PWT 2011

An Auxiliary Source of Randomness for Combined TRNG Based on Ring Oscillators

Łukasz Matuszewski
Faculty of Electronics and Telecommunications
Poznan University of Technology
Poznan, Poland
lukasz.matuszewski@et.put.poznan.pl

Mieczyslaw Jessa
Faculty of Electronics and Telecommunications
Poznan University of Technology
Poznan, Poland
mjessa@et.put.poznan.pl

***Abstract*—This paper presents the methodology of choosing an auxiliary source of randomness to improve the properties of a combined true random number generator consisting of ring oscillators. Not only very good statistical properties are needed in cryptographic application, but we also desire unpredictability, of a degree that is increased with added auxiliary source of randomness. To assess the unpredictability of output bits, the restart mechanism proposed in earlier papers was used.**

Keywords—true random number generator, ring oscillator, cryptography, FPGA

I. Introduction

TRUE random number generators (TRNG) are one of the basic cryptographic elements. There are some methods of producing random sequences. One is based on sampling physical phenomena such as thermal noise or phase jitter. The output sequence from this kind of generator cannot provide enough randomness for the generator to pass all known statistical tests. Therefore, the use of some additional mathematical operations called post-processing is needed. Post-processing eliminates the bias present in the output bit stream. Some of post-processing blocks employ linear feedback shift registers (LFSR) or cellular automata shift registers (CASR) like in [1], other blocks use von Neumann corrector, hash functions, or other post-processing methods [2], [3]. Another method to improve the statistical properties of output bit stream is a combination of independent randomness sources [3]. Digital TRNG can be implemented in reconfigurable devices such as FPGA; it is preferable because the generator should be consistent part of every cryptographic system. One of the implementations was proposed by Sunar *et al.* in [3]. This generator uses the phase jitter in ring oscillators composed of inverters. The outputs of ring oscillators are XOR combined, and then sampled using a low frequency clock, producing random bits. The generator of Sunar *et al.* produces sequences with high entropy but to achieve good statistical properties, a large number of source ring oscillators are needed. Wold and Tan in [4] proposed a modification of Sunar's TRNG. They added D flip-flop to each output of source ring oscillator before the XOR gate. This procedure allowed them to significantly reduce the number of source generators. The method was next optimized in speed of producing random bits in the FPGA [5]. Although the Wold's and Tan's generator provides sequences, which pass all statistical tests, but – which has been proved – it can also be done when the source ring oscillators do not produce any jitter [6].

As M. Jessa and M. Jaworski showed in [7], even a small amount of randomness is accumulated as a number of ring oscillators used in Wold's and Tan's method. Nevertheless that the produced sequences pass all statistical tests, a big amount of deterministic factor is present [7], [8]. To improve the unpredictability of output bits an auxiliary source of randomness (ASR) was used. An additional source of randomness perturbs the output signal from each ring oscillator. This makes the output bits more unpredictable, i.e., more random and can increase the bit rate. To distinguish between randomness and pseudo randomness, the authors of this article used the restart mechanism and the chi-square test.

In this paper we show the methodology of choosing and testing an auxiliary source of randomness for combined TRNG, based on the ring oscillator sampling method.

II. Circuit Description

A. Combined TRNG

The scheme of combined TRNG is shown in Figure 1. The generator uses the timing jitter found in single ring oscillator (RO) to produce random bits. The output signal from each RO is sampled. The D-type flip-flop is triggered by the quartz oscillator with frequency f_L, lower than frequency f_H of ring oscillator [9]. A delay element τ can be realized with an even number of inverters, a chain of latches or a delay line built in many FPGAs. The outputs of K with $K > 1$ of flip-flops are summed modulo 2, i.e. with the XOR operation.

