DOI 10.29042/2018-4375-4379

Helix Vol. 8(6): 4375- 4379

True Random Number Generator through Beat Frequency Oscillators in FPGA

Gouri Morankar

Department of Electronics Engineering, Shri Ramdeobaba College of Engineering & Management, Nagpur, Maharashtra, India.

Email: morankarg@rknec.edu

Received: 20th September 2018, Accepted: 11th October 2018, Published: 31st October 2018

Abstract

True random number generator finds applications in cryptography, communications, industrial testing, computer games, random padding and Monte Carlo simulations. Software based random number generators suffers from higher delay due to serial execution of codes, low quality due to correlated data, low throughput and large processing time. FPGA based random number generators are efficient due to parallelism and bit wise operation. In this paper, an attempt has been made to evaluate the performance of the true random number generator based on beat frequency oscillators using FPGA. Asynchronous modulo 'D' counters are explored as beat frequency oscillators. Varying the value of D for both the asynchronous clock divider units along with maintaining higher frequency at oscillator X, adds to the randomness of the generated bit streams. Clock frequency divider 'D' can be selected from the set of values stored in memory. One of the most important advantages of the beat frequency based true random number generator is its tunability through controlling parameter, partial reconfiguration, placement and routing. The technique does not depend on the type and manufacturer of the FPGA, placement and routing in the process of implementation and requires low hardware footprint. Mathematical model, VHDL simulation and experimental results of the true random number generator are discussed. The proposed architecture for TRNG was simulated using VHDL through Xilinx ISE 14.1 software platform and implemented on Virtex 5 XC5VLX20T FPGA device. The random bit stream passes the entire NIST statistical test suite.

Keywords

True Random Number Generator, Beat Frequency Oscillators, FPGA, NIST

Introduction

Tremendous development in the digital technology and Internet of Things (IoT) has fuelled the use of security components such as cryptography, password generation, authentication, ownership, copyright protection and key generation. True random number generator (TRNG) is the essential module in most of the above blocks. TRNG finds applications in cryptography, communications, industrial testing, computer games, random padding and Monte Carlo simulations. TRNG circuits exploits a random process, the source of randomness is mostly through noise, which is extracted, amplified and digitized. Finally a post processing module provides a uniform statistical distribution of the extracted bit streams. Hardware implementations of TRNG are found to be more advantageous with respect to delay, quality, throughput and large processing time as compared to software implementation [1] [2]. Field programmable gate array (FPGA) has emerged as the best platform for the implementation of TRNGs due to parallelism, bit wise operation and low cost. Besides being prototyping devices, FPGAs are also explored for cryptography applications. Hence many embedded cryptography systems requires better quality TRNGs implemented on FPGAs [3]. However, the biggest disadvantage of the FPGA lies in the resource constraint environment that includes digital logic blocks and does not incorporate analog blocks. Hence the only source of randomness lies in the metastability and jitter. It makes the design and implementation of TRNGs more challenging and difficult using FPGAs. It is essential that FPGA implementation of TRNGs should be independent of the technology, manufacturer and implementation process. In [4] [5] a TRNG was proposed through coherent sampling technique that uses an analog phase-locked loop (PLL) to control the clock signal. The proposed TRNG was implemented in an Altera FPGA that includes a PLL. Whereas, the proposed design is not portable with FPGAs available through other manufacturers such as Xilinx. In [6] a TRNG by utilizing dynamic partial reconfiguration (DPR) capabilities of modern FPGAs for varying the digital clock manager (DCM) modeling parameters was proposed. 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 [6]. These facilities are available with Xilinx clock management tiles (CMTs) that contain a dynamic reconfiguration port (DRP) which allows DPR to be performed through much simpler means but not in an Altera FPGAs. In [7] several ways to design TRNG in FPGAs using oscillator rings that consist of an odd number of NOT gates were discussed and its sampling frequency were analyzed. The proposed TRNG in [7] consists of 16 oscillator rings, each containing 3 NOT gates and the sampling frequency of 300MHz was achieved. The circuit using 48 inverters, 17 DFFs, 31 XORs and about 100 routing resources was demonstrated behaving truly random. In [8] LUT based shift register TRNG through first in first out shift register (FIFO

