ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Improved DCM-Based Tunable True Random Number Generator for Xilinx FPGA

Musham Phanindra Kumar
CVR College of Engineering,
Hyderabad, Telangana- 501510, India.

Rajashekar Reddy
CVR College of Engineering,
Hyderabad, Telangana- 501510, India.

Abstract:
True random number generators (TRNGs) play a very important role in modern cryptographic systems. Field-programmable gate arrays (FPGAs) form an ideal platform for hardware implementations of many of these security algorithms. In this thesis, presents 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 tunabilty through dynamic partial reconfiguration to improve randomness qualities. To solve the shortcomings, proposed an improved BFD-TRNG architecture suitable for FPGA based applications .The aim of this study is the design, analysis, and implementation of an easy-to-design, improved, low-overhead, and tunable TRNG for the FPGA platform.

The TRNG utilizes this tunabilty feature for determining the degree of randomness, thus providing a high degree of flexibility for various applications. Reconfigurable devices have become an integral part of many embedded digital systems, reconfigurable systems including FPGAs are being widely employed in cryptographic applications, as they can provide acceptable to high processing rate at much lower cost and faster design cycle time. Hence, many embedded systems in the domain of security require a high quality TRNG implementable on FPGA as a component.

Keywords:
Digital clock manager (DCM), dynamic partial reconfiguration (DPR), field- programmable gate arrays (FPGA), true random number generator (TRNG).

I. INTRODUCTION

TRUE random number generators (TRNGs) have become an indispensable component in many cryptographic systems, including PIN/password generation, authentication protocols, key generation, random padding, and nonce generation [1]. TRNG circuits utilize a nondeterministic random process, usually in the form of electrical noise, as a basic source of randomness. Along with the noise source, a noise harvesting mechanism to extract the noise and a post-processing stage to provide a uniform statistical distribution are other important components of the TRNG. Paper focus is to design improved field-programmable gate array (FPGA) based TRNGs, using purely digital components.

Using digital building blocks for RNGs has the advantage that the designs are relatively simple and well suited to the FPGA design flow [2]. However, digital circuits exhibit comparatively limited number of sources of random noise, e.g., meta stability of circuit elements, frequency of free-running oscillators, and jitters (random phase shifts) in clock signals [3]. As would be evident, Paper proposed TRNG circuit utilizes the frequency difference of two oscillators and oscillator jitter as sources of randomness. Reconfigurable devices have become an integral part of many embedded digital systems, predicted to become the platform of choice for general computing in the near future.

Cite this article as: Musham Phanindra Kumar & Rajashekar Reddy, "An Improved DCM-Based Tunable True Random Number Generator for Xilinx FPGA", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 5, Issue 12, 2018, Page 81-89.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

From being mainly prototyping devices, reconfigurable systems including FPGAs are being widely employed in cryptographic applications, as they can provide acceptable to high processing rate at much lower cost and faster design cycle time [4]. Hence, many embedded systems in the domain of security require a high quality TRNG implementable on FPGA as a component. Paper presents a TRNG for Xilinx-FPGA-based applications, which has a tunable jitter control capability based on dynamic partial reconfiguration (DPR) capabilities available on Xilinx FPGAs [5]. 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 tunabilty 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 [6]. 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. The following are our major contributions.

1) Paper investigates the limitations of the beat frequency detection (BFD)-TRNG when implemented on an FPGA design platform [7]. To solve the shortcomings, propose an improved BFD-TRNG architecture suitable for FPGA based applications. To the best of our knowledge, this is the first reported work which incorporates tunabilty in a fully digital TRNG [8].

2) Paper analyzes the modified proposed architecture mathematically and experimentally.

3) The Experimental results strongly support the mathematical model proposed. The Proposed TRNG has low hardware overhead, and the random bit streams derived from the Proposed TRNG pass all tests in the NIST statistical test suite.

