New High Entropy Element for FPGA Based True Random Number Generators

Michal Varchola and Milos Drutarovsky

Department of Electronics and Multimedia Communications,
Technical University of Kosice,
Park Komenskeho 13, 041 20 Kosice, Slovak Republic
michal@varchola.com, Milos.Drutarovsky@tuke.sk

Abstract. We demonstrate a new high-entropy digital element suitable for True Random Number Generators (TRNGs) embedded in Field Programmable Gate Arrays (FPGAs). The original idea behind this principle lies in the randomness extraction on oscillatory trajectory when a bi-stable circuit is resolving a metastable event. Although such phenomenon is well known in the field of synchronization flip-flops, this feature has not been applied for TRNG designs. We propose a new bi-stable structure – Transition Effect Ring Oscillator (TERO) where oscillatory phase can be forced on demand and be reliably synthesized in FPGA. Randomness is represented as a variance of the TERO oscillations number counted after each excitation. Variance is highly dependent on the internal noise of logic cells and can be used easily for reliable instant inner testing of each generated bit. Our proposed mathematical model, simulations and hardware experiments show that TERO is significantly more sensitive to intrinsic noise in FPGA logic cells and less sensitive to global perturbations than a ring oscillator composed from the same elements. The experimental TERO-based TRNG passes NIST 800-22 tests.

Keywords: TRNG, oscillatory metastability, randomness extraction, inner testability.

1 Introduction

Almost each cryptographic system contains a Random Number Generator (RNG) that produces random values for underlying algorithms. Random numbers are essential elements for secure transactions and therefore they should meet the highest strict requirements – they should be unpredictable, uniformly distributed on their range and independent [13].

RNGs can be divided into two main subgroups [9]: Pseudo RNG (PRNG) and True RNG (TRNG). The output of a PRNG is mathematically defined and all of its entropy is given by the random seed. On the other hand, entropy of a TRNG is increased by each generated bit. The TRNG operation is usually based on certain physical sources of entropy (e.g. thermal noise, timing jitter) that is present in modern electronic devices.

S. Mangard and F.-X. Standaert (Eds.): CHES 2010, LNCS 6225, pp. 351–365, 2010.
© International Association for Cryptologic Research 2010

Field Programmable Gate Arrays (FPGAs) are a popular implementation platform for modern crypto-systems thanks to their reconfigurability [24]. Weak or obsolete cryptographic protocols or algorithms can be updated easily even in devices deployed in a hostile environment. Thus users and FPGA devices can better resist security treats. Moreover, an entire system should be implemented in the same chip due to security reasons. Due to mentioned security considerations, research on TRNGs for the FPGAs is still an area of active research [9].

Recent TRNGs for FPGAs employ two main randomness sources. First, timing jitter of Ring Oscillators (ROs) [20,7] or Phase Locked Loops (PLLs) [6] and, second metastability of logic cells [5,21,22]. A comprehensive survey of various TRNG principles was reported in [9]. However, serious disputes on reliability of RO-based TRNG [20] led to a chain of papers [17,3,4,19,23,1] reporting various merits of it. Designers should also consider other issues such as frequency injection attack [12] or TRNG evaluation methodology according [10].

Deep study on metastability was done in [8,15] where the main focus was synchronization issues rather than TRNGs. Short-time metastability of a bistable structure was forced by critical combination of input signals. It was noted that even a small perturbation can cause escape from this state. The resulting logic state and trajectory of approaching it was analyzed as well. The resulting state and/or resulting trajectory can possess random properties such as a jitter of temporarily oscillatory trajectory. Forcing metastability on demand is not trivial and represents a great challenge for synthesis in FPGA fabric. However, most published TRNG designs are targeted to ASIC technology and extract randomness from the "final logic state" after a metastable event.

The most recent design [22] uses a metastabe RO, where each inverter is in short circuit to first reach a metastable state. After a while, inverters are switched to the single chain in order to form a RO. Metastable issues provide random starting conditions for the RO. Oscillation of RO and signal sampling by a D-Flip-Flop (DFF) are used for a resolution of the foregoing metastable event. The proof that authors of [22] reached a metastable state in the FPGA is questionable. As evidence, they provide oscilloscope waveforms that in our opinion cannot be acquired reliably from internal FPGA gates.

Each approach for a metastable TRNG [22,21,5] is based mainly on forcing a system to a metastable state and then evaluating the state where the system converges when the metastability phase is over. However, the phase of convergence towards stable state by a temporally oscillatory trajectory was still neglected as a randomness extraction mechanism. As it will be shown in this paper, such phase is worthy to consider for random bit generation. We must point out that the main focus of the paper is to introduce features of the new high entropy element for FPGA-based TRNGs rather than present a complete TRNG design.

Paper is organized as follows: Section 2 brings design goals of the new entropy element. Section 3 introduces the new entropy element and its mathematical model accompanied by SPICE and VHDL macro-model simulations. A hardware implementation is analyzed in the Sect. 4. Experimental results are presented in the Sect. 5. Conclusion and future work is given in the Sect. 6.

2 New Entropy Element Design Goals

Each design of TRNG based on logic cells has pros and cons. Going by the recent state of the art [9], there is still a gap for the TRNG based on better entropy element, that is possible to synthesize in the FPGAs fabric. The design goals for such an element are:

- sufficiently higher entropy rate than previous RO based designs,
- lower sensitivity on global interference and working conditions than previous RO based designs,
- ability to extract reliably intrinsic noise generated by logic elements,
- clear description of the mathematical model acceptable to the wide scientific community,
- inner testability feature in order to detect instantly when the entropy source is out of order and/or has weak statistical properties [18],
- ability to restart the element before each random bit generation period in order to utilize the stateless entropy concept [2],
- ability for several entropy elements operating independently and in parallel in order to place them into the same FPGA for enhancing statistical parameters and/or increasing the bit-rate,
- usage of least number of logic elements all implemented in the single block of logic to minimize signal paths, minimize interference, minimize resources utilization, and decrease the power consumption,
- element structure should have simple place and route strategies and clear recommendations on how to synthesize the structure,
- ability to utilize combinations of multiple known principles of randomness extraction, i.e. variation of time delay and the metastability phenomena.