SR), quadratic residue block, XOR gates connection block and parallel in parallel out shift register (PIPO SR) was demonstrated. The blocks together were able to generate random bit streams. It was observed that LUT-FIFO TRNG generates high quality random bit streams but it uses the RAM block from the FPGA resources. Also LUT-SR TRNGs generates random bit streams with reduced utilization of FPGA resource as compared to LUT-FIFO TRNGs. It does not generate good quality random bit streams as compared to LUT-FIFO TRNGs [9][10][11][12]. Variable probability pseudo random number generator (PRNG) using Gollmann Cascade [13] through linear feedback shift register was demonstrated in [14]. The PRNG was able to generate a sequence of N numbers based on variable probabilities. The corresponding probabilities were defined to each sequence in the beginning. The proposed algorithm demonstrated a personalized PRNG based on linear feedback shift register cascade and the resultant random bit streams was verified through NIST. Various techniques and its implementation in FPGAs are being explored by many researchers to achieve high quality random bit streams that satisfies NIST test suite. However it is equally necessary to achieve high throughput, parallelism, bit stream operations and reduced hardware resources to fully utilized the FPGA capabilities.

In this paper, an attempt has been made to evaluate the performance of the true random number generator based on beat frequency oscillators using FPGA. One of the most important advantages of the beat frequency based true random number generator is its tunability through controlling parameters, partial reconfiguration, placement and routing. The technique does not depend on the type and manufacturer of the FPGA, placement and routing in the process of implementation and requires low hardware footprint. Mathematical model, VHDL simulation and experimental results of the true random number generator are discussed. The introduction to TRNGs is illustrated in this section and the remaining paper is organized as, the mathematical model, design and implementation is briefed in section 2, experimental results are discussed in section 3 and finally concluded in section 4.

Model for TRNG

Basic BFDTRNG Architecture

The basic TRNG circuit using beat frequency detection (BFD) mechanism was successfully implemented using 65 nm CMOS technology. The structure of the TRNG BFD circuit was demonstrated in [15] and shown in figure 1. The circuit consists of two identical oscillators X and Y with similar circuits and components. Due to process variations during CMOS process manufacturing, one of the oscillators X oscillates faster as compared to oscillator Y. The output of the oscillator X and Y are connected to 'D' and 'clock' input of the D flip flop respectively. Due to beat frequency interval among the oscillations of oscillator X and Y, output 'Q' of D flip flop is logic '1' at random intervals. The output 'Q' of the D flip flop is used to reset the N-bit counter. Due to random jitter, the beat frequency intervals between oscillator X and Y varies generating random bit streams from the N-bit counter as shown in figure 1.

In case of BFD TRNGs, the quality of the random bit streams completely depends upon the ring oscillators. In FPGA, the design of ring oscillators with equal number of NOT gates but placed and routed differently results in varying count.

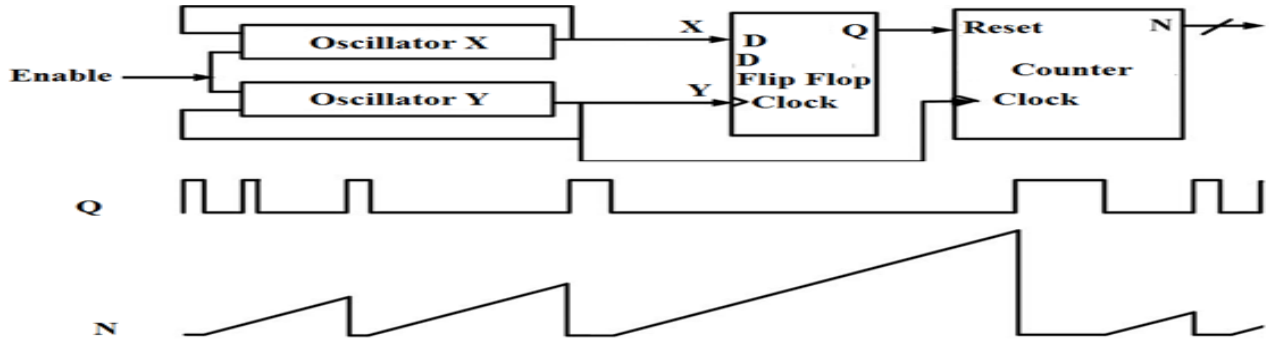

Figure 1: TRNG BFD Architecture [15]