II. DCM based Tunable TRNG

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. Without loss of generality in the rest of this chapter, we will assume that generators produce random bits. In applications where provability is essential, randomness sources (if involved) must also be provably random; otherwise, the whole chain of proofs collapses. In cryptography, where due to Kirchoffs' 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. For example, the well-known BB84 quantum key distribution protocol would be completely insecure if only an eavesdropper could calculate (or predict) either Alice's random numbers or Bob's random numbers or both.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

From analysis of the secret key rate presented therein, it is obvious that any predictability of random numbers by the eavesdropper would leak relevant information to him, thus diminishing the effective key rate. It is intriguing that in the case that the eavesdropper could calculate the numbers exactly, the cryptographic potential of the BB84 protocol would be zero. Indeed one of the recent successful attacks on quantum cryptography exploits the possibility to control local quantum RNGs by exploiting a design flaw of two commercial quantum cryptographic systems and one practical scientific system. This example shows that the local RNGs assumed in BB84 are essential for its security and may not be exempt from the security proof. Lotteries are yet another serious business where random numbers are essential. Due to the large sum of money involved (estimated six billion USD annually only online and only in the USA), some countries have set explicit requirements for RNGs for use in online gambling and lottery machines and have set certificate issuing authorities.

For example, the Lotteries and Gaming Authority (LGA) of Malta has prescribed a list of requirements for RNGs, stipulated in the Remote Gaming Regulations Act. An RNG that does not conform to this act may not be legally used for gambling business. These rules have been put forward in order to ensure fair game by providers and to prevent possibility that gamers manipulate the system by foreseeing outcomes. RNGs have been an occupation of scientists and inventors for along time. Whole branches of mathematics have been invented out of a need to understand random numbers and ways to obtain them. At the dawn of the modern computing era, John von Neumann was one of the first to note that deterministic Turing computers are notable to produce true random numbers, as he put it in his well-known statement that "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." RNGs are one of the hottest topics of research in recent years. There have been about 83 patents per year in the last decade, 1418 in total since 1970, and countless scientific articles published regarding true RNGs. Still, a sharp discrepancy between the number of publications and very modest number of products (only four quantum RNGs and a handful of Zener noise-based mostly phased-out RNGs) that ever made it to the market clearly indicates immaturity of most of the art. In our view, the main problems are lack of randomness proofs and poor reproducibility of the majority of solutions presented so far. The search for true randomness continues.

True Random Number Generators

Due to Kirchoffs' principle, the definition of a RNG suitable for cryptography must include that even if every detail is known about the generator (schematic, algorithms, etc.), it still must produce totally unpredictable bits. In contrast to PRNGs, physical (true, hardware) RNGs extract randomness from physical processes that behave in a fundamentally nondeterministic way which makes them better candidates for true random number generation. A physical RNG is a piece of hardware separate from the computer, usually connected to it via USB or PCI bus. Importing random numbers into a user program is complicated and requires original drivers. Prices range from 1k USD to 30k USD for bit production rates from 4 to 150 megabits per second. Examples of physical processes used to generate randomness include: Johnson's noise, Zener noise, radioactive decay, photon path splitting at the two-way beam splitter, photon arrival times, etc. Unlike the PRNGs, physical random number generators suffer from uneven probabilities of zeros and ones.This section briefly describes the basic BFD-TRNG model and the DPR methodology utilizing DRP ports available in Xilinx CMTs. There are two key techniques used to generate random numbers. One measures some physical phenomenon that is relied upon to be random and after that makes up for conceivable inclinations in the estimation procedure. Alternate uses mathematical algorithms that produce long sequence of clearly random numbers, which are in reality totally determined by an underlying worth, known as a seed.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The previous one is known as True random Number Generator (TRNG). In comparison with PRNGs, TRNGs remove randomness from physical phenomenon and bring it into a PC. One can envision this as a bite the dust associated with a PC. The physical phenomenon can be exceptionally basic, similar to the little varieties in mouse movements or in the measure of time between keystrokes. By and by, in any case, one must be watchful about which source one picks. For instance, it very well may be dubious to utilize key strokes in this design, since keystrokes are regularly supported by the PC's working framework, implying that few keystrokes are gathered before they are sent to the program. To a program sitting tight for the keystrokes, it will appear just as the keys were squeezed all the while, and there may not be a considerable measure of randomness there all things considered. In any case, there are numerous different techniques to get genuine randomness into your PC. A great physical phenomenon to utilize is a radioactive source.

