Area Efficient DCM Based True Random Number Generators

[1]DR.UDARA YEDUKONDALU, [2] B.VISALI
[1]Professor & HOD, [2]M.Tech scholar
[1]Department of ECE,
[1]Ramachandra College of engineering, Eluru, A.P, 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 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. The proposed TRNG has low hardware footprint and built-in bias elimination capabilities. The random bit streams generated from it pass all tests in the NIST statistical test suite.

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. 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. Our focus is to design improved field-programmable gate array (FPGA) based TRNGs, using purely digital components. Using digital building blocks for TRNGs has the advantage that the designs are relatively simple and well suited to the FPGA design flow, as they can suitably leverage the CAD software tools available for FPGA design. 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. As would be evident, our 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. 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. Hence, many embedded systems in the domain of security require a high quality TRNG implementable on FPGA as a component. We present 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. 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.

A digital clock manager (DCM) is an electronic component available on some FPGAs (notably ones produced by Xilinx). A DCM is useful for manipulating clock signals inside the FPGA, and to avoid clock skew which would introduce errors in the circuit.

Partial reconfiguration(PR) is the ability to reconfigure select areas of an FPGA any time after its initial configuration.

From the functionality of the design, partial reconfiguration can be divided into two groups: dynamic partial reconfiguration (DPR) and static partial reconfiguration. Dynamic partial reconfiguration, which is illustrated in figure 1, also known as active partial reconfiguration, permits to change a part of the device while the rest of an FPGA is still running.

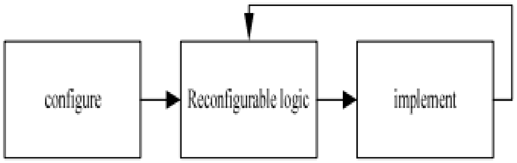


Fig.1.dynamic partial reconfiguration

II EXISTING SYSTEM

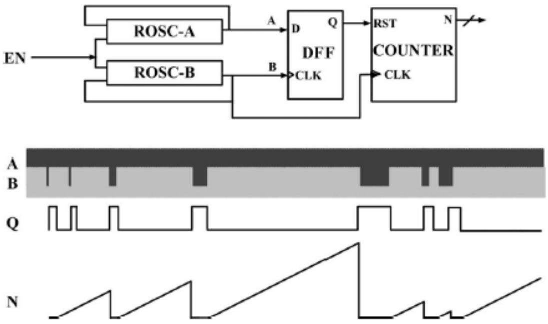


Fig.2. Architecture of the single-phase BFD-TRNG

This section briefly describes the basic BFD-TRNG model and the DPR methodology utilizing DRP ports available in Xilinx CMTs.

A.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.

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

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.

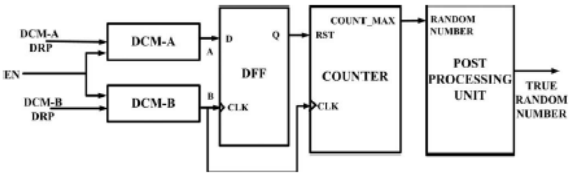


Fig.3. Overall architecture of the proposed DCM-based tunable BFD-TRNG

B.Tuning Circuitry

The architecture of the tuning circuitry is shown in Fig.4. 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 (details are discussed in Section IV). This makes it possible to tune the proposed TRNG using the predetermined stored M andD values. As unrestricted DPR has been shown to be a potential threat to the circuit, 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

