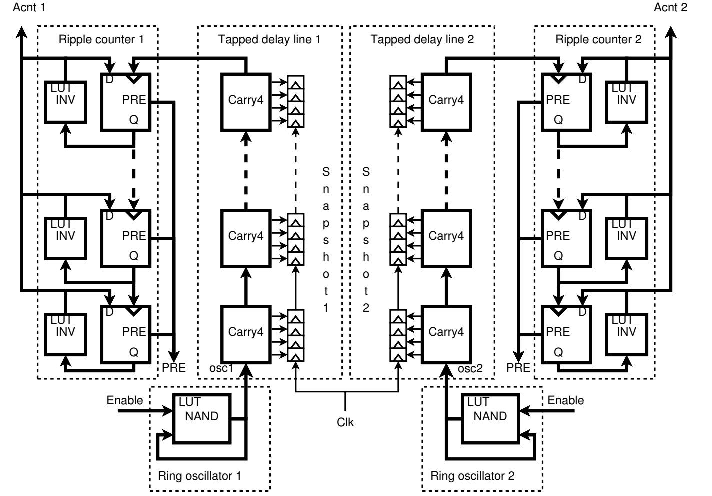

Fig. 4. Implementation of the jitter measurement on Xilinx platform.

$$t_{diff,j} = \begin{cases} T(m_{S1}, l_{S1}) - T(m_{S2}, l_{S2}), & for\ Acnt1 = Acnt2 \\ T(m_{S1}, l_{S1}) - T(m_{S2}, l_{S2}-1), & for\ Acnt1 = Acnt2 + 1 \\ T(m_{S1}, l_{S1}-1) - T(m_{S2}, l_{S2}), & for\ Acnt1 = Acnt2 - 1 \end{cases} \tag{12}$$

Then, the average value of $t_{diff,j}$ is:

$$\bar{t}_{diff} = \frac{\sum_{j=1}^{N_{exp}} t_{diff,j}}{N_{exp}}. \tag{13}$$

The expected value of $\bar{t}_{diff}$ is around 0. However, it may be different due to the effect of manufacture process variations.

In the last step, jitter accumulated during time t_m can be computed by:

$$\sigma_m(t_m) = \sqrt{\frac{\sum_{i=1}^{N_{exp}} (t_{diff,i} - \bar{t}_{diff})^2}{k-1}} \tag{14}$$

The parameter σ_m^2/t_m characterizes the rate of the Gaussian jitter accumulation.

IV. Experiments

The proposed method for jitter measurement is applied to two case studies on two different FPGA platforms.

A. Xilinx FPGA

Figure 4 shows the implementation of the proposed jitter measurement on a Xilinx Spartan-6 FPGA. Ring oscillators are implemented using a single look-up table (LUT). CARRY4 primitives are used to implement tapped delay lines. In total, 64 CARRY4 primitives are cascaded together to form a 256-stages delay line.

The frequency of the system clock Clk is set to $100MHz$. We select parameters $N_{acc} = 2^{13}$ and $N_{exp} = 100000$ for the T_0 measurement. The collected $\sum_{x=1}^{N_{exp}} Acnt_x$ are 6482648529 for ring oscillator 1 and 6500063667 for ring oscillator 2.

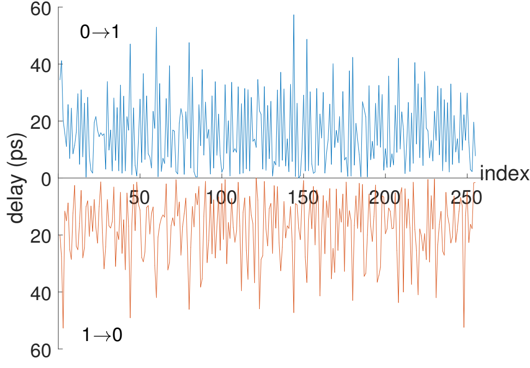

Fig. 5. Delays of each carry bit after characterization.

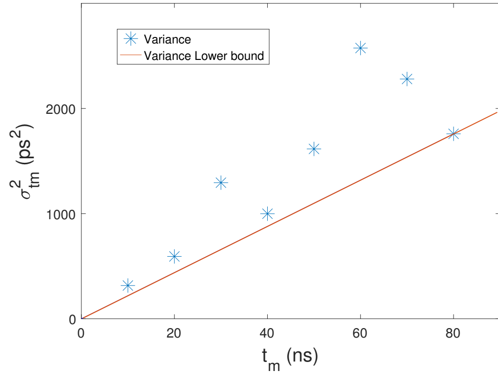

Fig. 6. Variance result of t_{diff}.

Therefore, the computed T_0 of ring oscillator 1 and 2 are $1.2637ns$ and $1.2603ns$ respectively.