Mathematical Model

In this design, the oscillators X and Y are considered as two modulo D counters and its feasibility as important component in TRNG is explored. The modulo D counter synthesizes clock signal whose frequency is expressed as

$$F_{clkout} = \frac{F_{clkin}}{D} \quad (1)$$

where F_{clkin} is the frequency of input clock signal to the modulo D counter, F_{clkout} is the frequency of the output clock signal and D is the division factor. The two oscillators X & Y can be configured to generate two different clock frequencies through the parameter D. Let us consider that oscillator X oscillates faster than oscillator Y, that clearly indicates that $F_x > F_y$ and $t_x < t_y$ and let N be the number of clock signals from the slower oscillator Y. If the faster oscillator X completes only one additional clock cycle as compared to oscillator Y then

$$t_x\,(N+1) = t_x\,(N+1) + \varphi_x \quad (2)$$

$$t_y\,N = t_y\,N + \varphi_y \quad (3)$$

where φ_x and φ_y are delays caused due to jitter in oscillator X and Y respectively. The delays φ_x and φ_y due to jitter in oscillator X & Y are different since the value of divider D is distinct. Assuming normal operation of the D flip flop, the time (t_c) required to reset the counter is given as

$$t_c = \frac{t_x\,(N+1) + t_y\,N + \varphi_x + \varphi_y}{2} \quad (4)$$

From the equation (4), it is clear that the time (t_c) completely depends upon delays φ_x and φ_y due to jitter and is a random function. Hence the counter counts until it gets reset after time (t_c) is also random. The D flip flop sets to logic '1' after time (t_c) and reset the counter automatically, thereafter counter starts counting again. Thus the outputs of the counter are random bit streams.

Architecture of Proposed TRNG

The architecture of the proposed TRNG through beat frequency oscillator is depicted in figure 2. It consist of two asynchronous modulo D counters designed using T flip flops, D flip flop, one 16- bit synchronous counter, post processing unit and divider 'D' selection mechanism. Asynchronous modulo counter was selected to function as clock divider since it provides facility to reset counter at desired value D as compared with synchronous counter. Synchronous counter needs hard wired circuit to operate at modulo D counter whereas tuning or varying the value of D is difficult and increases hardware. Varying the value of D for both the asynchronous clock divider units along with maintaining higher frequency at oscillator X, adds to the randomness of the generated bit streams. Divider D selection mechanism includes 8 bytes of memory, which can be address using three select lines obtain from the generated random bit streams. The Divider D is a four bit number each for two modulo D counters that makes it total eight bits. The value of D is unequal for the both the counters and can be stored / updated in memory. Post processing unit is a simple circuitry used to improve the statistical properties of the generated bit streams. It removes the recurring bit patterns of "00" or "11" by replacing it with "01"or "10".

