Fast Digital TRNG Based on Metastable Ring Oscillator

Ihor Vasyltsov, Eduard Hambardzumyan,
Young-Sik Kim, and Bohdan Karpinskyy

Samsung Electronics, SoC R&D Center, System LSI, Korea
ihor.vasiltsov@samsung.com

Abstract. In this paper, a new true random number generator (TRNG), based entirely on digital components is proposed. The design has been implemented using a fast random number generation method, which is dependent on a new type of ring oscillator with the ability to be set in metastable mode. Earlier methods of random number generation involved employment of jitter, whereas the proposed method leverages the metastability phenomenon in digital circuits and applies it to a ring oscillator. The new entropy employment method allows an increase in the TRNG throughput by significantly reducing the required entropy accumulating time. Samples obtained from simulation of TRNG design have been evaluated using AIS.31 and FIPS 140-1/2 statistical tests. The results of these tests have proven the high quality of generated data. Corners analysis of the TRNG design was also performed to estimate the robustness to technology process and environment variations. Investigated in FPGA technology, phase distribution highlighted the advantages of the proposed method over traditional architectures.

Keyword: Digital TRNG, Metastable Ring Oscillator, AIS.31, FPGA.

1 Introduction

The security of most cryptographic systems relies on unpredictability and irreproducibility of digital key-streams that are used for encryption and/or signing of confidential information. These key-streams are generated by random number generators (RNG), which are further split into two classes: true random number generators (TRNG) and deterministic random number generators (DRNG) [1], [2]. The key difference between TRNG and DRNG lies in the entropy source component. For TRNG, an analog physical process (electronic thermal noise, radioactive decay, etc.) is used, while for DRNG, a random number called seed is used [1], [2]. Since the seed value is constant, it must be refreshed regularly to maintain the required security level. This seed value is generated by a TRNG, so any security system should be comprised of a TRNG as the key part. Compromising on the TRNG means compromising on the whole security system. That's why a great degree of attention is paid to TRNG as the fundamental security component that guarantees the quality of the whole security system.

E. Oswald and P. Rohatgi (Eds.): CHES 2008, LNCS 5154, pp. 164–180, 2008.

In this paper, we introduced a TRNG based entirely on digital designs. For this purpose, a new type of ring oscillator was created. To validate the theoretical background of the proposed method, we implemented and simulated it in the Cadence Design Environment (CDE). Additionally, we performed the FPGA implementation for phase distribution investigation. The samples obtained were statistically evaluated according to AIS.31 and FIPS 140-1/2 standards [3], [4].

This paper contains the following sections: Section 2 describes the basic concept of digital TRNG and technology state of the art. Section 3 describes metastable ring oscillator theory, implementation and simulation, statistical evaluation, and robustness investigation. Section 4 describes the investigations in FPGA implementation and finally, Section 5 gives the conclusion of this paper.

2 Digital TRNG

Traditional TRNGs are based on a precise analog design requiring special custom layout. The migration of such TRNG products to a new platform or technology is complicated since it involves a heavy custom re-design, an increased budget, and more time-to-market. TRNG design, which is based entirely on digital components, is free from such drawbacks. By significantly reducing the need to custom re-design, it facilitates product migration. Hereafter, we will use the term *Digital TRNG* in this paper to explain this totally digital synthesizable design.

The first scheme considered as totally Digital TRNG was based on coupled oscillators. This method produces randomness from the phase noise in free-running oscillators. The output of the fast oscillator is sampled on the rising edge of a slower clock using a D flip-flop [5]. The main physical phenomenon used as an entropy source in such architectures is jitter, which is defined as the short-term variation of signal's significant instants from their ideal positions in time, due to the existence of thermal and shot noise in a semiconductor device. Oscillator jitter causes uncertainty in the exact sample values, ideally producing a random bit for each sample. By carefully selecting the ratio between the two oscillator frequencies, an artificially enhanced randomness can be achieved. But such synchronization of oscillators requires special custom design that increases the complexity of development. So, straightforward implementation of such a scheme cannot be achieved easily.

Another problem with such a scheme is that it necessitates wait for jitter accumulation and only after that accumulated entropy can be sampled as random data. The length of waiting time depends on the technology specification and component parameters, and usually takes from a few hundreds to several thousands of oscillator periods, limiting the throughput up to 1 Mbits/sec, which is considered critical for high-performance security applications.

There were many efforts to decrease the jitter accumulation time. For example, Jun and Kocher employed the hybrid TRNG [6], wherein the thermal noise source modulated the frequency of the slower clock. The variable, noise-modulated slower clock triggers the measurements of the fast clock. Drift between the two clocks thus provide the source of random binary digits. But such architecture cannot

be considered as purely digital because direct noise amplification circuit requires analog design. Another example of the mixed usage of digital and analog TRNGs is presented by Trichina, Bucci, Seta and Luzzi [7].