While characterizing the delay line, bubbles appear in the captured snapshots. In [13], this problem is handled by counting the number of $'1'$s and $'0'$s. The origin of the bubble might be the Carry Look Ahead Logic structure and the process variation during fabrication. In total, we found 8 types of bubbles. All of them can be easily corrected by reordering. After reordering, the $1 \rightarrow 0$ and $0 \rightarrow 1$ delays of all 256 stages are characterized and shown in Figure 5. The average delay of those stages is $16.8ps$.

In each measurement, the oscillators were enabled for t_m ns before the snapshots were taken. Snapshots and the counter values were transferred to the PC for post-processing. Based on the 100000 measurements for each, the computed variance of t_{diff} for $t_m = 20$ ns, 40 ns, 60 ns and 80 ns, are shown in Figure 6. The lower bound noise strength is obtained at $t_m = 80$ ns. The computed standard deviation $\sigma(t_{diff}) = 41.9413$ ps. Then, the computed noise strength is:

$$\frac{\sigma_m^2}{t_m} = 21.98\ fs\,. \tag{15}$$

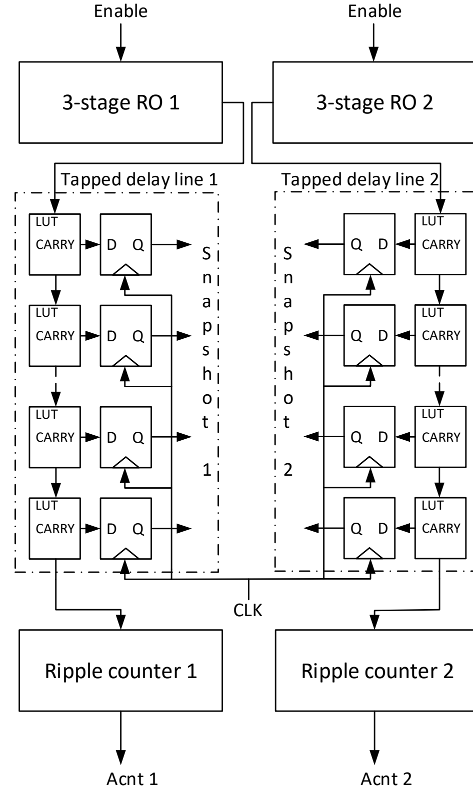

Fig. 7. Implementation of the jitter measurement on Cyclone IV.

B. Intel FPGA

The jitter measurement circuit for the Intel (Cyclone IV) FPGA is implemented in an analogous way as on the Xilinx FPGA. The architecture is shown in Figure 7. Stages of the ring oscillator are instantiated in the LUTs of neighbouring LEs (Cyclone IV Logic Element) by using low-level primitives. Fast (tapped) delay lines are implemented using small 3-input LUTs, which are specially dedicated for fast carry generation [12]. The delay lines are placed in LEs right below the ring oscillator by using placement constraints. Ripple counters are then placed below these delay lines in order to minimize interconnection delays. Bubbles in the delay line appear only near the signal edges, meaning that they originate from uneven propagation delays in the FPGA structure and timing violations on delay line capturing registers.

The average periods of ring oscillator 1 and 2 are 2394.2712 ps and 2529.605 ps. In each measurement, the oscillators were enabled for $t_m = 40$ ns before the snapshots were taken. The snapshots and the counter values were transferred to the PC for post-processing. Based on the 100000 measurements, the computed standard deviation of t_{diff} is $\sigma(t_{diff}) = 5.9739ps$. The computed noise strength is:

$$\frac{\sigma_m^2}{t_m} = 0.89\ fs\,. \tag{16}$$

V. Discussion

While the oscillator period T_0 can be estimated accurately using very simple counter-based methods, estimating the strength of the jitter is a more challenging task. Jitter in ring oscillators is often studied from the reliability perspective, where the goal is to provide the upper boundaries of jitter strength. However, for TRNG applications, the goal is to derive the lower bound on jitter strength in order to provide a conservative estimate of the output entropy. For this reason, different measurement procedures that isolate the contribution of the white noise need to be developed. In this section we compare the proposed jitter measurement methodology with other, more commonly used methods, and discuss the consequences of choosing a wrong jitter estimation.

The most straightforward solution for jitter measurement is to connect the oscillator output to the output pin and to analyze the signal using an oscilloscope. This approach was taken in [14]. The problem with this strategy is that the oscillator signal is corrupted on the pads, pins and measurement probe. In addition, commonly used oscilloscopes have a bandwidth of up to $4\ GHz$, which is not sufficient to analyze the Gaussian jitter which is in the order of picoseconds. The results presented in [14] show that oscillators with a period ranging from $2\ ns$ to $10\ ns$, all have a comparable Gaussian jitter. This result is not in line with the theory on the white noise which is a strong indicator that the measurement setup introduces some systematic error.

