A 32-bit Tunable True Random Number Generator with an enhanced DCM for Xilinx FPGA

Nakka Vaibhav Warlu [1] **G. Chenna Kesava Reddy** [2] **N. Sathish Kumar** [3]
nvaibhavwarlu@gmail.com[1] gckreddy2001@gmail.com[2] nsathish528@gmail.com[3]
[1]PG Scholar, VLSI, Teegala Krishna Reddy Engineering College, Hyderabad.
[2]Associate Professor, Dept of ECE, Teegala Krishna Reddy Engineering College, Hyderabad.
[3]Professor, Dept of ECE, Teegala Krishna Reddy Engineering College, Hyderabad.

Abstract: **Random numbers are required for cryptographic applications such as IT security products, smart cards etc. Hardware based random number generators are widely employed. Cryptographic algorithms are implemented on Field Programmable Gate Arrays (FPGAs). In this brief, we present a highly efficient and tunable TRNG based on the principle of beat frequency detection, specifically for Xilinx-FPGA-based applications. The main advantages of the proposed TRNG are its on-the-fly tunability through dynamic partial reconfiguration to improve randomness qualities. We describe the mathematical model of the TRNG operations and experimental results for the circuit implemented on Xilinx. The proposed TRNG has low hardware footprint and built-in bias elimination capabilities.**

Index Terms—Digital clock manager (DCM), dynamic partial reconfiguration (DPR), field-programmable gate arrays (FPGAs), true random number generator (TRNG).

I. INTRODUCTION

True random numbers and physical nondeterministic random number generators (RNGs) seem to be of an ever-increasing importance. Random numbers are essential in cryptography (mathematical, stochastic, and quantum), Monte Carlo calculations, numerical simulations, statistical research, randomized algorithms, lotteries, etc. Today, true random numbers are most critically required in cryptography and its numerous applications to our everyday life: mobile communications, e-mail access, online payments, cashless payments, ATMs, e-banking, Internet trade, point of sale, prepaid cards, wireless keys, general cyber security, distributed power grid security (SCADA), etc.

In cryptography, where due to Kerckhoffs' principle all parts of protocols are publicly known except some secret (the key or other information) known only to the sender and the recipient, it is clear that the secret must not be calculable by an eavesdropper, i.e., it must be random.

True RNGs are generally constructed such that the correlation among bits is small which is, namely, the idea of randomness. In some cases the physical system that is measured is being "reset" to an initial condition after production of each bit in order to reduce autocorrelation. Therefore in most cases only a few lowest-order autocorrelation coefficients are significant, ideally only the first one, which is named autocorrelation and denoted by a. There are very many constructions or true RNGs and research is still getting impetus, but in our view one can roughly classify the present art into four families:

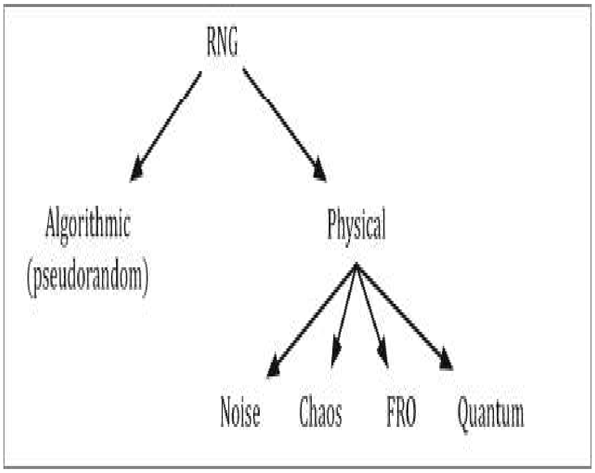


Fig. 1 Classification of random number generators

- Noise-based RNGs
- Free-running oscillator RNGs
- Chaos RNGs
- Quantum RNGs

The tree of RNGs is illustrated in Fig. 1. Mathematical, pseudorandom generators can also be divided into several categories depending on the type of algorithm used.

