A High-Speed True Random Number Generator Using Metastability with Clock Managers

CH. PRAKASH[1], Y. SRI DANESWARI[2], K. RATNA SRI LAKSHMI[3], K. DURGA SAIRAM[4], G. SRI LAKSHMI[5]

[1, 2, 3, 4] *U.G. Students, Department of ECE, Aditya College of Engineering & Technology, Surampalem, A.P., India*

[5] *Professor, Department of ECE, Aditya College of Engineering & Technology, Surampalem, A.P., India*

Abstract— This paper introduces an innovative method for constructing a True Random Number Generator (TRNG) tailored specifically for Field Programmable Gate Array (FPGA)-based digital systems. TRNGs are pivotal in ensuring the security of various applications, particularly within digital environments. The proposed technique harnesses the dynamic capabilities of Digital Clock Manager (DCM) hardware primitives to regulate the phase shift between clock signals. By exploiting the phenomenon of metastability in flip-flops (FFs), the system effectively generates random numbers of high quality. This auto-tuning mechanism not only simplifies the design process of TRNGs but also fortifies the security of FPGA-based systems by furnishing a reliable source of randomness for cryptographic task.

Index Terms- True Random Number Generator (TRNG), Metastability, Digital Clock Manager (DCM).

I. INTRODUCTION

In the dynamic landscape of cybersecurity, ensuring the integrity and confidentiality of sensitive information stands as a paramount concern. True Random Number Generators (TRNGs) occupy a pivotal role in cryptographic applications, serving as the cornerstone for generating secure cryptographic keys and authentication protocols. With the escalating reliance on digital systems for data transmission and storage, the demand for robust TRNGs has surged in recent years.

Hardware-based TRNGs have garnered significant attention due to their ability to harness physical phenomena such as thermal noise and quantum effects to generate truly random numbers. Unlike pseudo-random number generators (PRNGs) that rely on deterministic algorithms, TRNGs offer an inherent unpredictability that is indispensable for cryptographic applications. This paper aims to delve into the intricacies of TRNG architectures, exploring their diverse implementations across analog and digital platform.

This research paper aims to delve into the intricacies of TRNG architectures, exploring their diverse implementations across analog and digital platforms. From analog circuits exploiting phenomena like metastability and jitter to digital designs utilizing Field Programmable Gate Arrays (FPGAs), TRNGs exhibit a wide range of approaches to randomness generation. However, achieving the desired level of randomness poses significant challenges, often requiring sophisticated post-processing techniques and feedback control mechanisms.

From analog circuits exploiting phenomena like metastability and jitter to digital designs utilizing Field Programmable Gate Arrays (FPGAs), TRNGs exhibit a wide range of approaches to randomness generation. However, achieving the desired level of randomness poses significant challenges, often requiring sophisticated post-processing techniques and feedback control mechanisms. By unraveling the complexities of TRNG design and operation, this paper seeks to contribute to the advancement of secure data transmission and storage in modern cryptographic systems. Through a comprehensive analysis of TRNG architectures and their practical implications, this research endeavors to provide valuable insights into the critical aspects of random number generation, thereby enhancing the security posture of digital systems in an increasingly interconnected world.

II. LITERATURE SURVEY

Physically Unclonable Functions (PUFs) leverage manufacturing process variations to create binary keys (Weak PUFs) or binary functions (Strong PUFs), offering potential advantages in area and power over traditional cryptography. However, many Strong PUF implementations are vulnerable to machine-learning attacks, compromising security. In this paper, we propose using Weightless Neural Networks (WNN) to transform Weak PUFs into robust Strong PUFs, demonstrating resistance to machine-learning attacks while maintaining uniqueness and reliability.

Our approach repurposes neural network hardware, traditionally used for pattern recognition, to enhance security. By employing WNN-based Strong PUFs, we mitigate vulnerabilities to model-building attacks, addressing the limitations of current Strong PUF designs. Our novel architecture offers increased resistance to machine learning while maintaining high uniqueness and reliability.

In 2009, Lin et al. proposed employing DCT coefficient analysis to rapidly, automatically, and accurately identify manipulated JPEG images. A Bayesian approach combined with the DCT coefficient has been devised by the authors to identify spoofed blocks. It is possible to get rid of the faked area by using feature extraction.

Additionally, we introduce a security key generation method for SRAM PUFs based on Fourier analysis, offering an alternative to conventional fuzzy extractors. By analyzing the Fourier spectrums of SRAM devices, we identify random and noise-resistant sign-bits at specific frequency points for key generation. This method eliminates the need for error-correcting codes, making it suitable for resource-constrained systems.