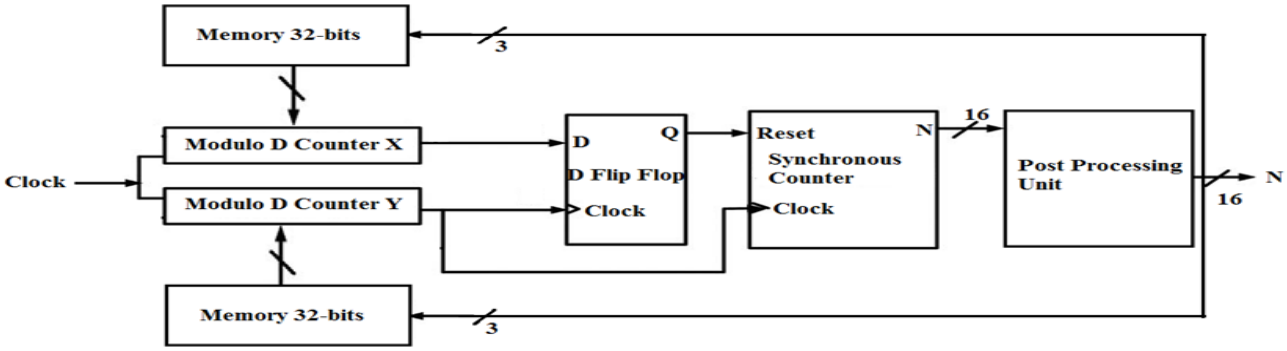

Figure 2: Architecture of the Proposed TRNG

Results and Discussion

The proposed architecture for TRNG was simulated using VHDL through Xilinx ISE 14.1 software platform and implemented on Virtex 5 XC5VLX20T FPGA device [16]. The complete circuit including memory and its controller was implemented using VHDL without using any soft core processor. It reduces the total FPGA resource utilization and requires low hardware footprints. Random bit streams generated after simulation of the proposed TRNG circuit is depicted in figure 3. Table 1 illustrates the total resources utilized by the complete TRNG circuit. Additionally the hardware resource does not vary with the FPGA devices. Whereas power dissipation varies with FPGA devices and change in clock frequencies. Table 2 illustrates the power dissipation incurred by the TRNG at various clock frequencies. The proposed circuit can be controlled through the parameter D which is not possible in conventional BFD TRNGs circuits. Moreover the conventional BFD TRNGs circuits generate different counter maximum values when implemented on various FPGA devices. The 16-bit generated random numbers with mean and entropy of 30437 and 14.9 respectively. The statistical performance of the random bit streams generated from the implemented circuit is illustrated in table 3. Experimental results clearly indicate randomness properties of the TRNG circuit with device independent operation, low hardware and low power dissipation. The comparison of proposed TRNG with recently implemented TRNG using coherent sampling with self-timed rings is illustrated in table 4. It clearly indicates better throughput and low complexities. Hardware complexities are categories as low complexities: easily implemented, medium complexities: uses PLL, DCM, analog blocks and high complexities: manual placement and routing.

Logic Utilization	Used		
	Virtex5	Spartan6	Kintex7
Number of Slice Registers	33	33	33
Number of Slice LUTs	50	50	50
Number of fully used LUT-FF pairs	33	33	33
Number of bonded IOBs	18	18	18

Table 1. Resource Utilization Summary

Frequency (MHz)	Static power (mW)	Dynamic power (mW)	Total power (mW)
50	188.25	62.75	251
100	188.5	65.5	254
200	189.25	70.75	260
500	190.5	88.5	279

Table 2. Power Dissipation

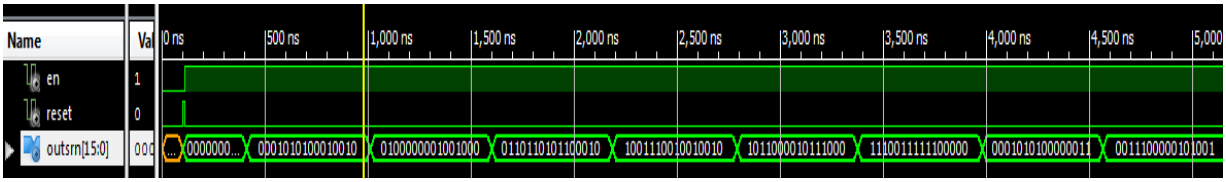

Figure 3: Random Bit Streams Generated Through TRNG

Conclusion

Fully digital true random number generator based on beat frequency oscillators, jitter and asynchronous modulo counter using FPGA is demonstrated. Bit wise operation was explored through asynchronous modulo counter and its divider parameter D that facilitates to select operating clock frequency of the oscillators. It assists in introducing randomness into the circuit and provides partial tunability. Asynchronous modulo counter can be successfully replaced with ring oscillators and PLL to explore device independent operation. Although the use of memory and its controller increases hardware footprint marginally, but can be easily accommodated in a FPGA. It also provides additional tunability and control over the circuit through parameters. Further the functionality of the circuit can be scaled by increasing number of flip flops in modulo counter, which may improve statistical properties of the random numbers. Also the vulnerability of the circuit can be tested for various attacks.