To overcome the de-synchronization of the sampling oscillator, another approach was used in [8], where Sunar, Martin and Stinson proposed to use a plurality of free running ring oscillators (RO), outputs of which are XORed. According to the authors, properly selected numbers of oscillators and their periods guarantee that the entire spectrum will be populated with transition zones. Also, sampling the waveform only in such zones would provide enough entropy. The area cost for this solution is huge. For example, in [9] even a minimal TRNG design based on 110 free running 3-cascades ring oscillators occupies 565 slices in Xilinx Virtex FPGA, what is more than the lightest known AES implementations [10]. Additionally, in [11] there were serious concerns about the unrealistic assumptions of the theoretical model used in [8] which raised questions about the practical implementation of such a Digital TRNG architecture. Bock, Bucci and Luzzi proposed a scheme where the oscillators are re-synchronized before each bit generation [12]. As a result, the periodical behavior typical for the oscillator-based source is suppressed and each bit generation restarts from the same state as with a direct-amplification source. Fischer and Drutarovsky proposed to sample the jittered signal by several shifted in-time flip-flops, aiming to guarantee that at least one of them will correspond to the random jitter [13]. However, obtained throughput was low. For the implementation in Altera APEX EP20K200 FPGA with a 88.245 MHz internal clock, it generated only 69 kbps.

Another type of Digital TRNG exploits the metastability of RS latches and edge-triggered flip-flops (for example, see [14]). The output of such a flip-flop may become unpredictable if the input and clock signals are such that the setup and/or hold times are violated. For example, when the data input signal is forced to change at nearly the same time as the clock signal the output signal then stabilizes on a random, typically biased value after a random amount of time. The metastability of D-type flip-flops can be exploited together with the jitter of underlying ring oscillator signals by using D-type flip-flops for sampling the ring oscillator signals. In any case, naturally occurred metastability events are relatively rare and when they occur are sensitive to temperature and voltage changes [14]. So, TRNGs, which are based solely on naturally occurred metastability events are relatively slow and do not appear to be very reliable.

Tkacik proposed the use of two oscillators of different sizes that were clocking linear feedback shift register and cellular automata shift register [15]. The investigation of individual statistical characteristics of LFSR and CASR outputs showed the presence of some weakness. To improve the design their outputs were XORed. Such architecture includes a pseudo randomness properties and does not comply with the AIS.31 P2.d)(vii) requirements for getting desirable statistical raw data characteristics [1], [2]. A theoretical attack for this TRNG is described by Dichtl in [16].

Golić introduced Fibonacci and Galois ring oscillators, which are both defined as generalizations of a typical ring oscillators [17]. He claimed that the high-speed

output oscillating signal has both pseudo and true randomness properties. True randomness accumulates from unpredictable variations in the delay of internal logic gates that get propagated and enhanced through feedback, possibly in a chaotic manner, and also from internal metastability events. It is suggested that further randomness due to metastability may be induced within a sampling unit (e.g., a D-type flip-flop) as well as that the mutual coupling effect between the oscillating and sampling signals may be significantly reduced by the pseudo random noise-like form of the oscillating signal. Recently, the inherited pseudo randomness property of Fibonacci and Galois ring oscillators was fixed by using restarting mode, which makes the generator stateless and excludes pseudo randomness as described in [11].

In spite of the many proposals for hardware-based TRNGs, finding an efficient and robust method for high-speed generation of true random numbers that can be implemented by using only logic gates in digital semiconductor technology remains a challenge. The ideal method should be efficient in terms of gate count, achievable speed, and power consumption. Further in this paper, the authors propose an original method which can be used for Digital TRNG implementation.

3 Metastable Ring Oscillator

3.1 Metastability Employment

To increase the throughput of the Digital TRNG based on jitter phenomena in ring oscillators, the available solutions require either a custom layout design or huge area costs. In this paper, we suggest the use of another physical phenomenon as entropy source in oscillators – metastability.

It is known that for any digital component with threshold level near the metastable state, the circuit behavior becomes totally stochastic and depends on the characteristics of the circuit noise [18]. Thus, a metastable state is the perfect entropy source. But, due to the mismatch of transistors, temperature imbalance within a chip, ionizing radiation, or any other parasitic fluctuation of the output voltages, the probability that the physical flip-flop circuit will stay in the metastable region is very small [19]. Therefore, straightforward employment of metastability phenomena in flip-flop circuits is inefficient due to the rare occurrence of natural metastability event [14] .

Thus, it is required to build a circuit with the ability to be put into a metastable state. Our investigation in CMOS technology showed that such a circuit could be implemented on an inverter. In Fig.1, the generic scheme of metastability employment based on a CMOS inverter is shown. If the inverter is connected into the loop by a switch, the output voltage converges to metastability level and stays there as long as required (see Fig.1b))[1]. Due to inherited thermal noise, the output voltage stochastically fluctuates around the metastable level.

[1] This state is stable as long as input and output are connected, and becomes metastable when the control signal allows the oscillator to run.

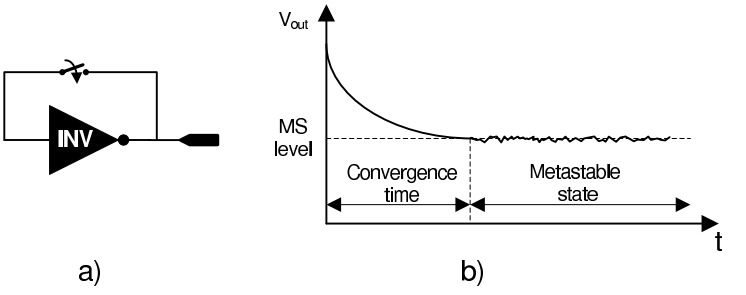

Fig. 1. Metastability employment scheme based on CMOS inverter a), and its convergence process b)

When a ring oscillator is composed of such schemes, after disconnecting the feedback loop, the initial state of the ring oscillator is completely defined by the entropy from stochastic fluctuations of each inverter (here we neglected the deterministic disturbances propagated through the power supply; such a special case was considered separately and showed that our design is robust for realistic ±10% voltage variation). In Fig.2, the explanation of metastability employment in an inverter-based ring oscillator circuit is shown.