In the experiment we used the scheme shown in Figure 2. Because the number of inputs in LUT blocks is limited to 6 (Xilinx XC5VLX50T), combining the streams of bits takes place in 3 steps. To determine the quality of generated sequences, the bits were sent via a buffer and USB 2.0 interface to personal computer (PC). We assumed that the source generators had frequencies, $f_{H,k}$, $k = 1, 2, \ldots, K$, which

were not lower than f_L. Frequencies $f_{H,\,k}$ for k = 1, 2, …, K were higher than 300MHz. The signal having frequency f_L was produced by the quartz oscillator, built into the evaluation board (ML505) containing Virtex-5. The placement of all elements depends on the software provided by the manufacturer of the FPGA. We did not perform any manual corrections for the localizations proposed by this software.

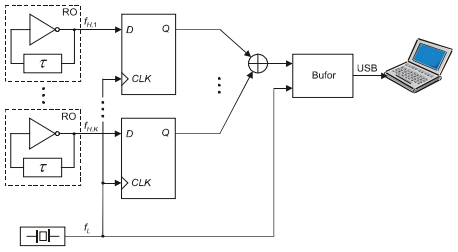

Fig. 1. TRNG using K ring oscillators [7]

This generator can be easily implemented in any type of FPGA and reaches relatively high output bit rate [5], [7]. In the output bit stream two components are present: the nondeterministic component, which is desirable in cryptographic applications and the deterministic component, which cannot be removed [10]. Both of them influence the statistical properties of sequences but only the nondeterministic pattern can provide true randomness. As shown in [7], [8], the more source generators, the less amount of pseudorandom component in the output bits stream. A huge number of source generators is not suitable because of space occupation in FPGA and power consumption. It can also lead to local heating and could cause a destruction of the device.

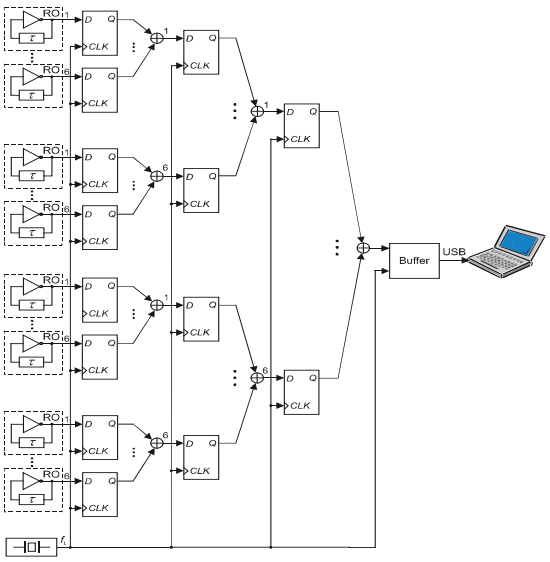

Fig. 2. A modification of the generator from Figure 1 preserving its functionality [8]

B. Enhancing the randomness of combined TRNG

In [8] authors showed experimentally that the use of an auxiliary source of randomness could significantly decrease number of source ROs. As extra source of randomness (Figure 3) they used Galois ring oscillator (GARO) with polynomial (1), because GARO produces much greater amount of nondeterministic phase jitter than a single RO [11], [12].

$$f(x) = x^{31} + x^{27} + x^{23} + x^{21} + x^{20} + x^{17} + x^{16} + x^{15} + x^{13} + x^{10} + x^{9} + x^{8} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1 \quad (1)$$

The reason behind this choice is the result of restarts, explained in section C, which is significantly better for a GARO than for a single RO [14]. A combined TRNG with signals of ROs perturbed by a GARO is shown in Figure 3.

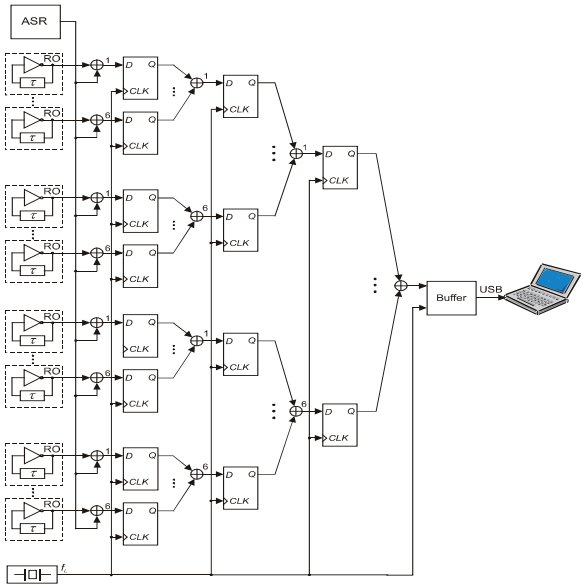

Fig. 3. A combined TRNG from Figure 2 with an ASR [8]