3 Transition Effect Ring Oscillator

A novel structure capable of extracting noise from logic cells is depicted in Fig. 1. Proposed structure was optimized for implementation on a single Spartan 3E Complex Logic Block (CLB). An entire set of experiments has been carried out using Xilinx Spartan 3E FPGA Starter Kit [25] for the purpose of this paper research goals. Its physical behavior in FPGA, including the shape of control waveforms, is shown in an oscilloscope screen-shot in Fig. 2. The $\text{XOR}_1 - \text{AND}_1 - \text{XOR}_2 - \text{AND}_2$ loop begins to oscillate at each edge of the *ctrl* signal. This effect is a “transition” of the loop and therefore this structure will be referred the Transition Effect Ring Oscillator (TERO) in this paper. When TERO circuit operates in the topology shown in Fig. 1 its operation will be referred as a “TERO mode” or just TERO. When $ctrl = '1'$ for XOR_1 and $ctrl = '0'$ for XOR_2 constantly the structure will behave as an RO of one inverting and three non-inverting elements. This operation will be denoted as an “RO mode” in text for the purpose of comparing TERO mode and RO mode features.

TERO operates as follows: The XOR gates act as inverters or buffers when the $ctrl = '0'$ or $ctrl = '1'$ respectively. In other words, loop incorporates two buffers and two inverters (*ctrl* $= '0'$) or just four buffers (*ctrl* $= '1'$) when assuming *rst* $= '0'$. Loop does not satisfy an oscillatory condition because it consists of even number of inverting elements in both cases. This is the reason why the loop does not oscillate by itself and settles to a steady state. However, when the edge of *ctrl* is applied to XORs, they invert their actual output level. Such action disturbs the steady state of the loop because the newly reached XORs' output levels begin to circulate through the loop. The logic level that was previously stable in the entire loop is switched to the opposite level in the half of loop. A pulse is raised as a result of this process, and it begins to run along the loop. The pulse will disappear after several runs (from tens to hundreds) of oscillations. The number of oscillations generated by TERO varies during each *ctrl* period. T-Flip-Flop (TFF) resolves if TERO made an odd or even number of oscillations during single *ctrl* period, and that represents one random bit ($tffout1$ and $tffout2$ signal, or just $tffout$ in next text). The purpose of the ANDs is to force the same initial conditions at the end of each *ctrl* period. The ANDs are controlled by the *rst* signal and their outputs are held in constant $'0'$ when *rst* $= '1'$ regardless the logic level on the other input. The $tffout$ is sampled at the falling edge of *rst*. After that $tffout$ is cleared by the *clr* signal. One can argue that there is no necessity of $tffout$ clearing because it does not affect volume of randomness, but despite this fact, it is cleared because of two reasons: first, it represents a Least Significant Bit (LSB) of an asynchronous counter used for measurements and the counter should start to increment from zero; and the second, more crucial reason is that slight bias will not be transformed to the correlation that enables the assumption of a stateless entropy concept[2].

TERO is a kind of bi-stable Flip-Flop (FF) with intentionally lengthened feedback paths. However, more common is to employ NAND gates or NOR gates instead of XORs for practical FFs. Employing NANDs or NORs in the proposed topology results in RS FF. Metastable behavior of RS FFs underwent rigorous analysis in [8] and [15]. Basically, when certain combination of signals appears at the inputs, RS FF can fall into the metastable state. After pico-to-nano-second time, this state is typically escaped by a trajectory, which may be temporarily oscillatory. Although TERO is not the RS FF regarding its functionality, it also escapes from the metastable state by the oscillatory trajectory due to lengthened feedback paths. This behavior is known as oscillatory metastable operation [8].

Practical evaluation of TERO by oscilloscope found that variation of the number of oscillations during each *ctrl* period is extensive in comparison to RO mode. Also observed is that TERO oscillates on the double frequency in comparison to RO mode. TERO operation was confirmed by qualitative SPICE analysis of TERO structure, as presented in the next subsection.

3.1 Transistor Level SPICE Simulation

The purpose of the SPICE simulation was to confirm qualitatively that behavior of an approximately balanced TERO structure exhibits oscillatory transient character as was reported for quite general semiconductor bistable structures [8]. SPICE simulation was not performed to examine the precise behavior in

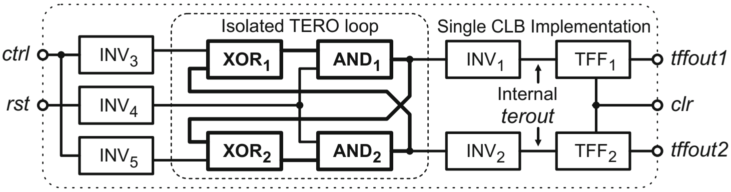

Fig. 1. Practical circuit of TERO used for the FPGA implementation. Entire TERO loop structure occupies just one CLB of Xilinx Spartan 3E. Usage of INV_1 – INV_5 enables such routing that signal directly connected to internal TERO loop will not be routed by off-CLB path. ANDs are used for forcing the same TERO initial conditions for each *ctrl* period by *rst* signal as well as for an additional delay needed for reliable oscillatory behavior of the circuit. TFFs are used for extraction of a random bits. TFFs are cleared at the end of each *ctrl* period by *clr* signal.