1. *Initialization.* The initialization is done by putting the RO system into the metastable point (threshold voltage level). The momentary voltage value of

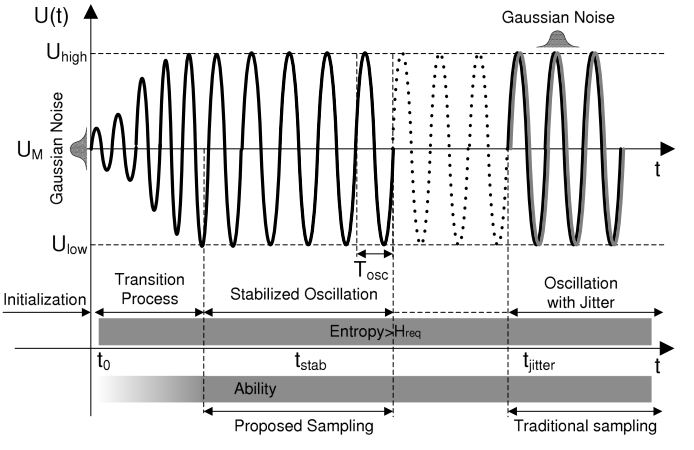

Fig. 2. Metastability employment in the inverter-based ring oscillator. Entropy exists at the beginning of the oscillation and transition periods, because initial voltage is defined by thermal noise. Because of low amplitude value and not stabilized period, the sampling is postponed until amplitude value is high enough and setup/hold time condition is satisfied. Usually it takes only few periods, so appeared latency is negligible comparatively to jitter accumulation process.

the initial noise influences the RO system and causes the oscillations, which at the beginning are very low by amplitude (and can be recovered by following momentary voltage values with bigger amplitude). Thus, the initial voltage value of an RO system is defined by the noise and already inherits enough entropy.
2. *Transition Process.* This process is semi-deterministic (almost does not increase entropy). Deterministic part consists of amplifying the noise signal obtained at the initialization mode. But due to the continuous influence of noise, this deterministic signal can be recovered and the initial entropy level can even be increased. Sampling in this period is not applied because the signal voltage value could be significantly lower than required[2].
3. *Stabilized Oscillations.* Full-range amplitude oscillations at stabilized periods allow for effective sampling, because of the inherited entropy from the initialization mode.

As can be seen from Fig.2, the main advantage of the proposed method is the significant decrease in the latency of TRNG due to earlier sampling times. Compare: with jitter accumulating it is required to wait a few hundreds/thousands of RO oscillation periods and for the method proposed in this paper it is enough to wait only few periods.

3.2 Generic Meta-RO Architecture

Based on the theoretical assumptions from the previous section, we propose an original architecture of a metastable ring oscillator (Meta-RO) as shown in the Fig.3. This architecture consists of:

- an odd plurality of inverters that can form either independent entropy source components while in metastable mode, or a traditional RO while in generation mode;
- a corresponding number of Switching Components (referred as MUXes) for re/dis-connecting inverters between two modes;
- a Control Clock Generator to control the random number generation process by switching between metastability (MS) and generation (Gener.) mode to guarantee the proper entropy collecting and entropy acquiring;
- a Sampling Component (referred as D flip-flop) for sampling the collected entropy from Meta-RO;
- a Delay Component to synchronize the sampling process with generating random data process by pre-defined delay.

The proposed method operates as follows (see Fig.3). First, the Control Clock Generator switches the system into MS mode by sending the corresponding signals to the Switching Components to disconnect each inverter from the others and connect it into a loop (this helps to apply the metastability point to the input of every inverter after a while). Since each inverter is disconnected from the other and the threshold point voltage is applied to its input, they form a set of *independent noise sources*.

[2] The gain of inverters of the modern technology is big enough, so usually transition process is very short.

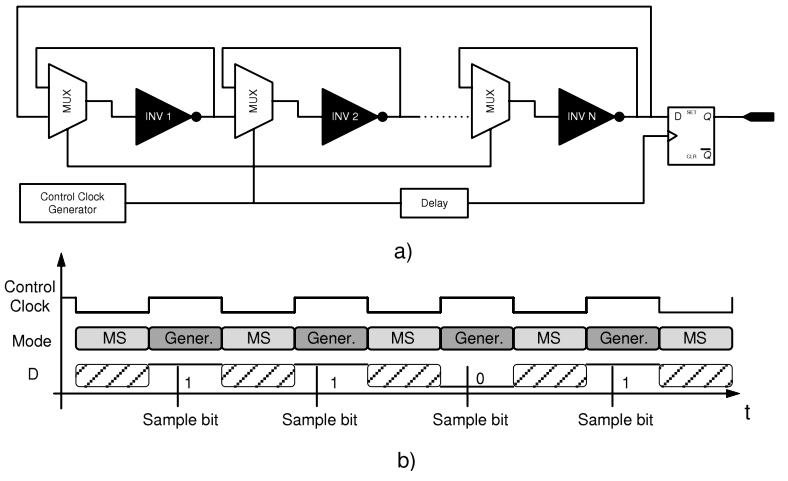

Fig. 3. Generic Meta-RO architecture a) and operational diagram b). The set of inverters could be used to form independent entropy sources (in metastable mode) or a regular ring oscillator to amplify and resolve the obtained random state.