The focuses in time at which a radioactive source rots are totally capricious, and they can without much of a stretch be identified and encouraged into a PC, staying away from any buffering instruments in the working framework. The Hot Bits benefit at Fourmi lab in Switzerland is an amazing case of an random number generator that uses this procedure. Another reasonable physical phenomenon is barometrical commotion, which is very simple to get with an ordinary radio. This is the approach utilized by RANDOM.ORG. You could likewise utilize foundation clamor from an office or disco, however you'll need to keep an eye out for designs. The fan from the PC can add to the clamor, and since the fan is a pivoting gadget, odds are the commotion it produces won't be as random as air commotion. Without a doubt one of the successful methodologies was the lavarand generator, which was worked by Silicon Illustrations and utilized depictions of astrolights to produce genuine random number.

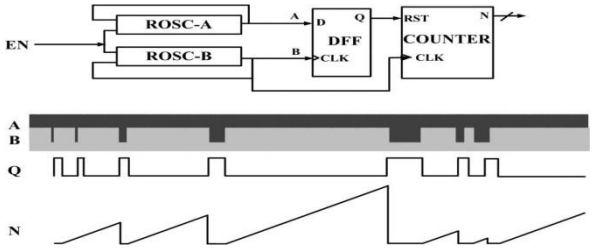

Fig.1: Architecture of the single-phase BFD-TRNG

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. 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.

1) 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.

2) 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.

3) 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4) The output of the counter is sampled by a sampling clock before it reaches its maximum value.
5) The sampled response is then serialized to obtain the random bit stream.

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 maxima. 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. We next describe the proposed tunable BFD-TRNG suitable for FPGA platforms.

Tunable BFD-TRNG

Design Overview

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.

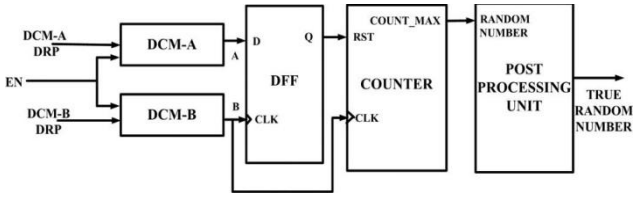

Fig.2: Overall architecture of the proposed DCM-based tunable BFD- TRNG

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 BFD TRNG. 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.

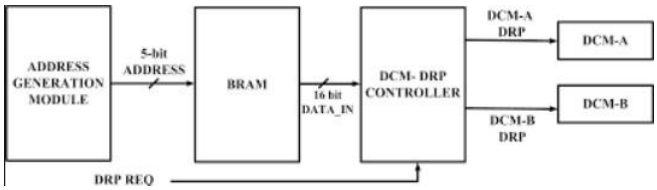

Fig.3: Architecture of Tuning Circuitry of TRNG

Additionally, we have a simple post processing unit using a Von Neumann corrector (VNC) 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Beat Frequency Detector Based High-Speed TRNG

The oscillator sampling method extracts randomness from phase noise in free-running oscillators. An example of this technique is showninFig.4.3, where the output of a fast oscillator is sampled on the rising edge of a slower ring oscillator using a D flip- flop (DFF). Note that the design parameters for the inverters of the two ROSCs are not necessarily the same. The timing fluctuations of the edges of the slow signal relative to the fast oscillator is the source of the randomness in the ROSC based TRNG. Oscillator jitter causes uncertainty in the exact sample values, ideally producing a random bit for each sample.