The major contribution of this brief is the development of an architecture which allows on-the-fly tunabilty of statistical qualities of a TRNG by utilizing DPR capabilities of modern FPGAs for varying the digital clock manager (DCM) modeling parameters. To the best of our knowledge, this is the first reported work which incorporates tunability in a TRNG. This approach is only applicable for Xilinx FPGAs which provide programmable clock generation mechanism and capability of DPR. DPR is a relatively new enhancement in FPGA technology, whereby modifications to predefined portions of the FPGA logic fabric are possible on-the-fly, without affecting the normal functionality of the FPGA. Xilinx clock management tiles (CMTs) contain a dynamic reconfiguration port (DRP) which allows DPR to be performed through much simpler means. Using DPR, the clock frequencies generated can be changed on-the-fly by adjusting the corresponding DCM parameters. DPR via DRP is an added advantage in FPGAs as it allows the user to tune the clock frequency as per the need. Design techniques exist to prevent any malicious manipulations via DPR which in other ways may detrimentally affect the security of the system. The goal of this brief is the design, analysis, and implementation of an easy-to-design, improved, low-overhead, and tunable TRNG for the FPGA platform.

II. LITERATURE SURVEY

A PUF-enabled secure architecture for FPGA-based IoT applications.

The Internet of Things (IoT) is a dynamic, ever-evolving “living” entity. Hence, modern Field Programmable Gate Array (FPGA) devices with Dynamic Partial Reconfiguration (DPR) capabilities, which allow in-field non-invasive modifications to the circuit implemented on the FPGA, are an ideal fit. Usually, the activation of DPR capabilities requires the procurement of additional licenses from the FPGA vendor. In this work, we describe how IoTs can take advantage of the DPR capabilities of FPGAs, using a modified DPR methodology that does not require any paid “add-on” utility, to implement a lightweight cryptographic security protocol. We analyze possible threats that can emanate from the availability of DPR at IoT nodes, and propose possible solution techniques based on Physically Unclonable Function (PUF) circuits to prevent such threats.

A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.

This paper discusses some aspects of selecting and testing random and pseudorandom number generators. The outputs of such generators may be used in many cryptographic applications, such as the generation of key material. Generators suitable for use in cryptographic applications may need to meet stronger requirements than for other applications. In particular, their outputs must be unpredictable in the absence of knowledge of the inputs. Some criteria for characterizing and selecting appropriate generators are discussed in this document. The subject of statistical testing and its relation to cryptanalysis is also discussed, and some recommended statistical tests are provided. These tests may be useful as a first step in determining whether or not a generator is suitable for a particular cryptographic application. However, no set of statistical tests can absolutely certify a generator as appropriate for usage in a particular application, i.e., statistical testing cannot serve as a substitute for cryptanalysis. The design and cryptanalysis of generators is outside the scope of this paper.

Brief description of the basic BFD-TRNG model and the DPR methodology utilizing DRP ports available in Xilinx CMTs.

Single-Phase BFD-TRNG Model

The BFD-TRNG circuit is a fully digital TRNG, which relies on jitter extraction by the BFD mechanism, originally implemented as a 65-nm CMOS ASIC. The structure and working of the (single phase) BFD-TRNG can be summarized as follows, in conjunction with Fig. 2.

1) The circuit consists of two quasi-identical ring oscillators (let us term them as ROSCA and ROSCB), with similar construction and placement. Due to inherent physical randomness originating from process variation effects associated with deep sub-micrometer CMOS manufacturing, one of the oscillators (e.g., ROSCA) oscillates slightly faster than the other oscillator (ROSCB). In addition, the authors proposed to employ trimming capacitors to further tune the oscillator output frequencies.

2) The output of one of the ROs is used to sample the output of the other, using a D flip-flop (DFF). Without loss of generality, assume that the output of ROSCA is fed to the D-input of the DFF, while the output of ROSCB is connected to the clock input of the DFF.