After a while, the system is switched into the Generation mode, where inverters are re-connected to each other to form a traditional RO. Since in the previous MS mode the value of each inverter output was defined by random noise, the momentary voltages inside the RO are also random, causing high entropy. After sampling a random bit, the TRNG system again is switched to MS mode to collect a new random value. Since for whole process it is required to wait just several periods of RO oscillation, the total TRNG throughput can be increased significantly compared to traditional jitter employment architectures.

3.3 Implementation in Cadence Design Environment

For appropriate and accurate investigation of the proposed architecture, Meta-RO5st (a 5-stage metastable ring oscillator) has been implemented in Cadence Virtuoso Environment version 5.10.41 within a 65nm technology process library.

The specifics of our investigation are such that even if we are investigating a Digital TRNG case to consistently prove the proposed Meta-RO architecture, we still have to provide analog simulation with transient analysis of random data generation. In this case, the realistic implementation of the proposed method into existent ASIC technology will verified[3].

The whole design of the core of the Meta-RO5st architecture (FIFO, external control and interfaces not included) covered up to 70 transistors. Taking into

[3] The relevancy of the simulation to the real chip processes still is an open question. In this paper the authors could not solve it completely, but at least consider the technology process and temperature variations. Another advantage of the simulation consists in the absence of complex patterns in the power supply lines, which complicates the distinguishing between true and pseudo randomness.

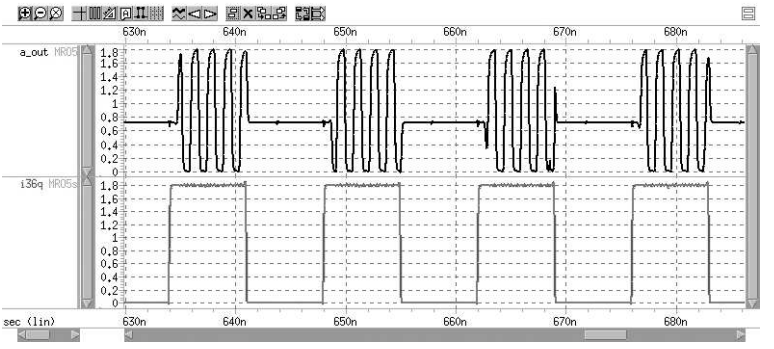

Fig. 4. Results of Meta-RO5st simulation. From the figure it is clear the difference between MS and Generation mode following the control clock signal.

account the nominal parameters of CMOS transistor in 65nm technology, the raw estimation for the covered area is about $1\mu m^2$, which is the smallest area estimation for the known Digital TRNGs.

Simulation was performed by Virtuoso Spectre Circuit Simulator. This simulator allows the use of an embedded transient noise feature during simulation which gives a realistic estimation for the internal noise value and behavior inside the device.

In Fig.4, an example of Meta-RO5st simulation is shown. The figure clearly shows that in MS mode the Meta-RO comes to the metastability point.

3.4 Statistical Evaluation

There are several standards and criteria for evaluating random number generators including PRNG and TRNG. FIPS 140-1/2 [3], [4] is one of the most accepted standard series. In FIPS 140-1, four statistical tests are presented for evaluating RNG used in crypto systems. Note that the statistical tests in FIPS 140-2 are almost the same as in FIPS 140-1, except for the thresholds and ranges of each test. (The statistical tests in FIPS 140-2 are stricter than those in FIPS 140-1.) However, in the later version of FIPS 140-2, the statistical requirements for the RNG are omitted as a result of amendment. AIS.31 [1] is a German standard for the necessary properties of secure TRNGs and their evaluations. This standard includes 9 statistical tests for the evaluation of random output from TRNG. Detailed description of the test and methodology on how to use it can be found in [1] and [2]. Note that statistical tests T0–T5 required a relatively strict statistical quality of the sample since they are applied to the output of a post-processing. Furthermore, T1–T4 are exactly the same as the statistical tests in FIPS 140-1. T6 is a uniform distribution test consisting of two sub-tests. T7 is a comparative test for multinomial distributions that consists of two sub-tests. Finally T8 is an entropy test that corresponds to Coron's entropy estimation. Note that the last 3 statistical tests T6–T8 required relatively loose conditions since these tests are applied for the direct output of TRNG.

In this evaluation, we performed statistical tests for the random samples from Spectre simulation. Because of the complexity of analog simulation (large number of parameters, high precision, large number of simulated and stored points, etc.) the obtaining of a big sample was limited. To perform appropriate simulation instead of one long simulation 20 experiments (every for 7 μs) with different noise seed have been run. Sampling period equals 7ns, giving throughput above 140 Mbits/sec. In total a sample of 20,000 bits was obtained. Raw sample inherits Bias = 0.484075796 and Shannon Entropy = 0.999268198. The size of this random sample was too short to apply the original AIS.31 statistical tests. Instead, we used the modified version of AIS.31 with re-estimated boundaries for every test. In Table 1 a summary on the results of the AIS.31 test is shown. Tests T0, T6-2, T7-1, T7-2, and T8 are not available because of the sample size. As it is shown in the table, the generated sample passed the tests, except T1 for FIPS 140-2. Detailed investigation showed that reason of fail was the stronger boundaries for the bias in the FIPS 140-2 test[4].