Parameters at 50 MHz	Values
Max count	61345
Min count	505
Normalized Frequency	0.89
Mean	30437
Entropy	14.9
Overlapping Template	0.23
Non overlapping Template	0.77

Table 3. NIST Statistical Parameters Summary

Parameters	[4]	Our Work
LUTs	32	50
Registers	48	33
Entropy	7.99	14.9
Complexities	Medium	Low
Portability	Yes	Yes
Tunability	No	Yes
Throughput	4 Mb/s	5 – 25 Mb/s

Table 4. TRNG Comparison

References

[1] J. L. Danger, S. Guilley, and P. Hoogvorst. High speed true random number generator based on open loop structures in FPGAS. In: Microelectronics Journal, vol. 40, no. 11, 2009, pp. 1650–1656.

[2] S. Saab, J. Hobeika, and I. Ouaiss. A novel pseudorandom noise and band jammer generator using a composite sinusoidal function. In: IEEE Transaction on Signal Processing, vol. 58, no. 2, 2010, pp. 535–543.

[3] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay. A PUF-enabled secure architecture for FPGA-based IoT applications. In: IEEE Trans. Multi-Scale Computing System, vol. 1, no. 2, 2015, pp. 110–12.

[4] Honorio Martin. A New TRNG Based on Coherent Sampling with Self-Timed Rings. In: IEEE Transactions on Industrial Informatics, Vol. 12, No. 1, 2016, pp 91 – 100.

[5] M. Fischer and V. Drutarovsky. True random number generator embedded in reconfigurable hardware. In: Proceedings of International Workshop Cryptography Hardware Embedded System (CHES'02), vol. 2523, 2002, pp. 415–430.

[6] Anju P. Johnson, Rajat Subhra Chakraborty, and Debdeep Mukhopadyay. An Improved DCM-Based Tunable True Random Number Generator for Xilinx FPGA. In: IEEE Transactions On Circuits And Systems—II: Express Briefs, Vol. 64, No. 4, 2017, pp 452 – 456.

[7] Xiufeng Xu, Yuyang Wang. High Speed True Random Number Generator Based on FPGA. In: IEEE International Conference on Information Systems Engineering, 2016.

[8] Remya Justina, Binu K Mathew and Susan Abe. FPGA Implementation of High Quality Random Number Generator using LUT based Shift Registers. In: Elsevier Procedia Technology, vol. 24, 2016, pp 1155 – 1162.

[9] D. B. Thomas and W. Luk. High quality uniform random number generation using LUT optimized state-transition matrices. In: Journal of VLSI Signal Processing, vol. 47, no. 1, 2007, pp. 77–92.

[10] D. B. Thomas and W. Luk. The LUT-SR Family of Uniform Random Number Generators for FPGA Architectures. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 4, 2013, pp. 761 – 770.

[11] K. Wold and C. H. Tan. Analysis and enhancement of random number generator in FPGA based on oscillator rings. In: International Journal of Reconfigurable Computing, vol. 2009, pp. 4:1–4:8.

[12] O. Cret, A. Suciu, and T. Gyorfi. Practical issues in implementing TRNGS in FPGAS based on the ring oscillator sampling method. In: Proceedings of 10th International Symposium on Symbol Number Algorithms Science Computing (SYNASC'08), 2008, pp. 433–438.

[13] Dominik Jochinger and Franz Pichler. A New Pseudo-Random Generator Based on Gollmann Cascades of Baker-Register-Machines. In: International Conference on Computer Aided System Theory, 2005, pp 311 – 316.

[14] Andrei Marghescu, Paul Svasta and Emil Simion. High Speed and Secure Variable Probability Pseudo/True Random Number Generator using FPGA. In: IEEE 21st International Symposium on Design and Technology in Electronic Packaging, 2015.

[15] 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: Proceedings of IEEE Custom Integrated Circuits Conference, 2014, pp. 1–4.

[16] Xilinx. Virtex-5 Family Overview, http://www.xilinx.com, v5.1, 2015.