3) At certain time intervals (determined by the frequency difference of the two ROCs), the faster oscillator signal passes, catches up, and overtakes the slower signal in phase. Due to random jitter, these capturing events happen at random intervals, called “beat frequency intervals.” As a result, the DFF outputs a logic-1 at different random instances.

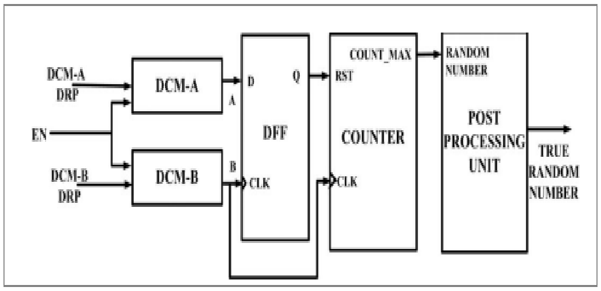


Fig.3 Overall architecture of the proposed DCM-based tunable BFD-TRNG

4) A counter controlled by the DFF increments during the beat frequency intervals and gets reset due to the logic-1 output of the DFF. Due to the random jitter, the free running counter output ramps up to different peak values in each of the count-up intervals before getting reset.

5) The output of the counter is sampled by a sampling clock before it reaches its maximum value.

6) The sampled response is then serialized to obtain the random bit stream.

B. Shortcoming of the BFD-TRNG

One shortcoming of the previous BFD-TRNG circuit is that its statistical randomness is dependent on the design quality of the ring oscillators. Any design bias in the ring oscillators might adversely affect the statistical randomness of the bit stream generated by the TRNG. Designs with the same number of inverters but different placements resulted in varying counter maximas. Additionally, the same ring-oscillator-based BFD-TRNG implemented on different FPGAs of the same family shows distinct counter maxima. Unfortunately, since the ring oscillators are free-running, it is difficult to control them to eliminate any design bias. The problem is exacerbated in FPGAs, where it is often difficult to control design bias because of the lack of fine-grained designer control on routing in the FPGA design fabric. A relatively simple way of tuning clock generator hardware primitives on Xilinx FPGAs, particularly the phase-locked loop (PLL) or the DCM as used in this work, is by enabling dynamic reconfiguration via the DRPs. Once enabled, the clock generators can be tuned to generate clock signals of different frequencies by modifying values at the DRPs on-the-fly, without needing to bring the device offline.

III. PROPOSED SYSTEM

TUNABLE BFD-TRNG FOR FPGA-BASED APPLICATIONS

A. Design Overview

Fig. 3 shows the overall architecture of the proposed TRNG. In place of two ring oscillators, two DCM modules generate the oscillation waveforms. The DCM primitives are parameterized to generate slightly different frequencies by adjusting two design parameters M (multiplication factor) and D (division factor). In the proposed design, the source of randomness is the jitter presented in the DCM circuitry. The DCM modules allow greater designer control over the clock waveforms, and their usage eliminates the need for initial calibration . Tunability is established by setting the DCM parameters on-the-fly using DPR capabilities using DRP ports. This capability provides the design greater flexibility than the ring-oscillator-based BFDTRNG. The difference in the frequencies of the two generated clock signals is captured using a DFF. The DFF sets when the faster oscillator completes one cycle more than the slower one (at the beat frequency interval). A counter is driven by one of the generated clock signals and is reset when the DFF is set. Effectively, the counter increases the throughput of the generated random numbers. The last three LSBs of the maximum count values reached by the count were found to show good randomness properties.