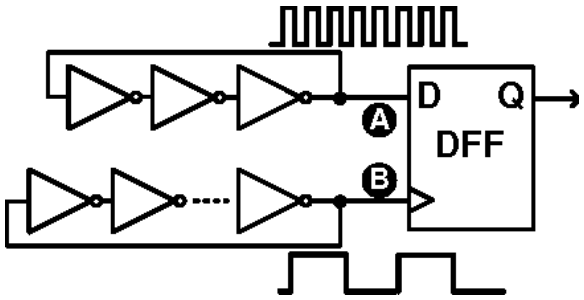

Fig.4: Two Oscillator TRNG

Built on the prior work of ROSC based TRNGs, we have proposed a novel TRNG design to harvest randomness from jitter variation based on the beat frequency detector (BFD) . A beat frequency detector captures the frequency difference between the two ROSCs with a very high resolution, which was originally used to measure frequency degradation of digital circuits. As shown in Fig. 2, the ROSC A is continuously sampled by a ROSC B whose frequency is slightly different from ROSC A. The output of the DFF exhibits the beat frequency Δf, which is determined by the frequency difference of the two ROSCs. A counter measures the beat frequency with ROSC B as the clock. The counter output increments every ROSC period until it reaches the beat frequency interval after which the count is sampled and reset. The output count will fluctuate due to the random jitter in the circuit. The mean of the frequency difference of the two ROSCs is caused by manufacturing process variations, and can be further adjusted by trimming capacitors associated with the ring oscillators. The average frequency tune resolution is 0.1%. The ROSC frequency decreases as we increase the load of each ROSC stage by enabling more MOS capacitors. For example, if we would like to increase the counter values, we can either enable additional capacitors on the fast ROSC or disable capacitors on slow ROSC to achieve the target count range. In our test chip data, the initial count measured from different chips ranges from 200 to 1000 when using the same trimming capacitor setting. Through extensive testing, we found that a count range of 200 to 500 provides a reasonable trade-off between speed and bit efficiency.

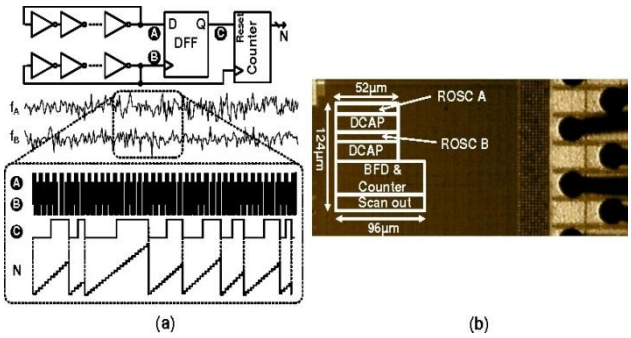

Fig.5: BFD TRNG (a) basic principle,(b) die microphotograph in 65nm.

Random Numbers in Deterministic Cryptography

Therefore it is important to explore what makes contemporary commercial-grade protocols secure and what could be done to get the maximum security out of them. Our hypothesis is that if a protocol requires random numbers, then use of a TRNG maximizes its security. Without ambition to make a strict proof or to give a comprehensive review, here let us have a look at several examples supporting this hypothesis:

III. Simulation and Results

Proposed DCM based TRNG

The section deal with the RTL Schematic, Technology Schematic, Simulation Results of Proposed DCM based Tunable True Random Number Generator as shown in fig 6