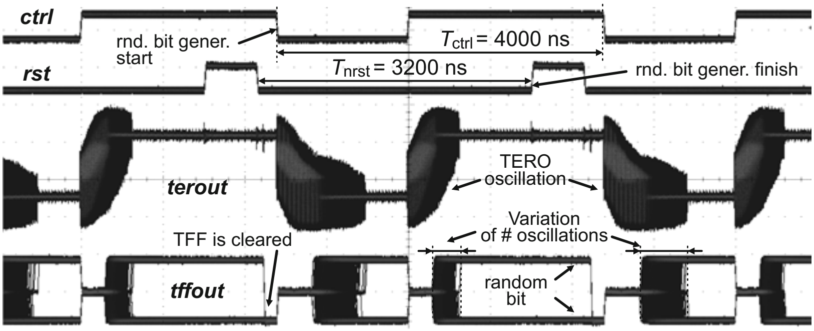

Fig. 2. The TERO operation oscilloscope screen-shot captured using infinite persistence mode and 20 MHz low-pass filter on *ctrl*, *rst*,and $tffout$ channels. The image was acquired by the Tektronix MSO 4104 oscilloscope. Each edge of the *ctrl* signal causes oscillation of TERO loop (*terout* signal). The number of oscillations observed varies during each *ctrl* period. TFF resolves if TERO made odd or even number of oscillation periods during one *ctrl* period that represents one random bit ($tffout$ signal). TERO is initialized to the same operating conditions at the end of each *ctrl* period by the *rst* signal. The $tffout$ is sampled on the falling edge of *rst*. Then $tffout$ is cleared.

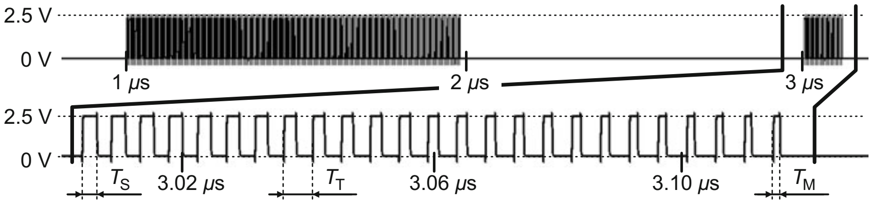

Fig. 3. LT Spice simulation of the TERO. TERO starts to oscillate at the rising (1 μs) or falling (3 μs) edge of the *ctrl* signal having a 4 μs period. Zoomed region shows that excited pulse disappears due to its shortening each loop crossing. T_{T} stands for the mean value of the oscillation period, T_{S} and T_{D} denote time durations of logic $'1'$ level when oscillation behavior raises and disappears respectively.

particular FPGA or CMOS manufacturing process but to get a qualitative results. Therefore the SPICE simulation was based on publicly available Linear Technology LT Spice IV tool [11] and 250 nm, 2.5 V CMOS transistors models [14] that were used also in [4]. Only the isolated TERO loop that consists of XOR$_1$ – AND$_1$ – XOR$_2$ – AND$_2$ (Fig. 1) was simulated. Tested XORs used standard 12 transistors layout and tested ANDs used standard 6 transistor layout. Wires were implemented as buffers of certain time delay and time constant.

LT Spice simulation (Fig. 3) confirms oscillatory behavior of the TERO in approximately balanced loop. Rising and falling edges of *ctrl* cause oscillations excitation in the loop (at 1 μs and 3 μs time respectively) that is in a good agreement with results of general bistable structures [8]. The zoomed region at the bottom of Fig. 3 shows that the excited pulse disappears due to its shortening in each loop passing. This phenomenon was also in a good agreement with [15].

Satisfactory results were obtained when the simulation was performed using maximal time step of 0.5 ps, otherwise numerical computation errors caused non-repeatability of that simulation. Simulation did confirm the double frequency of TERO in comparison to RO. LT Spice transient simulation does not take into account any contribution of transistor noises and therefore a simplified mathematical model is defined in the following section. This model allows for easier analysis of oscillation number variation due to noise. Features observed due to LT Spice simulation were used for the TERO basic model parameters derivation.

3.2 TERO Mathematical Model Based on Effects of Intrinsic Noise

LT Spice simulation provides a starting point for definition of a TERO mathematical model that takes intrinsic noise into account. Intrinsic noise, in which the samples of amplitude are assumed to be iid (independent and identically distributed) and follows a normal distribution $\mathcal{N}\left(0, \sigma^2\right)$, affects timing instability of signal edges passing through logic cells. We show that this noise has substantial impact on TERO performance. Graphical representation of the proposed mathematical model is given in the Fig. 4. Model works as follows: when a rising or falling edge of *ctrl* appears, TERO loop begins to oscillate (Fig. 2). The mean value of TERO oscillation period is equal to total delay of TERO loop T_{T}. An excited pulse of starting logic $'1'$ level time length T_{S} is shortened in each oscillation by T_{D} time due to slight intrinsic non-symmetry of the loop. Excited pulse will disappear when instant logic $'1'$ level time length reaches minimal possible value T_{M}. Asymmetry T_{D} is assumed to be affected by a period jitter $\Delta_{\mathrm{T}_{ij}}$, where i and j stands for i-th T_{T} period and j-th T_{ctrl} period respectively. The final number of oscillations executed for j-th T_{ctrl} is denoted as Y_{T_j}. The basic mathematical model of TERO mode is expressed as:

$$T_{\mathrm{S}} - T_{\mathrm{M}} = \sum_{i=1}^{Y_{\mathrm{T}_j}} \left(T_{\mathrm{D}} + \Delta_{\mathrm{T}_{ij}}\right) = T_{\mathrm{D}}\, Y_{\mathrm{T}_j} + \sum_{i=1}^{Y_{\mathrm{T}_j}} \Delta_{\mathrm{T}_{ij}} \ . \tag{1}$$

Both, T_{S} and T_{M} can be slightly affected by intrinsic noise and so considered as a contribution to final randomness. Value of the former can be affected by actual noise conditions when circuit is entering to oscillatory metastable state and value of the latter can be affected by actual noise conditions when the circuit does (or does not) allow to pass last pulse. However, according to our oscilloscope measurements the jitter accumulation over oscillatory trajectory exhibits the best entropy extraction and therefore this is the only focus in this paper. Investigation on T_{S} and T_{M} instability contribution to final randomness will be a subject of future research.