Experimental validation confirms the efficacy of our proposed methods, demonstrating the randomness and noise resistance of sign-bits in Fourier spectrums. Overall, our contributions advance the field of PUFs and offer promising solutions for enhancing security in resource constrained systems.

III. METHODOLOGY

PUFs, exploiting manufacturing variations, offer distinct advantages over conventional cryptography in generating binary keys or functions. However, the vulnerability of many Strong PUFs to machine-learning attacks poses significant security risks. In response, our research proposes a novel approach utilizing Weightless Neural Networks to enhance Weak PUFs into robust Strong PUFs, ensuring resilience against such vulnerabilities while maintaining reliability and uniqueness.

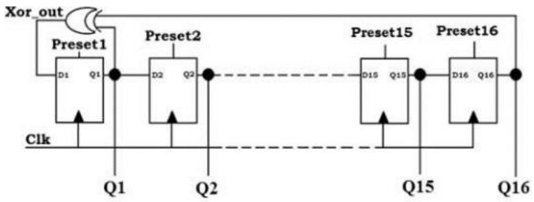


Fig 1. Conventional LFSR

Our methodology involves repurposing neural network hardware, traditionally used for pattern recognition, to fortify security measures against model-building attacks. Furthermore, we introduce an innovative security key generation method for SRAM PUFs based on Fourier analysis.

This alternative to conventional fuzzy extractors eliminates the need for error-correcting codes, rendering it suitable for deployment in resource-constrained system.

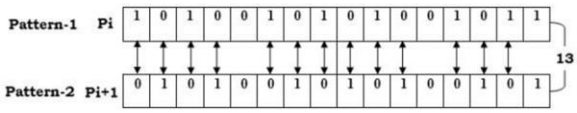


Fig. 2. Switching activity of conventional LFSR

Through rigorous experimentation and validation, we confirm the efficacy of our proposed methodologies, demonstrating the randomness and noise resistance of the generated keys. These contributions not only advance the field of PUF technology but also offer promising solutions for strengthening security in constrained system environments, thus paving the way for more secure and reliable authentication mechanisms in the digital era.

IV. PROPOSED METHODOLOGY

Thus, it does not matter which of them became metastable. The output of the XOR is then sampled by an FF (FFXOR) clocked by the system clock (i.e. ck_in). If the phase difference between clk_in and clk_out is high enough to avoid metastability (due to either the initial DCM state or an unpredictable difference of the signals routing), tshe four FFs of the randomness generator sample the same stable value and the output T of FFXOR is 0. The signal T is the input of a Finite State Machine (FSM) that controls the configuration signals of the DCM in a feedback fashion.

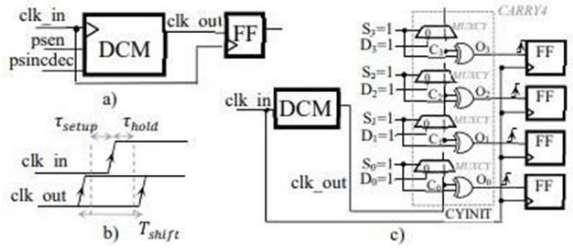


Fig. 3 a) The proposed DCM-based metastability generator; b) timing example; c) the joint proposed use of the CARRY4 and DCM

The clock manager hardware primitive is repurposed for a high-speed TRNG, utilizing dynamic phase shifting (DPS) to induce metastability in FFs. Automatic phase tuning triggers random sequence generation. To augment randomness, the carry-chain primitive is unconventionally used with configurable feedback. A DSP slice facilitates on-chip post-processing without compromising bit production rate
A preliminary analysis, on different placement sites, demonstrated that after the autocalibration which takes 160 clock cycles on average, at least one of the fours FFs actually enters the metastable region. Indeed, over a 10Mb sequence outputted by the XOR gate, the percentage of 0's and 1's is close to 50%. In the proposed scheme, the signal T represents the raw random bit that is generated with a throughput equal to the system clock frequency. With the goal of increasing the randomness of the signal T, a further technique is here adopted in conjunction with the use of the DCM.