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

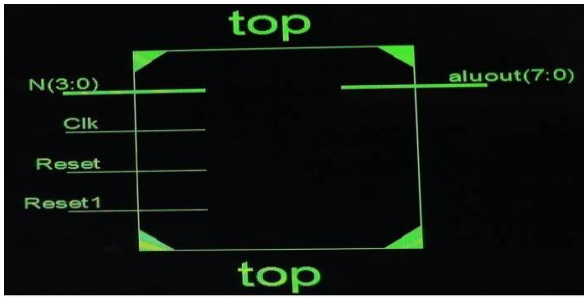

Fig 6: RTL Schematic of DCM based TRNG

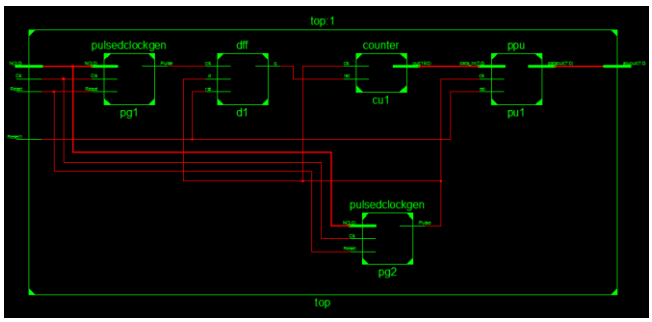

Fig 7: Technology Schematic of DCM based TRNG

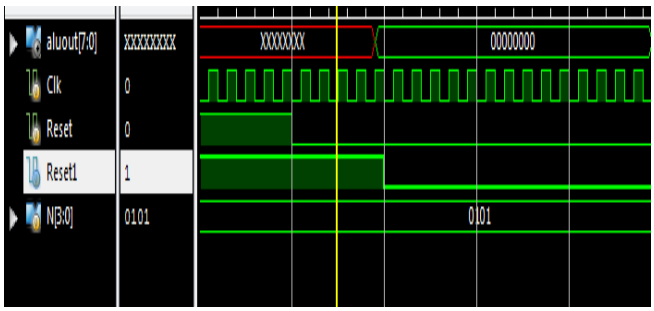

Fig 8: Simulation Results of DCM based TRNG when reset is zero

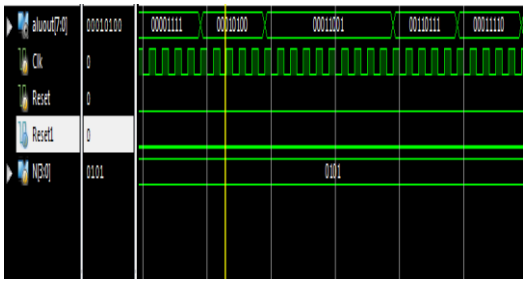

Fig 9: Simulation Results of DCM based TRNG when reset is one

Proposed Pulse Clock Generator

This Section deals with the RTL Schematic, Technology Schematic, and Simulation Results of Proposed Pulse Clock Generator. The RTL Schematic of Pulse clock Generator.

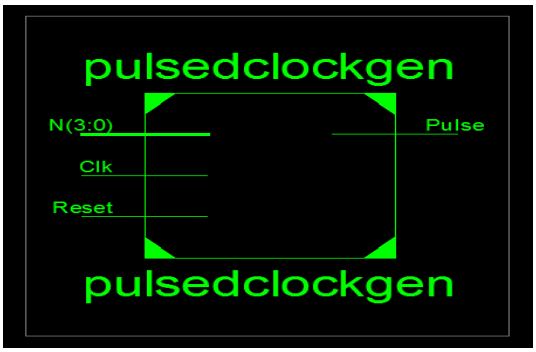

Fig 10: RTL Schematic of Pulse Clock Generator

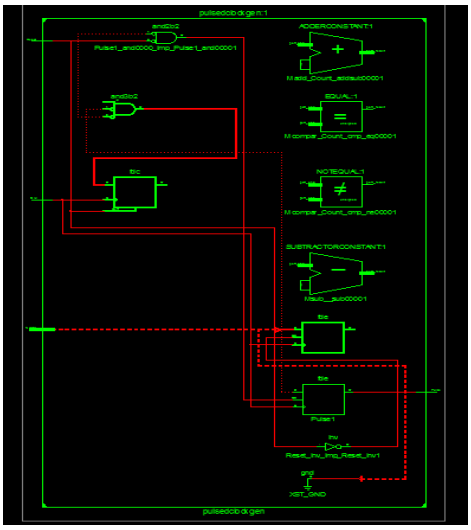

Fig 11: Technology Schematic of Pulse Clock Generator

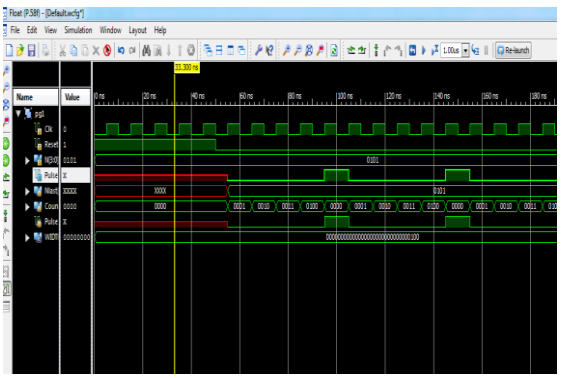

Fig 12: Simulation Results of Pulse Clock Generator when Reset is zero

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

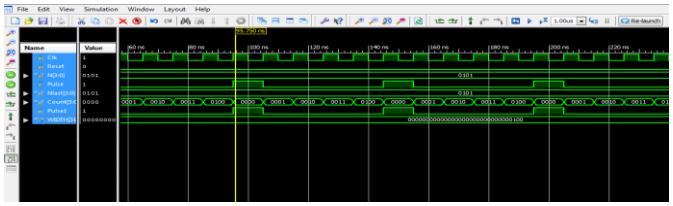

Fig 13: Simulation Results of Pulse Clock Generator when Reset is one

Proposed D-flip flop

This Section deals with the RTL Schematic, Technology Schematic, Simulation Results of Proposed D-flip flop