Similarly, it is possible to express the basic model of the same circuit in RO mode. In Fig. 2, T_{nrst} denotes a time period when RO is not reset and is oscillating at $T_{\mathrm{R}} = 2\,T_{\mathrm{T}}$ oscillation periods (Fig. 4). Each oscillation period is assumed to be affected by period jitter $\Delta_{\mathrm{R}_{ij}}$. Final number of oscillations executed for j-th T_{ctrl} is denoted as Y_{R_j}. Thus, the basic mathematical model of the RO mode circuit is expressed as:

$$T_{\mathrm{nrst}} = \sum_{i=1}^{Y_{\mathrm{R}_j}} \left(T_{\mathrm{R}} + \Delta_{\mathrm{R}_{ij}}\right) = 2\,T_{\mathrm{T}}\,Y_{\mathrm{R}_j} + \sum_{i=1}^{Y_{\mathrm{R}_j}} \Delta_{\mathrm{R}_{ij}} \quad . \tag{2}$$

Both, TERO mode (1) and RO mode (2) models were implemented in Matlab in order to evaluate them and to compare their sensitivity to intrinsic noise. The $\Delta_{\mathrm{T}_{ij}} = \Delta_{\mathrm{R}_{ij}} \approx \mathcal{N}\left(0, \sigma^2\right)$ simplification is assumed. The TERO mode model (1) was implemented using a pseudo-code Algorithm 1, where N stands for number of *ctrl* periods. The RO mode model (2) was implemented similarly.

Accordingly, Fig. 5 shows that Y_{T_j} is affected in a greater manner than Y_{R_j} when exposing the circuit in TERO mode and in RO mode to the same noise conditions. The ratio between their standard deviations is derived in following section.

3.3 Analytical Comparison of the TERO and RO Modes

Analytical derivation of a ratio between standard deviations of Y_{T_j} and Y_{R_j} is the objective of this section. Denote the standard deviation and mean value of Y_{T_j} as $\sigma_{Y_{\mathrm{T}}}$ and $\overline{Y_{\mathrm{T}}}$ respectively for the TERO mode. Similarly, denote the standard deviation and mean value of Y_{R_j} as $\sigma_{Y_{\mathrm{R}}}$ and $\overline{Y_{\mathrm{R}}}$ respectively for the RO mode. It is possible to show, that according (1) and (2) and under assumption $\Delta_{\mathrm{T}_{ij}} = \Delta_{\mathrm{R}_{ij}} \approx \mathcal{N}\left(0, \sigma^2\right)$, good approximations of $\sigma_{Y_{\mathrm{T}}}$, $\overline{Y_{\mathrm{T}}}$, $\sigma_{Y_{\mathrm{R}}}$, and $\overline{Y_{\mathrm{R}}}$ for simplified comparison of TERO and RO sensitivities are:

$$\overline{Y_{\mathrm{T}}} \approx \frac{T_{\mathrm{S}} - T_{\mathrm{M}}}{T_{\mathrm{D}}} \quad , \quad \sigma_{Y_{\mathrm{T}}} \approx \frac{\sigma}{T_{\mathrm{D}}} \sqrt{\frac{T_{\mathrm{S}} - T_{\mathrm{M}}}{T_{\mathrm{D}}}} = \frac{\sigma}{T_{\mathrm{D}}} \sqrt{\overline{Y_{\mathrm{T}}}} \ , \tag{3}$$

$$\overline{Y_{\mathrm{R}}} \approx \frac{T_{\mathrm{nrst}}}{2\,T_{\mathrm{T}}} \ , \quad \sigma_{Y_{\mathrm{R}}} \approx \frac{\sigma}{2\,T_{\mathrm{T}}} \sqrt{\frac{T_{\mathrm{nrst}}}{2\,T_{\mathrm{T}}}} = \frac{\sigma}{2\,T_{\mathrm{T}}} \sqrt{\overline{Y_{\mathrm{R}}}} \ . \tag{4}$$

The ratio $\frac{\sigma_{Y_{\mathrm{T}}}}{\sigma_{Y_{\mathrm{R}}}}$ is derived from combining (3) and (4):

$$\frac{\sigma_{Y_{\mathrm{T}}}}{\sigma_{Y_{\mathrm{R}}}} \approx \frac{2\,T_{\mathrm{T}}}{T_{\mathrm{D}}} \sqrt{\frac{2\,T_{\mathrm{T}}\ (T_{\mathrm{S}} - T_{\mathrm{M}})}{T_{\mathrm{D}}\,T_{\mathrm{nrst}}}} \ . \tag{5}$$