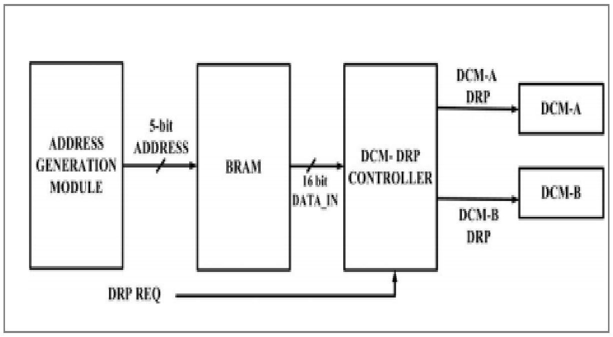


Fig. 4. Architecture of tuning circuitry

Additionally, we have a simple post processing unit using a Von Neumann corrector (VNC) [5] to eliminate any biasing in the generated random bits. VNC is a well-known low overhead scheme to eliminate bias from a random bit stream. In this scheme, any input bit “00” or “11” pattern is eliminated; otherwise, if the input bit pattern is “01” or

"10," only the first bit is retained. The last three LSBs of the generated random number are passed through the VNC. The VNC improves the statistical qualities at the cost of slight decrease in throughput.

B. Tuning Circuitry

The architecture of the tuning circuitry is shown in Fig. 3. The target clock frequency is determined by the set of parameter values actually selected. The random values reached by the counter as well as the jitter are related to the chosen parameters M and D. This makes it possible to tune the proposed TRNG using the predetermined stored M and D values. As unrestricted DPR has been shown to be a potential threat to the circuit [6], the safe operational value combinations of the D and M parameters for each DCM are predetermined during the design time and stored on an on-chip block RAM (BRAM) memory block in the FPGA.

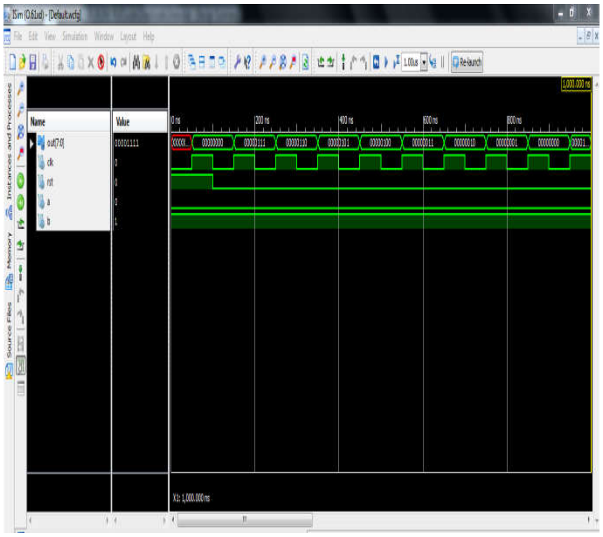
There are actually two different options for the clock generators—one can use the PLL hard macros available on Xilinx FPGAs or the DCMs. We next describe analytical and experimental results which compelled us to choose DCM in favor of the PLL modules for clock waveform generation.

IV. RESULTS

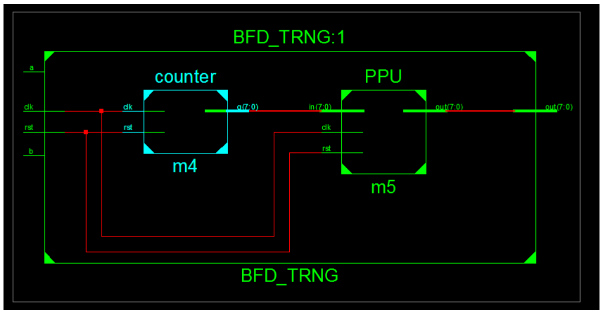
Different blocks of proposed system are designed coded in VERILOG HDL, simulated in I simulator and Xilinx ISE is the software tool used for FPGA synthesis.