C. The restart mechanism

Two basic methods were proposed to distinguish pseudo randomness and true randomness for ring oscillator-based TRNGs. In both methods, we repeat experiments many times, starting from identical initial conditions. Because pseudo randomness is deterministic, it shows identical behavior in each repetition of the experiment. True randomness provides different behavior for identical initial conditions. To ensure identical initial conditions, we replaced the inverters shown in Figures 1, 2 and 3 with NAND gates triggered by an external signal. In the method described in [11], the authors recorded 1000 restarts of a ring oscillator, Fibonacci ring oscillator (FIRO) and Galois ring oscillator (GARO). The amount of true randomness was measured by computing of the standard deviation of the output voltage as a function of time. If this standard deviation was relatively large, then extracting one bit of true randomness by sampling was easy and reliable [11]. The second method does not assess true randomness contained in the output of a single ring oscillator, but it checks the quality of sequences obtained for successive positions of a bit in time sequences produced by restarts [7].

Therefore, this method is more suitable for assessing the randomness of TRNGs containing more than one RO. To explain this method, let us assume that during one restart we produce M bits. The restarts are then repeated N times. At the end of the experiment, we obtained M, N-bit sequences. If the observed behavior is deterministic, each restart should provide the same sequence. If the behavior depends also on a non-deterministic factor or factors, the sequences starting from the same initial conditions should diverge. In this case, we expect that the N-element sequences should be random from a certain $m = m_{min}$, $m = 1, 2, \ldots, M$ [7]. If the observed behavior depends only on true randomness, the N-bit sequences should be random from $m_{min} = 1$. To assess the randomness of N-bit sequences, a simple statistical test – the chi-square test – was proposed. If the test is satisfied, then the number of ones in an N-bit sequence is comparable with the number of zeros, and there is no reason to reject the hypothesis (with significance level α) that the restarts provide zeros and ones with the same probability. The authors of [7] have observed that such a situation occurs for relatively large m_{min} for the TRNG described by Wold and Tan. The authors of [7] have observed that the value of m_{min} decreases with an increase in the number of source generators.

The value of statistic χ^2 is computed as

$$\chi^2 = \frac{\left(N_0 - N \cdot P_0\right)^2}{N \cdot P_0} + \frac{\left(N_1 - N \cdot P_1\right)^2}{N \cdot P_1}, \qquad (2)$$

where N_0 is the number of zeros in an N-bit sequence *and* N_1 *is* the number of ones in this sequence. $N \cdot P_1$ is the expected number of zeros and $N \cdot P_1$ is the expected number of ones. We assumed that during a single restart $M = 20000$ bits were produced. The number N of restarts was equal to 2048. For significance level $\alpha = 0.01$ the critical χ^2_c value is equal to 6.635. A sequence composed of $N = 2048$ bits passes the chi-square test if and only if $\Delta < 116.5696\ldots$, where Δ is the difference between the number of zeros and the number of ones. We can write that $p = 1 - p'$, where p is the probability that the number of zeros differs from the number of ones by at least 117 and p' is the probability that the number of zeros differs from the number of ones by no more than 116. Because N is even, Δ can be only be even, including $\Delta = 0$. For a perfect random source, the probability that a 2048-bit sequence passes the chi-square test is the sum of probabilities that $\Delta = 0, \Delta = 2, \ldots, \Delta = 116$. Because the positions of zeros and ones in a 2048-bit sequence are arbitrary, we can write that

$$p' = \frac{1}{2^{2048}}\left[C_{2048}^{1024} + \sum_{i=1}^{58}\left(C_{2048}^{1024-i} + C_{2048}^{1024+i}\right)\right] = 0.990289... \qquad (3)$$