Table 1. Statistical test on Meta-RO5st (20 kbits simulated by Spectre CDE)

Test	AIS.31	FIPS 140-1	FIPS 140-2
T1: Monobit Test	P	P	F
T2: Poker Test	P	P	P
T3: Run Test	P	P	P
T4: Long Run Test	P	P	P
T5: Autocorrelation Test	P	NA	NA
T6-1: Uniform Test Results	P	NA	NA

3.5 Corners Analysis

One of the major challenges facing semiconductor companies today is how to increase yield. The ability to predict and improve yield becomes even more vital as processes move to geometries under 100 nm. To account for process variations, an IC designer not only has to design for good electrical performance, but also for high manufacturing yield. There are many factors that effect yield. Manufacturing issues such as defect density on the silicon, maturity of the process, and effectiveness of design rules all affect yield. Another factor is how the design reacts to technological process variation and environment conditions (for example, high/low temperatures and voltage fluctuations) simultaneously. So, to be convinced of the robustness of our design to technological process and environment variations, special investigation must be performed.

Corners simulation is perhaps the most widely used method to test for process, temperature, and voltage variations. With this method, a designer determines the worst case corners, or conditions, under which the design will be expected to function. The process variations mean the variation on used pmos and nmos transistors. They can be “slow” or “fast”, so there are possible 4 corners (SS,

[4] The obtained value equaled 9681, while acceptance boundaries were [9725, 10275].

SF, FS and FF). This kind of simulation is very important because parameters of used transistors in real scheme can be very different, causing design malfunctioning.

To estimate the robustness of the proposed design to process and temperature variations (PTVA), the Corners analysis for 65 nm technology library was run in CDE. Process variations ran all 4 possible technology variation sets (FF, SS, FS, and SF) while temperature changed from −25 to 100 with step = 25 degrees of Celsius. Similar to the nominal case, the sampling period equaled 7 ns, giving throughput above 140 Mbits/sec. For every specific PTVA point a 2,600 bits sample was generated. For every PVTA point the bias was estimated, the data is collected in Table 2.

Table 2. Bias estimation for Corners analysis on Meta-RO5st

	Temperature					
PVA	−25	0	25	50	75	100
FF	0.4665	0.4896	0.5135	0.5281	0.5442	0.5565
SS	0.3892	0.3689	0.3792	0.4073	0.4323	0.4515
FS	0.4119	0.4258	0.4377	0.4496	0.4558	0.4577
SF	0.4808	0.5039	0.5046	0.4865	0.4711	0.4554

The analysis of Table 2 showed that the technology process and temperature variation significantly influenced the quality of the generated data. We can propose three approaches to solve this problem.

The *most common method* consists of decreasing the operation rate. In this case, the period of the metastability mode is proportionally increasing, causing longer time for convergence and assuring a metastable state is reached (see Fig. 1b)).

The *second most common approach* consists of applying a post-processing to the raw data to increase the original entropy. There are many post-processing schemes: XOR, von Neumann, resilient, etc [1], [2], [8], [17], [21]–[23]. Von Neumann corrector stands as the most powerful method of significantly reducing the existing bias (in spite of degradation in performance by factor 4 in average). The general method is described in [21], and modern advanced methods are represented in [22] and [23]. Fig. 5 is the result of statistical evaluation of previously generated samples (Meta-RO5st Corners analysis) after post-processing by von Neumann corrector. Since the sample size was too short, only an online test could be applied [1]. This test is intended to detect some kinds of statistical defects from the sampled random sequences. As can be seen in the figure, the post-processed data passed the online test for every PTVA point. The potential throughput was decreased approximately 4 times and was estimated as 35 Mbits/sec.

The *third approach* consists of balancing the design. The parasitic RC characteristic of a digitizer circuit influences the loads of the last inverter in Meta-RO, causing change of the output voltage value from the original metastable level.

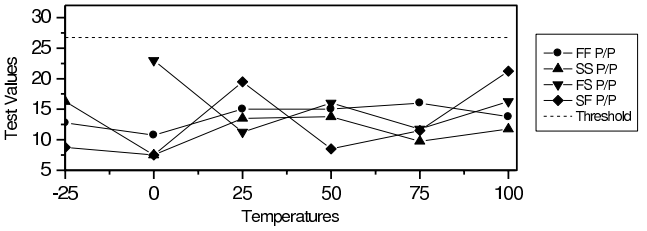

Fig. 5. Online test values via temperature for different process variations after post-processing by von Neumann

Therefore the metastable levels between the last and other inverters in RO are mismatched, causing some bias to the generated data. If the output of every inverter is similarly loaded, then the difference in the metastability level between inverters will be minimized, reducing the bias. So, another approach consists of balancing the digitizer circuits.

Table 3 shows the bias estimation for data generated by a simulation of balanced Meta-RO5st design with decreased operation rate period (T= 20ns allows to get throughput of 50 Mbits/sec). Again, for every specific PTVA point 2,600 bits sample was generated. The analysis showed that only 2 PVTA points (marked as * in the table) are slightly out of the AIS.31 acceptable boundaries [0.475, 0.525]. First, it must be noted that boundaries [0.475, 0.525] were defined for a 20 kbits sample, and for a 2,600 bits sample they could be wider. Also, we believe that further decreasing of the operation rate will refine the bias in those points as well.