The simplest on-chip method for jitter evaluation consists of counting the toggles of the ring oscillator output over a fixed amount of time and computing the jitter based on the variance of the obtained results. This methodology is used in [15] and [16]. We will refer to this technique as *the counter methodology*. The errors of this methodology are caused by the quantization noise and the low frequency noise. If the measurement time is too short, the quantization noise of the counter is dominant and it is impossible to measure the Gaussian jitter precisely. For longer measurement times the quantization noise is reduced but the contribution of the low-frequency noise becomes dominant. In either case, the Gaussian noise is significantly overestimated.

To illustrate the importance of differential measurements, a single delay line based measurement method was used for comparison. We refer to this method as *Single Delay Line* methodology. Jitter is computed using only the data from $Snapshot1$. The jitter is computed from the variance of the timing of the last rising edge $T(m_{S1}, l_{S1})$. This method reduces the quantization noise and low frequency noise but it doesn't filter out the global noise from the power supply.

Table I summarizes the jitter results measured using three different methodologies on a Xilinx Spartan-6 FPGA. The proposed *Differential Delay Line* methodology filters out the quantization noise, low frequency noises and the global noise and thus results in the lowest jitter estimate. The single delay line method, doesn't filter out the global noise and thus overestimates the jitter variance by a factor of 5. The counter methodology grossly overestimates the jitter strength.

TABLE I
Jitter measurement results using three different methodologies.

Method	Counter	Single Delay Line	Differential Delay Line
σ_G^2/T_0	$11\ ps$	$154\ fs$	$22\ fs$
σ_G	$118\ ps$	$13.9\ ps$	$5.26\ ps$

Another related work is the coherent sampling method for jitter measurement presented in [17]. We don't see any problems with this methodology and the reported results are similar to those obtained by the *Differential Delay Line* method.

VI. Conclusions

Accurate jitter characterization is an essential step in designing ring-oscillator-based TRNGs for FPGAs. The measurement setup and the methodology for jitter characterization on FPGA platforms have been described. The proposed setup utilizes fast carry chain elements that are available on most FPGAs. Differential measurements are used to reduce the influence of the global noise sources. The methodology has been applied to measure jitter on Xilinx Spartan-6 and Intel Cyclone IV FPGAs.

References

[1] M. Dichtl, "How to predict the output of a hardware random number generator," in *CHES*, 2003, pp. 181–188.

[2] T. E. Tkacik, "A hardware random number generator," in *CHES*, 2002, pp. 450–453.

[3] G. Marsaglia, "DIEHARD battery of tests of randomness."

[4] E. Barker and J. Kelsey, "Recommendation for the entropy sources used for random bitgeneration," ser. NIST DRAFT Special Publication 800-90B, 2016.

[5] W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators," ser. BDI, Bonn, 2011.

[6] 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*, vol. 56, no. 1, pp. 109–119, Jan 2007.

[7] D. Schellekens, B. Preneel, and I. Verbauwhede, "FPGA vendor agnostic true random number generator," in *FPL*, 2006, pp. 1–6.

[8] K. Wold and C. H. Tan, "Analysis and enhancement of random number generator in FPGA based on oscillator rings," in *ReConFig*, 2008, pp. 385–390.

[9] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert, "A very high speed true random number generator with entropy assessment," in *CHES*, 2013, pp. 179–196.

[10] V. Rožić, B. Yang, W. Dehaene, and I. Verbauwhede, "Highly efficient entropy extraction for true random number generators on FPGAs," in *DAC*, 2015, pp. 116:1–116:6.

[11] Xilinx, "Spartan-6 FPGA Configurable Logic Block," February 2010.

[12] Intel, "Cyclone IV Device Handbook," March 2016.

[13] C. Favi and E. Charbon, "A 17ps time-to-digital converter implemented in 65nm fpga technology," in *FPGA*, 2009, pp. 113–120.

[14] O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, "A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices," in *FPL*, 2016, pp. 1–10.

[15] B. Valtchanov, A. Aubert, F. Bernard, and V. Fischer, "Modeling and observing the jitter in ring oscillators implemented in FPGAs," in *DDECS*, 2008, pp. 158–163.

[16] Y. Ma, J. Lin, T. Chen, C. Xu, Z. Liu, and J. Jing, "Entropy Evaluation for Oscillator-Based True Random Number Generators," in *CHES*, 2014, pp. 544–561.

[17] D. Lubicz and N. Bochard, "Towards an oscillator based TRNG with a certified entropy rate," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 1191–1200, 2015.