PROPOSED RESULT:

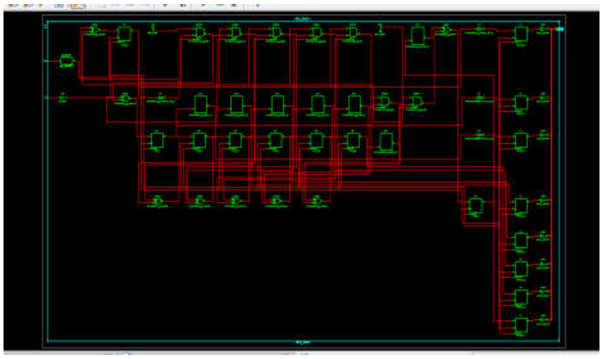
Simulation for 8 bit:



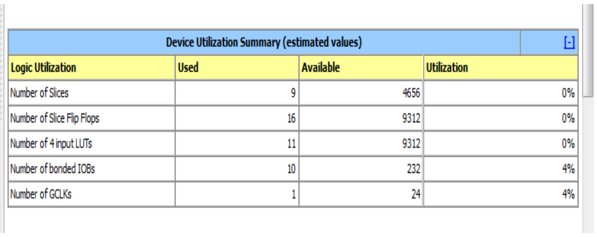
RTL Schematic:



Technology Schematic:

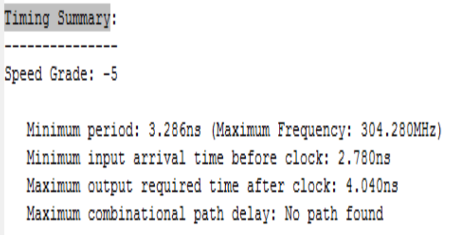


Design Summary:



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	9	4656	0%
Number of Slice Flip Flops	16	9312	0%
Number of 4 input LUTs	11	9312	0%
Number of bonded IOBs	10	232	4%
Number of GCLKs	1	24	4%

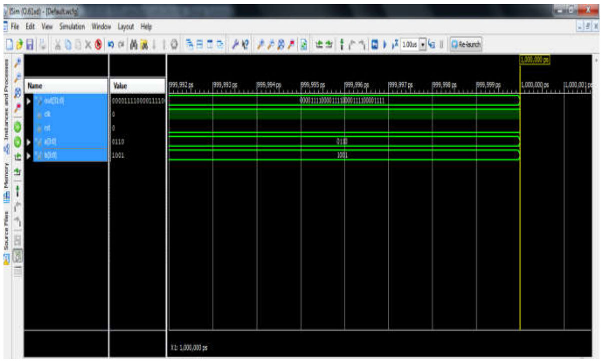
Timing Summary:



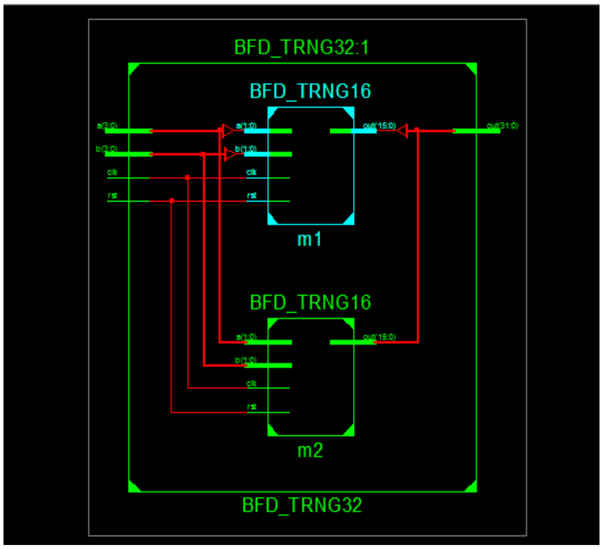
```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 3.286ns (Maximum Frequency: 304.280MHz)
   Minimum input arrival time before clock: 2.780ns
   Maximum output required time after clock: 4.040ns
   Maximum combinational path delay: No path found
```