Table 3. Bias estimation for Corner analysis on Meta-RO5st (balanced design)

	Temperature					
PVA	−25	0	25	50	75	100
FF	0.5169	0.4940	0.4967	0.4785	0.4924	0.5006
SS	0.5111	0.5075	0.5120	0.5117	0.5111	0.5155
FS	0.4618*	0.4672*	0.4880	0.5016	0.5170	0.5100
SF	0.4757	0.5137	0.5019	0.5110	0.5100	0.5019

Also, similar stable results (with usage of balanced Meta-RO5st design) were obtained for 150 nm semiconductor technology. The properties of Digital TRNG for the following variations were investigated: PVA (FF, FS, SF and SS), temperature (−40, −25, 25 and 125 of Celsius) and supply voltage variation (1.45 V and 1.9 V with 50 mv noise harmonic at 20 MHz and 100 kHz).

Thus, any of the methods listed above (as well a combination) could be used for building a Meta-RO–based TRNG robust to process and environment variations.

4 Investigation in FPGA Technology

In order to provide a proof of the proposed Meta-RO concept we conducted the experiments in FPGA technology (Xilinx XC2V3000–5). We noted that straightforward implementation of the Verilog code of Meta-RO5st design is not possible, because the logic synthesizer performs unnecessary design optimization, causing malfunctioning of Meta-RO. To avoid this, special constraints had to be used. Additionally, every logic function in FPGA is implemented by a look-up table, the dynamic properties of which are different from the properties of inverters or other gates. That is why the designer has to be very careful while implementing a Meta-RO in FPGA.

Direct measurement of the Meta-RO5st analog signal by oscilloscope (see Fig. 6) confirms that Meta-RO5st digital TRNG functioned properly and followed the theoretical assumption discussed above. We can see during metastability mode how the voltage is converged, arriving at metastable level. When the control signal takes a high value, the generation mode is started.

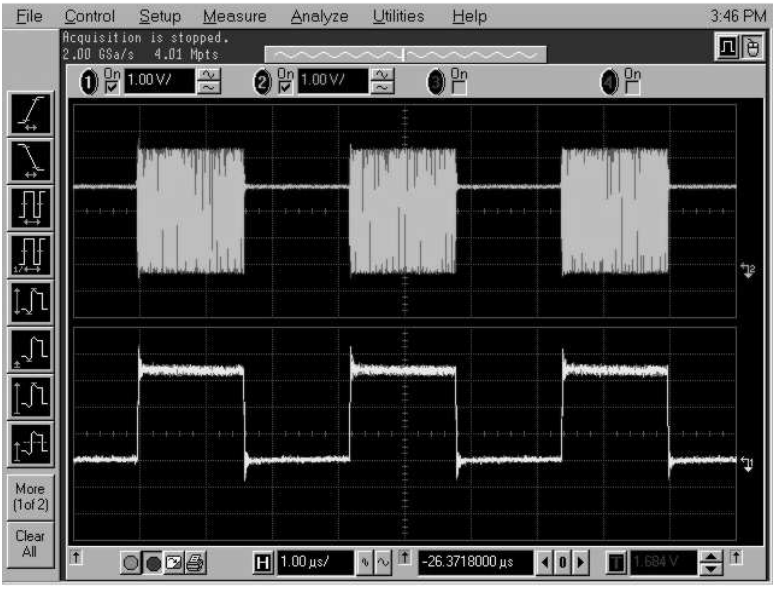

Fig. 6. Random number generation process in FPGA technology. Digital TRNG output is switched between metastable and generation state following the control signal.

To check pseudo randomness properties we used the Dichtl and Golić idea [11] for measuring the data from the same initial conditions. In Fig. 7 the results of the measurement of several consequent D-TRNG runs are shown. In the figure, the horizontal axis is the time, the period of time shown for each run is 100 μs. The vertical axis is the output voltage of the sampled signal. To guarantee the

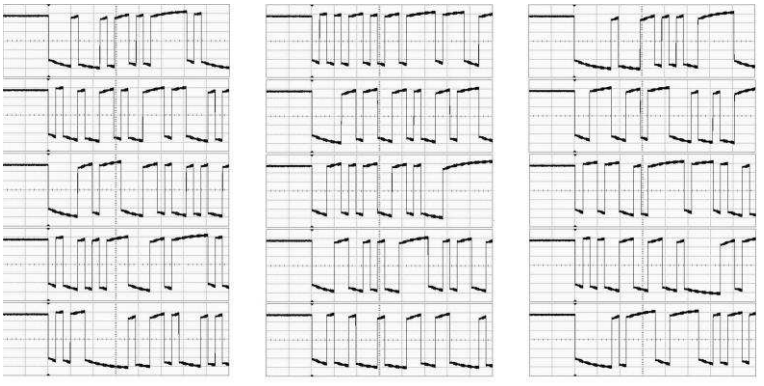

Fig. 7. Consequent runs of Meta-RO5st after restarting. Since after restarting, every generated sample is different, this is evidence that output of the proposed D-TRNG is not defined by a deterministic source.

same initial conditions we must wait a few minutes between consequent runs, powering off the FPGA board. Every run in the figure corresponds to a different sample, i.e., the output of D-TRNG is not defined by a deterministic source.

To confirm the advantages of the proposed method, it is necessary to compare the phase distribution characteristics of Meta-RO and traditional ring oscillator. In [20], Bucci and Luzzi introduced the concept of stateless generator. The stateless hypothesis can be fulfilled by resetting TRNG to a constant value for every state variable in both the entropy source and the post-processor, before the generation of a new bit. For a random number generator built on a traditional RO, this means resetting the RO to some constant value before generating every new bit. Thus, in the following experiments we examined the phase distribution for traditional RO with Reset and Meta-RO5st (unbalanced).