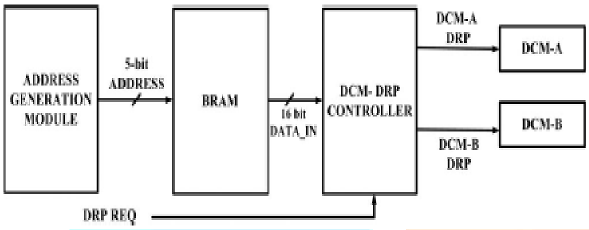
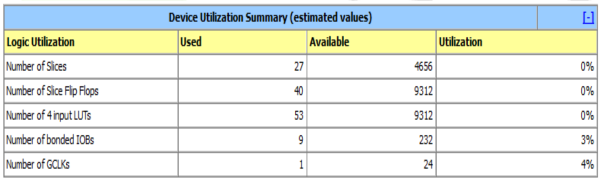


Fig. 4. Architecture of tuning circuitry

IV RESULTS:

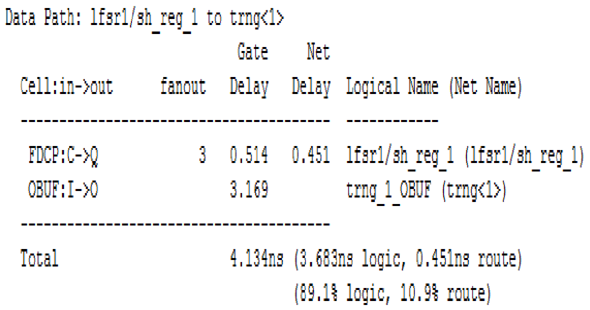
EXTENSION

Design summary:



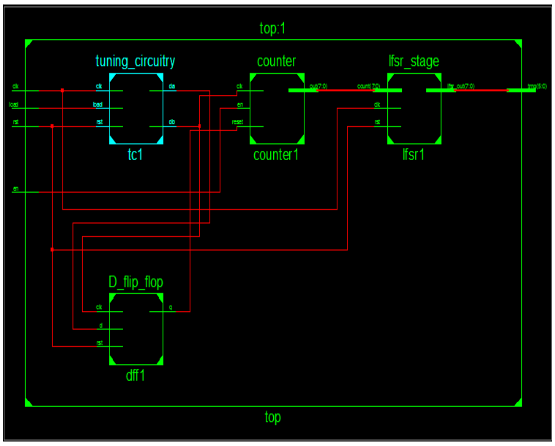
Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	27	4656	0%
Number of Slice Flip Flops	40	9312	0%
Number of 4 input LUTs	53	9312	0%
Number of bonded IOBs	9	232	3%
Number of GCLKs	1	24	4%

Timing report:

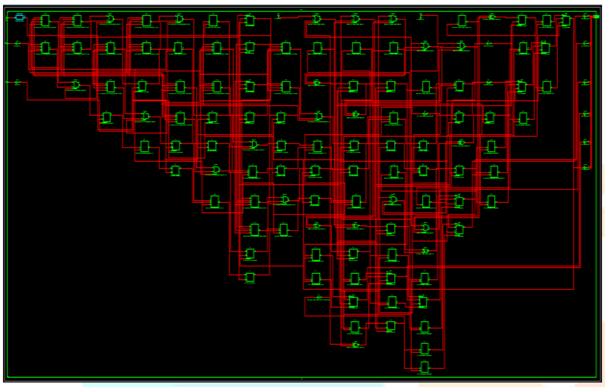


```
Data Path: lfsr1/sh_reg_1 to trng<1>
                              Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   FDCP:C->Q             3   0.514   0.451  lfsr1/sh_reg_1 (lfsr1/sh_reg_1)
   OBUF:I->O                 3.169          trng_1_OBUF (trng<1>)
  ----------------------------------------
  Total                      4.134ns (3.683ns logic, 0.451ns route)
                                     (89.1% logic, 10.9% route)
```

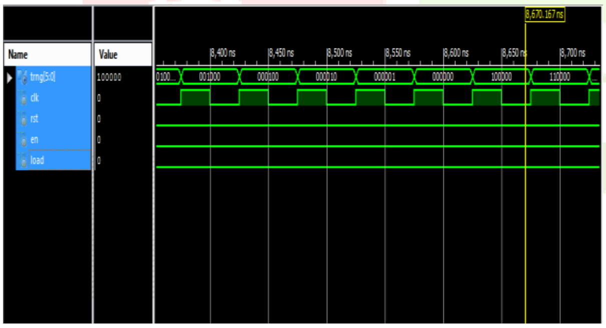
RTL schematic:



Technology schematic:

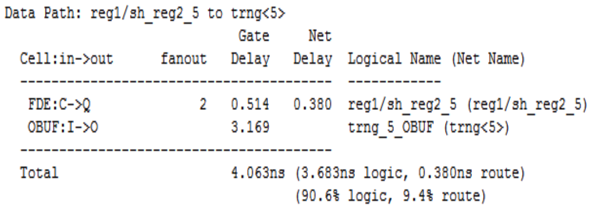


Simulation results:



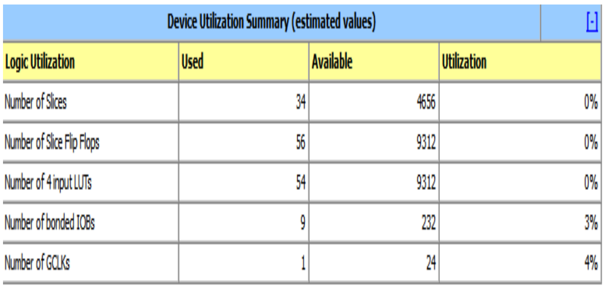
PROPOSED:

Timing report:



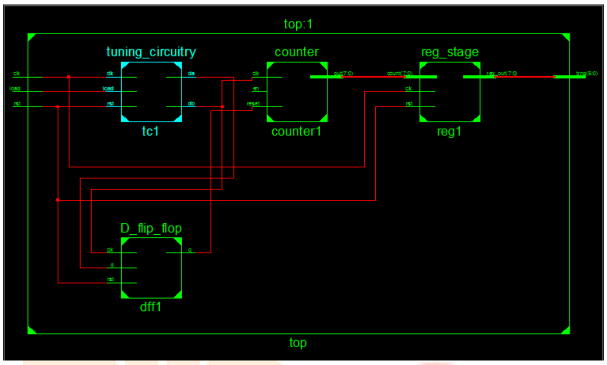
```
Data Path: reg1/sh_reg2_5 to trng<5>
                                Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   FDE:C->Q              2   0.514   0.380  reg1/sh_reg2_5 (reg1/sh_reg2_5)
   OBUF:I->O                 3.169          trng_5_OBUF (trng<5>)
  ----------------------------------------
  Total                      4.063ns (3.683ns logic, 0.380ns route)
                                     (90.6% logic, 9.4% route)
```

Design summary:

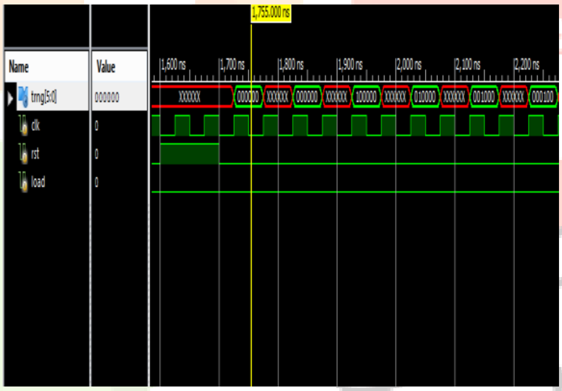


Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	34	4656	0%
Number of Slice Flip Flops	56	9312	0%
Number of 4 input LUTs	54	9312	0%
Number of bonded IOBs	9	232	3%
Number of GCLKs	1	24	4%

RTL schematic:



Simulation results:



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. The proposed design successfully passes all NIST statistical tests.

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

[2] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-enabled secure architecture for ```111FPGA-based IoT applications," IEEE Trans. Multi- Scale 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.