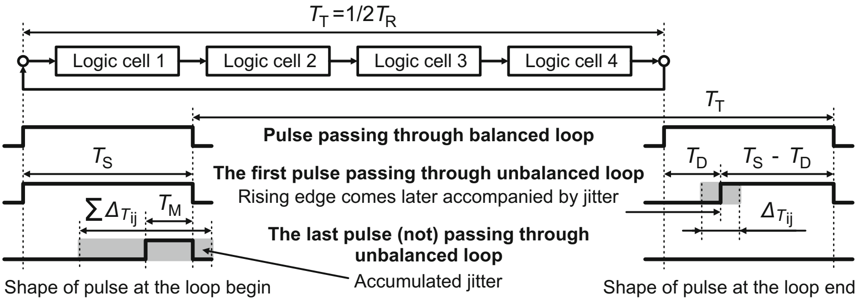
$T_T = 1/2T_R$
Logic cell 1
Logic cell 2
Logic cell 3
Logic cell 4
T_T
Pulse passing through balanced loop
T_S
The first pulse passing through unbalanced loop
Rising edge comes later accompanied by jitter
T_D
$T_S - T_D$
$\sum \Delta T_{ij}$
T_M
ΔT_{ij}
The last pulse (not) passing through unbalanced loop
Accumulated jitter
Shape of pulse at the loop begin
Shape of pulse at the loop end

Fig. 4. Graphical representation of the mathematical model, where the instability of the pulse shortening during circulation around TERO is a key issue. Therefore the pulse will disappear after several oscillations.

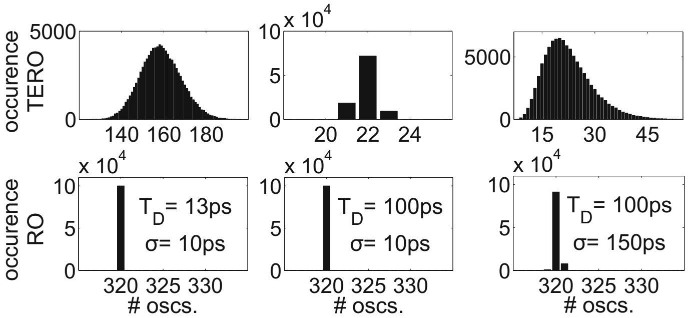

Fig. 5. Simulation of TERO (1) and RO (2) basic models performed in Matlab. Both models share the same parameters in order to compare their randomness extraction performance. Histograms show the occurrence of the recorded number of oscillations under three sets of different operating conditions, which vary in σ and T_D, where instability of T_D follows $\mathcal{N}\left(0, \sigma^2\right)$. Other parameters remain constant over all simulations: $T_T = 5\,\text{ns}$, $T_{\text{nrst}} = 3200\,\text{ns}$, $T_S - T_M = 0.4\,T_T$, $N = 10^5$.

Algorithm 1. TERO mathematical model simulation

```
Require: $T_S$, $T_M$, $T_D$, $T_T$, $\sigma$, $N$
Ensure: $Y_{T_1}, Y_{T_2} \dots Y_{T_j} \dots Y_{T_N}$
  for $j = 1$ to $N$ do
    $Y_{T_j} \Leftarrow 0$
    $acc \Leftarrow 0$
    while $acc < T_S - T_M$ do
      $acc \Leftarrow acc + T_D + \mathcal{N}\left(0, \sigma^2\right)$
      $Y_{T_j} \Leftarrow Y_{T_j} + 1$
    end while
  end for
```

When applying practical values acquired from both hardware and LT Spice simulation: $T_{\mathrm{nrst}} = 3200\,\mathrm{ns}$, $T_{\mathrm{T}} = 5\,\mathrm{ns}$, $T_{\mathrm{D}} = 0.013\,\mathrm{ns}$, $T_{\mathrm{S}} - T_{\mathrm{M}} = 0.4\,T_{\mathrm{T}}$ to (5), then $\frac{\sigma_{Y_{\mathrm{T}}}}{\sigma_{Y_{\mathrm{R}}}} \doteq 533$.

In other words, the proposed circuit is hundreds of times more sensitive in TERO mode to the period jitter than the same circuit in RO mode. At this point one can argue that this feature will increase vulnerability to external interferences or attacks. As it will be shown in more complex simulation that follows in next subsection, TERO thanks to its differential structure can decrease influence to the global (outside of CLB) perturbations while still maintaining high sensitivity to local (inside CLB) intrinsic noises.

3.4 TERO and RO Response under External Perturbations

The response of TERO to deterministic perturbations patterns was simulated using Macro-Model (MM) that in contrary to two previous subsections takes into account the same TERO loop structure as was implemented in real FPGA hardware (Fig. 1). MM was written in VHDL and simulated using ModelSim. The MM simulation setup including the simulated loop is shown in Fig. 6. The logic function of each component is computed instantly with an addition of a synthetic delay. The delay is implemented as a simple state machine which allows independent control of the delay of both the rising edge and the falling edge.

The noise pattern samples for each edge of each component are stored in separated file which was generated by Matlab. The simulated patterns are composed of deterministic perturbation and intrinsic noise. Deterministic perturbation represents global influence of power supply variations or electro-magnetic interference that can affect time delays of logic elements. It is assumed that deterministic perturbation affects same logic elements by the same manner. On the other hand intrinsic noise is assumed to be independent for each logic and follows the normal distribution $\mathcal{N}\left(0, \sigma^2\right)$.

Simulation results for various compositions of deterministic perturbation and intrinsic noise for both TERO and RO mode are shown in Fig. 7. Frequency of noise was assumed to be higher than frequency of TERO oscillations. It is possible to get qualitatively similar results when the noise is band limited as well. Results of the MM simulation confirm that TERO randomness extraction performance is superior to RO performance. Accordingly, results show in Fig. 8 that both the basic mathematical model simulation and the VHDL MM simulation are in good agreement with the results acquired from the FPGA hardware.