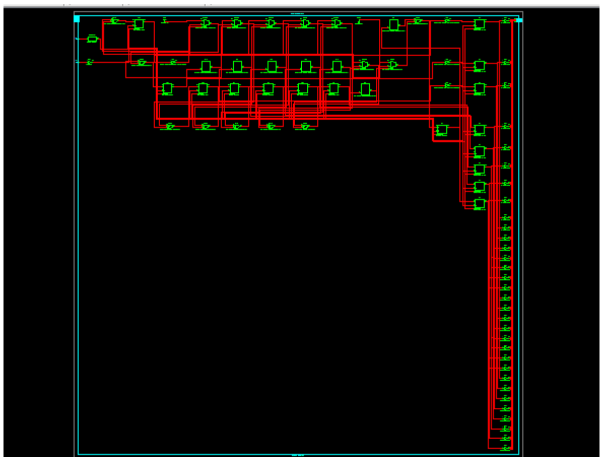
Simulation for 32 bit:



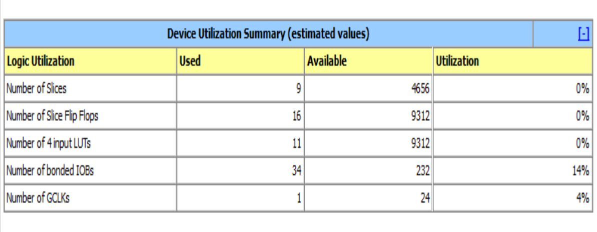
RTL Schematic:



Technology Schematic:

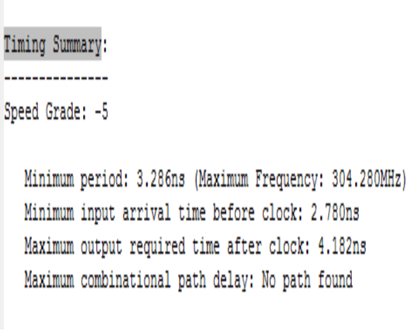


Design Summary:



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	9	4656	0%
Number of Slice Flip Flops	16	9312	0%
Number of 4 input LUTs	11	9312	0%
Number of bonded IOBs	34	232	14%
Number of GCLKs	1	24	4%

Timing Summary:



```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 3.286ns (Maximum Frequency: 304.280MHz)
   Minimum input arrival time before clock: 2.780ns
   Maximum output required time after clock: 4.182ns
   Maximum combinational path delay: No path found
```

V. CONCLUSION

We have presented an improved fully digital tunable TRNG for FPGA-based applications, based on the principle of BFD and clock jitter, and with built-in error-correction capabilities. The TRNG utilizes this tunability feature for determining the degree of randomness, thus providing a high degree of flexibility for various applications.

REFERENCES

[1] Virtex-5 FPGA Configuration User Guide UG 191 (v3.11) Xilinx Inc., San Jose, CA, USA, Accessed: May 2016. [Online]. Available: www. wilinx.com/support/documentation/user_guides/ug19.pdf.

[2] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, “A PUF-enabled secure architecture for FPGA-based IoT applications,” IEEE Trans. MultiScale Comput. Syst., vol. 1, no. 2, pp. 110–122, Apr.–Jun. 1, 2015.

[3] Q. Tang, B. Kim, Y. Lao, K. K. Parhi, and C. H. Kim, “True random number generator circuits based on single- and multi-phase beat frequency detection,” in Proc. IEEE Custom Integr. Circuits Conf., Sep. 2014, pp. 1–4.

[4] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, DTIC Document, Tech. Rep., 2001.

[5] J. Von Neumann, “Various techniques used in connection with random digits,” Nat. Bureau Standards Appl. Math. Ser., vol. 12, pp. 36–38, 1951.

[6] A. P. Johnson, S. Saha, R. S. Chakraborty, D. Mukhopadyay, and S. Gören, “Fault attack on AES via hardware Trojan insertion by dynamic partial reconfiguration of FPGA over Ethernet,” in Proc. 9th WESS, Oct. 2014, pp. 1–8.

[7] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, “A novel attack on a FPGA based true random number generator,” in Proc. 10th WESS, Oct. 2015, pp. 1–6.