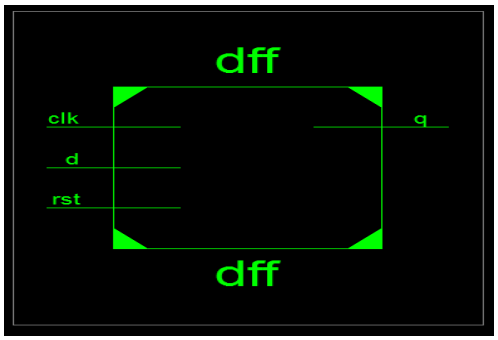

Fig 14: Proposed D-flip flop

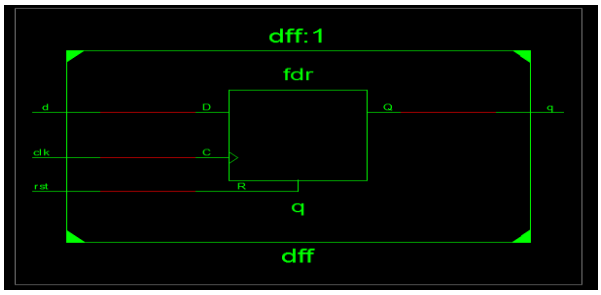

Fig 15: Technology Schematic of Proposed D-flip flop

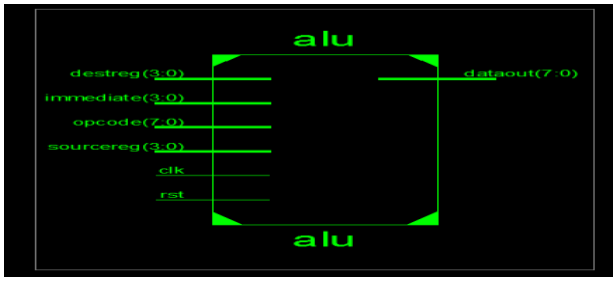

Fig 16: RTL Schematic of ALU block

Simulation Results of Control Unit

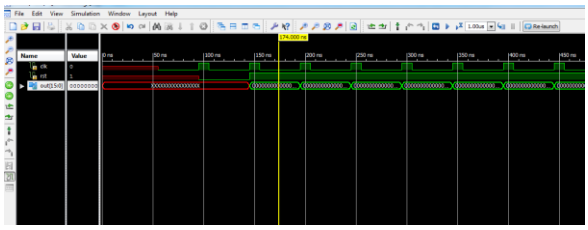

Fig 17: Simulation Results of Control Unit block

IV. CONCLUSION AND FUTURE SCOPE

This Thesis describes 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 Random Number Generator is truly random as the seed value is taken from Jitter source. Hence it completely avoids the possibility of determinism which is risk factor in Pseudo Random Number Generators. It is also much easier to implement as compared to other hardware Random Number Generators which use radioactive decay, atmospheric noise, electrical noise from resistor, etc and require a much complex hardware setup to generate randomness.

The TRNG utilizes this tunability feature for determining the degree of randomness, thus providing a high degree of flexibility for various applications. Random number generator design is an exciting field, every year new requirements and designs are derived. It is quite challenging as the design has to be secure from attacks and should be fault proof. True Random Number Generator is to reduce the overall processing time.16bit range (0-65536) of random numbers can be increased by using a bigger sized compiler and microprocessor, while maintaining the high degree of randomness.

REFERENCES

[1] Virtex-5 FPGA Configuration User Guide UG 191 (v3.11) Xilinx Inc., San Jose,CA,USA, Accessed: May 2016. [Online]. Available: www.xilinx.com/support/documentation/user_guides/ug191.pdf

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[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,"inProc. IEEE Custom Integr. Circuits Conf., Sep. 2014, pp.1–4.

[4] A.Rukhin, J. Soto, J. Nechvatal, M.Smid,and E. Barker,"Astatistical 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 StandardsAppl.Math. Ser.,vol.12, pp. 36–38,1951.

[6] A.J. Jara, M.A.Zamora-Izquierdo, andA.F. Skarmeta, "Interconnection framework for mHealth and remote monitoring based on the Internet of Things," IEEE J.Sel.Areas Commun.,vol.31, no. 9, pp. 47–65, Sep.2013.

[7] S.Kelly,N. Suryadevara, and S. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoringinhomes," IEEE Sensors J.,vol.13, no. 10, pp. 3846–3853, Oct.2013.

[8] S.Wang,"Spatial data mining under Smart Earth,"inProc. IEEE Int.Conf.Granular Comput., Nov. 2011, pp.717–722.