and $p = 0.009711\ldots$ If 2048-bit sequences are equally probable, they fail the chi-square test statistically every $1/p \approx 103$ sequences. Two successive sequences do not pass the same test every $1/p^2 \approx 10604$ sequences and three successive sequences fail every $1/p^3 \approx 10^6$ sequences. If we find such m, $m = 1, 2, \ldots, 2000$ that the sequences obtained for m, m - 1 and m - 2 do not pass the chi-square test, we conclude that the source is not random. If among 20000 sequences we find one or two successive sequences that fail the chi-square test, we assume that they may be produced by a perfect random source. Therefore, among 20000 results of the chi-square test, we search for the greatest m for which the 2048-bit sequences produced for m, m - 1 and m - 2 fail the test. For $j > m$ there is no reason to reject the hypothesis that the 2048-bit sequences are produced by a random source. The smallest j is equal to $m + 1$ and it is denoted as m_{min}.

D. Results of testing FIRO and GARO with the use of the restart mechanism

The Galois and Fibonacci ring oscillators consist of inverters connected together with XOR. GARO and FIRO are defined by a feedback polynomial, as in classical LFSR Galois and Fibonacci type, but memory elements are replaced with the inverters. GARO and FIRO are shown in Figure 4 and Figure 5. In the same way as in LFSR, the feedback polynomial should be primitive. In FIRO the number of inverters can be chosen either odd or even, except 2. On the other hand, in GARO it should be odd. The output signal can be taken from any inverter output in FIRO, however in GARO it should be the last one [11]. As it was mentioned, also for both configurations – GARO and FIRO – the output of oscillating signal comprises both pseudo randomness and true randomness. In a real circuit true randomness is the result of random transition times and propagation delays in all circuit paths and gates, because of shot noise and thermal noise [13].

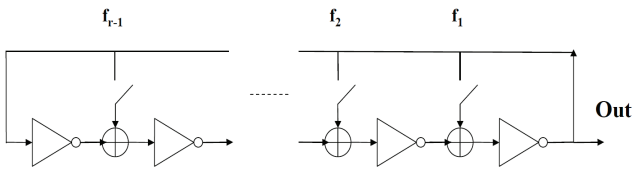

Fig. 4. Galois ring oscillator [11]

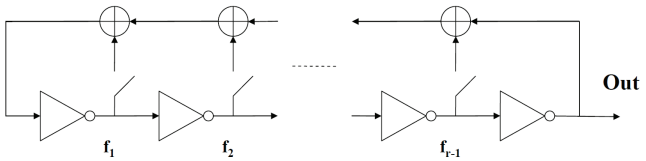

Fig. 5. Fibonacci ring oscillator [11]

In order to assess which GARO and FIRO can be used as potential ASR, we took advantage of the restart mechanism and examined the following feedback polynomials [11]:

$$f(x) = x^7 + x^5 + x + 1 \qquad (4)$$
$$f(x) = x^7 + x^6 + x^2 + 1 \qquad (5)$$
$$f(x) = x^{15} + x^{14} + x^7 + x^6 + x^5 + x^4 + x^2 + 1 \qquad (6)$$
$$f(x) = x^{20} + x^{18} + x^{16} + x^{15} + x^{13} + x^{12} + x^5 + x^4 + x^2 + 1 \qquad (7)$$
$$f(x) = x^{21} + x^{19} + x^{17} + x^{16} + x^7 + x^3 + x^2 + 1 \qquad (8)$$
$$f(x) = x^{31} + x^{27} + x^{23} + x^{21} + x^{20} + x^{17} + x^{16} + x^{15} + x^{13} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1 \qquad (9)$$

The constructions mentioned above were implemented in Xilinx Virtex-5 FPGA (XC5VLX50T). The outputs were sampled and the obtained bits were sent via USB 2.0 to personal computer. The results are given in Table I for sampling frequency f_L of 100MHz. We can conclude that for all polynomials FIRO does not produce any true random bits. Consequently, FIROs are not suitable for use as ASR.