In Fig. 8 the measurement of Meta-RO5st is shown[5]. We measured the time of the first transition of the signal starting from 30 ns to 40 ns after switching to generation mode. Since the period of Meta-RO5st oscillation is about 10 ns, we can interpret this measurement as a phase distribution in generation mode. Compared to the phase distribution of a traditional 5-stage ring oscillator (see Fig. 9), we noted that the phase distribution of Meta-RO5st was spread over the complete period of oscillation. This effect allows faster entropy accumulation for random number generation compared to traditional jitter-based TRNG. Thus, digital TRNG based on Meta-RO provides higher entropy for significantly increased throughput. Additionally, period-wide phase distribution of Meta-RO guarantees some minimal entropy accumulating (far different from zero) in any instance of time during sampling, significantly decreasing the risk of random number quality degradation due to parasitic synchronization of Meta-RO with other processes in the system.

[5] Agilent oscilloscope "Measuring Jitter Using Histogram" feature and methodology has been used for this experiment.

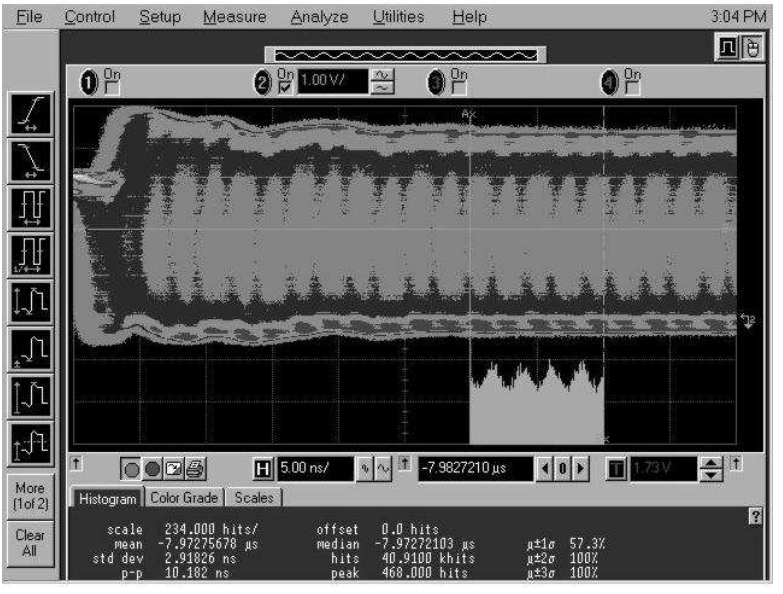

Fig. 8. Histogram of phase distribution in Meta-RO5st digital TRNG. The phase distribution occupies the whole period of the Meta-RO oscillations, significantly increasing the entropy.

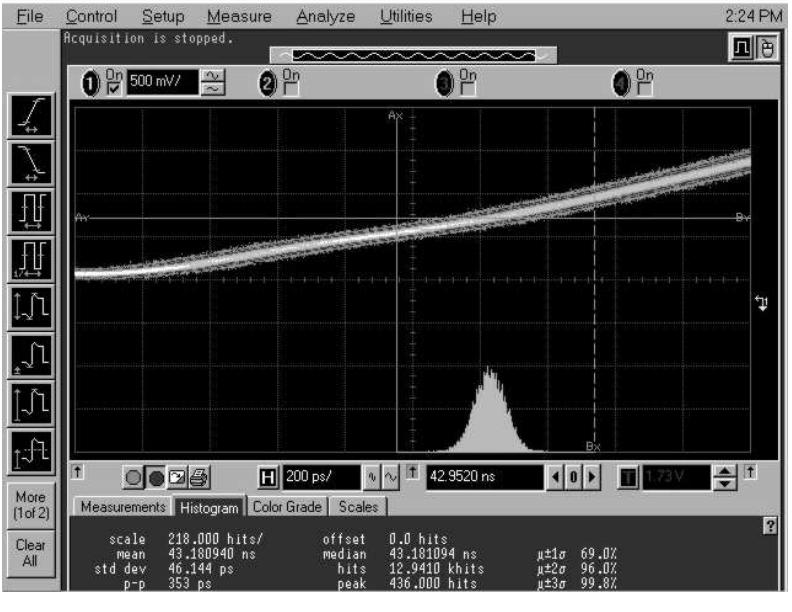

Fig. 9. Histogram of phase distribution in traditional 5-stage RO with reset (measured for rise transition from 30 ns after restart)

Table 4. Summary of the statistical tests on Meta-RO5st (FPGA implementation)

Test Suite	Tests	Without post-processing, %	With post-processing, %
FIPS 140-1/2	T1-T4	68	100
AIS.31 Class P1	T0-T4	68	100
	T5 (Autocorrelation)	100	Not needed
AIS.31 Class P2	T0-T4	68	100
	T5 (Autocorrelation)	100	Not needed
	T6-T8	88	Not allowed
NIST STS	Spectrum test	100	Not needed

For FIPS 140-1/2 and AIS.31 tests, evaluation was made completely over 1 Gbits of data samples. The preliminary investigation showed that the statistical properties of the samples vary during the generation. The reason for such instability can be explained by the fact that the FPGA design is sensitive to temperature fluctuations and voltage supply noise. Improving FPGA operation and environment conditions (using a stable power supply source and installing a cooler over the FPGA) allowed us to obtain more satisfactory results, summarized in Table 4.