4 Hardware Implementation

Xilinx Spartan 3E Starter Board was used as an evaluation platform [25]. TERO shown in Fig. 1 was placed into one CLB of Xilinx Spartan 3E because of simpler proper routing, using only local, not global paths. Even though there are 9 logic functions and only 8 LUTs in single CLB, the TERO fits inside due to

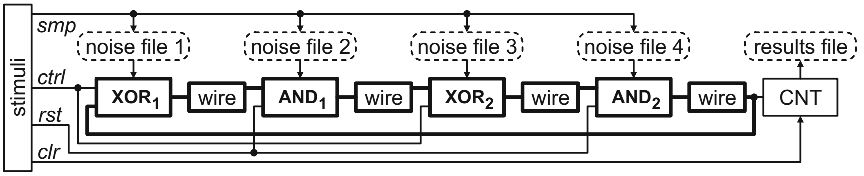

Fig. 6. The TERO macro-model implemented in VHDL. Constant delay of rising and falling edge can be set independently for each element, including wires. Non-symmetry was achieved by different rising edge delay and falling edge delay in the XOR$_1$ element. Signals *ctrl*, *rst* and *clr* have the same purpose as was described in Sect. 3 above. Signal *smp* controls noise sampling from the noise files 1–4. Noise files were generated by Matlab and contains data for both rising edge delay and falling edge delay instability. Final number of oscillations for each *ctrl* period is recorded in the results file.

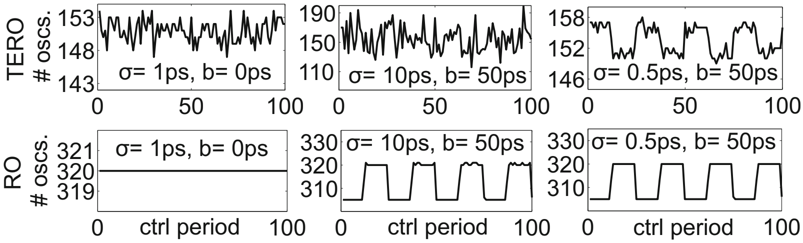

Fig. 7. VHDL macro-model simulation using ModelSim shows number of oscillations in 100 consecutive T_{ctrl} periods for both TERO (up) and RO (down) mode. Three different compositions of noise patterns (each column different) were used; where intrinsic noise follows $\mathcal{N}\left(0,\sigma^2\right)$ and b stands for amplitude of global deterministic perturbation of a square shape. It is obvious, that TERO is able to extract intrinsic noise of $\sigma = 0.5\,\mathrm{ps}$ while RO is barely able to extract intrinsic noise of $\sigma = 10\,\mathrm{ps}$ when both were exposed to the same operating conditions. Moreover, TERO is less sensitive to global deterministic perturbation than RO.

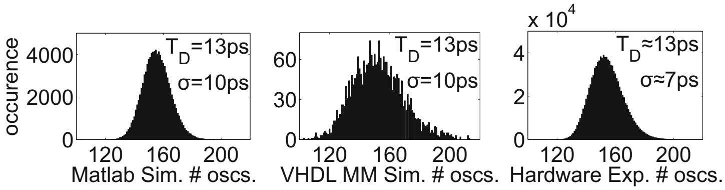

Fig. 8. Comparison of simulations and hardware experiment results. Histograms shows occurrences of TERO oscillations number for direct simulation of mathematical model using Matlab (left), VHDL macro-model simulation using ModelSim (midle) and hardware experiment (right). Intrinsic noise follows $\mathcal{N}\left(0,\sigma^2\right)$ in the simulations. Rest parameters were: $T_{\mathrm{T}} = 5\,\mathrm{ns}$, $T_{\mathrm{S}} - T_{\mathrm{M}} = 0.4\,T_{\mathrm{T}}$. $N = 10^5$, $N = 2600$, and $N = 10^6$ for the Matlab simulation, MM simulation, and the experiment respectively. Parameters T_D and σ of the hardware experiment are estimated according to VHDL MM simulation. The histogram of VHDL MM simulation is wider than the histogram of mathematical model simulation due to presence of the noise source $\mathcal{N}\left(0,\sigma^2\right)$ in each logic element.

using hardwired XORs in carry chain logic. There is indication, that this fast XOR causes more stable TERO performance. Consequently, a good constellation of placement and routing is locked by the user constrain file. Locking the circuit makes it portable through all CLBs with acceptable dispersion of the TERO parameters. The example of proper place and route of two TEROs in neighboring CLBs is shown in Fig. 9 b.

The entire system used for evaluating the TERO performance is shown in Fig. 9 a. There are two TEROs in order to investigate their cross-correlation. The number of oscillations are counted by asynchronous counters which are faster than synchronous counters. A typical frequency of TERO in four element loop structure is 200 MHz approximately. Asynchronous counters are implemented by the chain of TFFs. A control state machine ensures communication via USB that is used for transferring counter values to the computer for further analysis. The benefit of this structure is that expensive oscilloscopes are not necessary – just investigating of counter values is fairly enough for detailed analysis.

5 Experimental Results

Mean values of asynchronous counters, bias of extracted LSBs and autocorelations of generated bit streams were used for fast evaluation of of TERO (RO) performance (dependency) in closely placed CLBs ("Next" configuration) as well as far-away placed CLBs ("Diag." configuration) in the target Xilinx FPGA. Mixing of two CLB outputs was performed by XOR operation that performs standard decimation by a factor of 2. Results shown in Fig. 10 and Table 1 were evaluated for 1 Mbit sequences acquired from the evaluation platform. More complex NIST [16] tests were performed for a 250 Mbit sequence in order to detetct potentially more complex deviations from the ideal one.