TABLE I
THE RESULTS OF RESTARTS

Polynomial	FIRO		GARO	
	m_{min}	χ^2	m_{min}	χ^2
(4)	19999	874.143	190	5.281
(5)	19994	1.220	592	2.673
(6)	19734	0.281	19720	6.570
(7)	19999	9.298	19999	16.173
(8)	19955	0.281	19204	0.945
(9)	19999	39.382	3	0.281

The case of GARO was different. We can notice that GARO with the feedback polynomial (9) can be effectively applied as ASR because every 3rd bit of generated sequence is true random. Also GAROs with the feedback polynomials (4) and (5) are noteworthy. Polynomials (6), (7) and (8) have very poor prosperities in both types of generators (i.e. FIRO and GARO). It is interesting that longer polynomial does not necessarily means more true randomness.

III. CONCLUSION

In this experiment we have shown that the restart mechanism is suitable to assess the randomness of Galois and Fibonacci ring oscillators applied as auxiliary source of randomness. We have shown that using the restart mechanism and chi-square test on the sampled signal, we can choose an ASR that significantly reduces the resources of FPGA necessary to produce random bits. Unfortunately, there is no theory explaining how to create or choose polynomial for GARO or FIRO to start oscillating and producing high-speed fluctuations and eventually true random bits.

REFERENCES

[1] T. E. Tkacik. "A Hardware random number generator," i*n Proc. of CHES'2002,* pp. 450~453.

[2] Vasyltsov, E. Hambardzumyan, Y.-S. Kim, and B. Karpinskyy, "Fast digital TRNG based on metastable ring oscillator," in *Proc. Workshop Cryptograph. Hardware Embed. Syst. CHES'2008*, Washington, 2008, LNCS 5154, pp. 164-180.

[3] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Trans., Comput.*, vol. 56, pp. 109-119, Jan. 2007.

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

[5] K. Wold and S. Petrović, "Optimizing speed of a true random number generator in FPGA by spectral analysis," *Proc. of Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT'09*, 24-26 Nov. 2009, pp. 1105-1110.

[6] N. Bochard, F. Bernard, and V. Fischer, "Observing the randomness in RO-based TRNG," *Proc. of ReConFig 2009,* 9-11 Dec. 2009, pp. 237-242.

[7] M. Jessa, M. Jaworski, "Randomness of a combined TRNG based on the ring oscillator sampling method," *Proc. of International Conference on Signals and Electronic Systems, ICSES'10*, Sept. 2010, pp. 323-326.

[8] M. Jessa, Ł. Matuszewski, "Enhancing the randomness of a rombined true random number generator based on the ring oscillator sampling method," *Proc. of ReConFig 2011,* Cancun 2011, in press.

[9] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti M. Varnonuovo, "A high-speed oscillator-based truly random number source for cryptographic applications on a smartcard IC," *IEEE Trans. on Computers*, vol. 52, pp. 403-409, April 2003.

[10] V. Fischer, A. Aubert, F. Bernard, B. Valtchanov, J.-L. Danger, and N. Bochard, "True random number generators in configurable logic devices," *Project ANR – ICTeR,* February 12, 2009.

[11] M. Dichtl and J. D. Golić, "High speed true random number generation with logic gates only, *Proc. Workshop Cryptograph. Hardware Embed. Syst. CHES'2007*, Vienna, 2007, LNCS 4727, pp. 45-62, 2007.

[12] I. G. Târşa, G. D. Budariu, and C. Grozea, "Study on a true random number generator design for FPGA," *Proc. of Conference on Communications, COMM'2010*, June 2010, pp.461-464.

[13] Ü. Güler, S. Ergün, G. Dündar, "A digital IC random number generator with logic gates only," *Proc. of 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS),* Dec. 2010, pp. 239-242.

[14] B. Valtchanov, A. Aubert, F. Bernard, and Fischer, "Modeling and observing the jitter in ring oscillators implemented in FPGAs," *Proc. IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, DDECS'08*, April 2008, pp. 158-163.