As it can be seen from the table, our FPGA design has no correlation problem, i.e., in 1 Gbit of total data there was no single failure in either the AIS.31 T5 Autocorrelation test or the NIST STS Spectrum test[6]. There are still some bias weaknesses, however, which could be fixed by post-processing (where applicable). Thus, our FPGA design successfully passes FIPS 140-1/2 and AIS.31 Class P1, but problems may arise with AIS.31 Class 2. Taking into account the fact that FPGA implementation is not very stable compared to ASIC, we can expect that real ASIC implementation will have no such weaknesses.

5 Conclusion

In this paper, a method for true random number generation was proposed. The highlight of this method lies in the usage of metastability phenomena in the ring oscillator for entropy accumulating, compared to traditional methods based on jitter. For practical realization of this method, a special ring oscillator architecture with the ability to be set in metastable mode was discovered. This ring oscillator is based on digital components only and does not require special custom design.

For validation of the proposed method, a Meta-RO5st (5-stage metastable ring oscillator) component was implemented and simulated in Cadence. Collected samples were tested according to AIS.31 and FIPS 140-1/2 standard

[6] National Institute of Standards and Technology. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22b.pdf

requirements and inherit Bias = 0.48407 and Shannon Entropy = 0.99926 for raw samples. The throughput reached 140 Mbits/sec in nominal conditions. To compensate for process and temperature variations, the sampling rate was decreased, and as a result the throughput reached 35–50 Mbits/sec. The estimated area for 65 nm semiconductor technology is approximately $1\,\mu m^2$ (for Digital TRNG core only).

Physical experiments in FPGA technology showed that phase distribution of the proposed metastable RO occupies the complete oscillating period with stronger entropy value, allowing faster entropy accumulating for random number generation. Thus, Digital TRNG based on Meta-RO provides high entropy for significantly increased throughput. Statistical evaluation showed that our FPGA design could successfully pass FIPS 140-1/2 and AIS.31 Class P1. Further improvements in FPGA operation environment conditions could increase the quality of the proposed TRNG to pass AIS.31 Class P2.

The patent for this method of true random number generation and Meta-RO architecture is pending.

Acknowledgement

We deeply appreciate the support of Markus Dichtl, whose useful comments and notes significantly increase the quality and value of the paper.

References

1. Killmann, W., Schindler, W.: AIS 31: Functionality Classes and Evaluation Methodology for True (Physical) Random Number Generators, version 3.1, Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn (2001)
2. 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)
3. FIPS PUB 140-1: Security requirements for cryptographic modules (1994)
4. FIPS PUB 140-2: Security requirements for cryptographic modules (2001)
5. Fairfield, R., Mortenson, R., Coulthart, K.: An LSI random number generator (RNG). In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 203–230. Springer, Heidelberg (1985)
6. Jun, B., Kocher, P.: The Intel random number generator, White paper for Intel Corporation, Cryptography Research Inc. (April 1999), `http://www.cryptography.com/resources/whitepapers/IntelRNG.pdf`
7. Trichina, E., Bucci, M., De Seta, D., Luzzi, R.: Supplemental Cryptographic Hardware for Smart Cards. IEEE Micro. 21(6), 26–35 (2001)
8. Sunar, B., Martin, W., Stinson, D.: A provably secure true random number generator with built-in tolerance to active attacks. IEEE Trans. Computers 56(1), 109–119 (2007)
9. Schellekens, D., Preneel, B., Verbauwhede, I.: FPGA vendor agnostic true random number generator. In: 16th Int. Conf. Field Programmable Logic and Applications - FPL 2006, pp. 1–6 (2006)

10. Chodowiec, P., Gaj, K.: Very Compact FPGA Implementation of the AES Algorithm. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 319–333. Springer, Heidelberg (2003)
11. Dichtl, M., Golić, J.: High-Speed True Random Number Generation with Logic Gates Only. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 45–62. Springer, Heidelberg (2007)
12. Bock, H., Bucci, M., Luzzi, R.: Offset-compensated oscillator-based random bit source for security applications. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 268–281. Springer, Heidelberg (2004)
13. 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)
14. Epstein, M., Hars, L., Krasinski, R., Rosner, M., Zheng, H.: Design and implementation of a true random number generator based on digital circuits artifacts. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 152–165. Springer, Heidelberg (2003)
15. Tkacik, T.: A hardware random number generator. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 450–453. Springer, Heidelberg (2003)
16. Dichtl, M.: How to predict the output of a hardware random number generator. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 181–188. Springer, Heidelberg (2003)
17. Golić, J.D.: New methods for digital generation and postprocessing of random data. IEEE Trans. Computers 55(10), 1217–1229 (2006)
18. Horstmann, J., Eichel, H., Coates, R.: Metastability behavior of CMOS ASIC flip-flops in theory and test. IEEE J. Solid-State Circuits 24(1), 146–157 (1989)
19. Kacprzak, T., Albicki, A.: Analysis of metastable operation in RS CMOS flip-flops. IEEE J. Solid-State Circuits 22(1), 57–64 (1987)
20. Bucci, M., Luzzi, R.: Design of testable random bit generators. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 147–156. Springer, Heidelberg (2005)
21. Neumann, J.: Various techniques for use in connection with random digits. In: Von Neumann's Collected Works, vol. 5, pp. 768–770. Pergamon (1963)
22. Peres, Y.: Iterating von Neumann's Procedure For Extracting Random Bits. The Annals of Statistics 20(3), 590–597 (1992)
23. Juels, A., Jakobsson, M., Shriver, E., Hillyer, B.: How to turn loaded dice into fair coins. IEEE Trans. Inf. Theory 46(3), 911–921 (2000)