Autocorrelation test evaluates correlations between the sequence of extracted random bits and shifted (by number of *ctrl* periods) versions of it. Random bits are extracted as LSBs of number of oscillations of TERO (RO) at each period of *ctrl*. The statistic used in Fig. 10 for autocorrelations is normalized according to (5.5) on p.182 in [13] which approximately follows $\mathcal{N}(0, 1)$ distribution for ideal random source. According to the 3σ rule, any value outside of the $< -3, 3 >$ interval indicates a probable deviation from ideal source of randomness.

From presented experimental results we can state that TERO can produce uncorrelated (or with high probability of independent) sequences which is not the case of RO composed of the same elements as TERO. Moreover, the XOR combination of two channels of TERO greatly improves statistical properties of generated sequences. This was confirmed by improving the mean value and especially passing of the NIST 800-22 or FIPS 140 statistical tests suites which indirectly indicate bit sequence independence. When evaluating XOR combination of two channels of RO the situation is worse as a consequence of the dependence of the examined bit sequences.

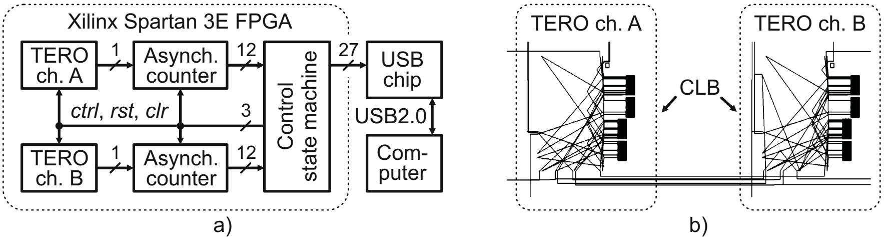

Fig. 9. (a) – Evaluation platform setup. (b) – Example of proper place and route of two TERO channels in two most closest CLBs; both TEROs are routed in the same way.

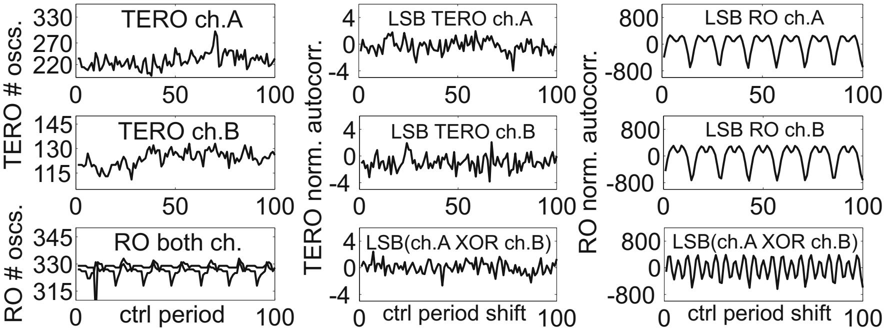

Fig. 10. TERO and RO mode comparison in hardware. Channels A and B are synthesized in the closest CLBs. Left column shows number of oscillations for both channels of TERO and RO in 100 consecutive *ctrl* periods. Normalized autocorrelation of TERO channels A and B and their XOR combination for 1–100 *ctrl* periods shift is given in the middle column. The same results, but for RO mode is depicted in right column.

Table 1. Results of the TERO evaluation in Xilinx Spartan 3E FPGA. Position "Next" means TERO (RO) A and B are placed in the closest CLBs, while "Diag." means A and B are placed in CLBs that are in opposite corners of the FPGA fabric.

TEST	Source	Next(TERO)	Diag.(TERO)	Next(RO)	Diag.(RO)
Mean	LSB A / LSB B	0.51/0.48	0.51/0.48	0.47/0.44	0.55/0.46
Value	LSB(A XOR B)	0.5002	0.4999	0.4539	0.7926
Normalized cross-correlation (for shift=0)	LSB (A,B)	0.4160	-0.0917	-94.3378	599.3945
NIST / FIPS	Only LSB A	F / P	F / F	- / F	- / F
Statistical	Only LSB B	F / F	F / F	- / F	- / F
tests result	LSB(A XOR B)	P / P	P / P	- / F	- / F

6 Conclusion and Future Work

A new high entropy element for FPGA-based TRNGs was introduced. This element was denoted as TERO and reasonably satisfies design goals formulated in Section 2. Its greatest advantage is high sensitivity to random processes inside FPGA logic cells, while rejecting global perturbation.

Moreover, TERO is straightforwardly inner testable. Instant evaluation of the number of oscillations and consequent estimation of noise parameters from them according to (3) allow instant detection of malfunctions when the random bit is generated. The implemented testing system can decide whether the bit will be used as a member of the resulting random sequence accordingly.

Furthermore, TERO has a clear basic mathematical model that was confirmed by LT Spice simulation, VHDL MM simulation and hardware evaluation. The TERO has hundreds times greater sensitivity to random processes inside logic gates then RO build up from the same components according to our proposed model. VHDL MM simulation shows that TERO can reject global perturbation better than RO due to its differential structure. In particular TERO can extract noise of $\sigma = 0.5\,\mathrm{ps}$ standard deviation in the presence of 50 ps global perturbation.

An experimental TRNG which uses an XOR combination of two TEROs was introduced. The source of randomness occupies just two CLBs and produce random sequence at 250 kbps bit-rate of such quality that it can pass NIST 800-22 statistical tests without any need for further complex post-processor. This shows great potential of TERO for the FPGA based TRNGs designs.

Experiments showed that proper place and route strategies are essential for TERO and therefore further research will be focused on reliable lock of place and route synthesis constrains, analysis of TERO features in different FPGA internal positions, and evaluation of TERO operation in FPGAs of other vendors.

Although our entire investigation was carried out using one Spartan 3E board for the purpose of the paper, our latest experiments were processed using Actel Fusion FPGAs due to two reasons; first, to investigate TERO using different FPGA technology, and second, the availability of a greater number (ten) of equal Actel boards. Actel does not provide any tool for custom routing as Xilinx does so the routing is black-box in Actel. Nevertheless, preliminary results show that the variance of TERO oscillations in each tested Actel board at each tested position was satisfactory and also under nonstandard operation conditions through wide temperature and power supply range. The final random bit sequence composed of 16 TERO channels XOR combination passes the NIST 800-22 tests for every described setup. However, the quantity of results acquired from the Actel platform are excessive and will be a subject of an upcoming paper.

Future work will also include synthesis of a testing system that can estimate the statistical parameters of noise from the variance of oscillation number for the purpose of malfunction detection. This testing system will be incorporated in the final, ready-to-use TRNG.

Acknowledgement. This work has been done in the frame of the Slovak scientific project VEGA 1/0045/10 2010-2011 of Slovak Ministry of Education. Authors would like thank to Actel University Program for a donation of 10 Actel Fusion FPGA evaluation boards, which enabled us to confirm the TERO principle using this large number of FPGA boards.

References

1. Bochard, N., Bernard, F., Fischer, V.: Observing the randomness in RO-based TRNG. In: International Conference on Reconfigurable Computing and FPGAs, Cancun, Quintana Roo, Mexico, December 9-11, pp. 237–242. IEEE Computer Society, Los Alamitos (2009)
2. Bucci, M., Giancane, L., Luzzi, R., Varanonuovo, M., Trifilett, A.: A Novel Concept for Stateless Random Bit Generators in Cryptographic Applications. In: 2006 IEEE International Symposium of Circuits and Systems - ISCAS 2006, Island of Kos, Greece, May 21-24, pp. 317–320. IEEE Computer Society, Los Alamitos (2006)
3. Dichtl, M., Golić, J.: High-Speed True Number Generation with Logic Gates Only. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 45–62. Springer, Heidelberg (2007)
4. Dichtl, M., Meyer, B., Seuschek, H.: SPICE Simulation of a "Provably Secure" True Random Number Generator (2008), http://eprint.iacr.org/2008/403.pdf
5. Epstein, M., Hars, L., Krasinski, R., Rosner, M., Zheng, H.: Design and Implementation of a True Random Number Generator Based on Digital Circuit Artifacts. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 152–165. Springer, Heidelberg (2003)
6. Fischer, V., Drutarovsky, M.: True Random Number Generator Embedded in Reconfigurable Hardware. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 415–430. Springer, Heidelberg (2003)
7. Golic, J.: New Methods for Digital Generation and Postprocessing of Random Data. IEEE Transactions on Computers 55(10), 1217–1229 (2006)
8. Kacprzak, T.: Analysis of Oscillatory Metastable Operation of an R-S Flip-Flop. IEEE Journal of Solid-State Circuits 23(1), 260–266 (1988)
9. Koç, Ç.K. (ed.): Cryptographic Engineering. Springer, Heidelberg (2009)
10. Killmann, W., Schindler, W.: Functionality classes and evaluation methodology for true (physical) random number generators, Version 3.1 (September 2001), http://www.bsi.bund.de/zertifiz/zert/interpr/trngk31e.pdf
11. Linear Technology: LT Spice IV, http://www.linear.com/designtools/software/
12. Markettos, A.T., Moore, S.W.: The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 317–331. Springer, Heidelberg (2009)
13. Menezes, J., Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, New York (1997), http://www.cacr.math.uwaterloo.ca/hac/
14. Rabaey, J.M., Chandrakasan, A., Nilolic, B.: Digital Integrated Circuits, 2nd edn. Prentice-Hall, Englewood Cliffs (February 2010), http://bwrc.eecs.berkeley.edu/IcBook
15. Reyneri, L., Corso, D., Sacco, B.: Oscillatory Metastability in Homogenous and Inhomogeneous Flip-Flops. IEEE Journal of Solid-State Circuits 25(1), 254–264 (1990)

16. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications, NIST Special Publication 800-22 rev1a (April 2010), `http://csrc.nist.gov/groups/ST/toolkit/rng/`
17. Schellekens, D., Preneel, B., Verbauwhede, I.: FPGA Vendor Agnostic True Random Number Generator. In: Proceedings of the 16th International Conference on Field Programmable Logic and Applications (FPL), Madrid, Spain, August 28-30, pp. 1–6. IEEE Computer Society, Los Alamitos (2006)
18. Schindler, W., Killmann, W.: Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 431–449. Springer, Heidelberg (2003)
19. Sunar, B.: Response to Dichtl's Criticism (March 2008), `http://ece.wpi.edu/~sunar/preprints/comment.pdf`
20. Sunar, B., Martin, W.J., Stinson, D.R.: A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks. IEEE Transactions on Computers 56(1), 109–119 (2007)
21. Tokunaga, C., Blaauw, D., Mudge, T.: True Random Number Generator With a Metastability-Based Quality Control. IEEE Journal of Solid-State Circuits, 78–85 (January 2008)
22. Vasyltsov, I., Hambardzumyan, E., Kim, Y.S., Karpinskyy, B.: Fast Digital TRNG Based on Metastable Ring Oscillator. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 164–180. Springer, Heidelberg (2008)
23. Wold, K., Tan, C.H.: Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings. International Journal of Reconfigurable Computing 2009, 1–8 (June 2009), `http://www.hindawi.com/journals/ijrc/2009/501672.html`
24. Wollinger, T., Guajardo, J., Paar, C.: Security on FPGAs: State-of-the-art implementations and attacks. ACM Transactions on Embedded Computing Systems (TECS), 534–574 (2004)
25. Xilinx: Spartan-3E Starter Kit, `http://www.xilinx.com/products/devkits/HW-SPAR3E-SK-US-G.htm`