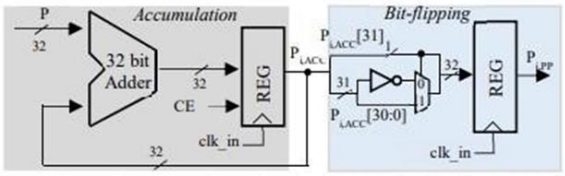


Fig 4: The proposed DSP-based post-processing circuit

As visible in Fig. 4, the signals O[3:0] are in a feedback loop to drive the selectors S[3:0] of the multiplexers of the carry chain. When the signal En is 1, the auto-calibration phase is still running and S[3:0] is set to "1111", as explained above. Once metastability has been ingenerated, En is set to 0 and S[3:0] is set to O[3:0]. Such a selection is performed by four multiplexers, controlled by En, whose logic can be implemented by the four Look-up Tables (LUTs) within the same slice of the CARRY4. The purpose of the proposed scheme is to force the XOR gates of the CARRY4 in a race condition.

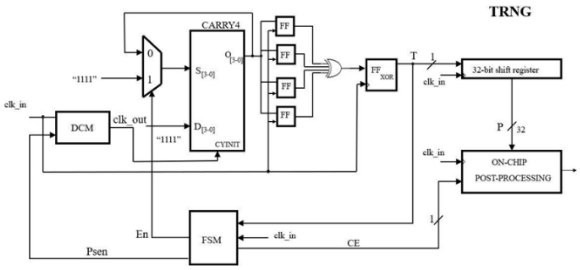


Fig. 5. The proposed TRNG core

V. RESULTS

The simulation results of the True Random Number Generator (TRNG) employing metastability with clock managers demonstrate its efficacy in generating random sequences. By harnessing dynamic phase shifting (DPS) capabilities, the TRNG induces metastability in flipflops, ensuring unpredictability. These simulations validate the automatic phase tuning mechanism, confirming the initiation of random sequence generation upon metastable conditions. Additionally, the utilization of clock managers for metastability offers promising outcomes in terms of randomness and reliability, paving the way for robust TRNG designs in hardware security applications.

Simulation Results:

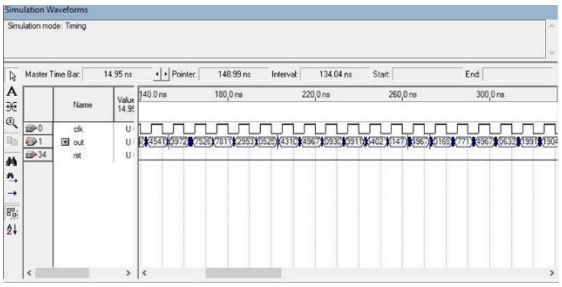


Fig6. Sample test conducted on NI image

Area:

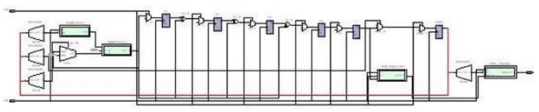
Flow Status	Successful - Wed Feb 08 14:50:47 2023
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	cde
Top-level Entity Name	TOP32
Family	Cyclone II
Device	EP2C35F672C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	172 / 33,216 (< 1 %)
Total combinational functions	139 / 33,216 (< 1 %)
Dedicated logic registers	164 / 33,216 (< 1 %)
Total registers	164
Total pins	34 / 475 (7 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

DELAY:

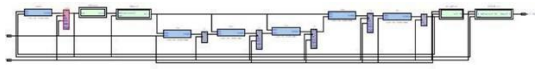
Timing Analyzer Summary

	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1	Worst-case tsu	N/A	None	2.674 ns	rst	shift_reg32:sh1|tmp[4]	--	clk	0
2	Worst-case tco	N/A	None	8.650 ns	POST_PRO:pr1|final_out[19]	out[19]	clk	--	0
3	Worst-case th	N/A	None	-0.165 ns	rst	shift_reg32:sh1|tmp[2]	--	clk	0
4	Clock Setup: 'clk'	N/A	None	44.67 MHz (period = 22.388 ns)	shift_reg32:sh1|out[0]	POST_PRO:pr1|reg_out[31]	clk	clk	0
5	Total number of failed paths								0

RTL Schematic:



Technology Schematic:



CONCLUSION

A new design of a DCM-based TRNG for an easy implementation on FPGA devices has been presented. It exploits the dynamic capability of the DCMs hardware primitives to fine tune the phase difference between two clock signals. The metastability ingenerated by the latter signals is used as a randomness source. The required phase difference is automatically set by a simple FSM. A smart use of the CARRY4 hardware primitive further increases the randomness of the generated bits. Finally, a low-latency on-chip postprocessing scheme is also presented.

REFERENCES

[1] M. Drutarovsky, and P. Galajda, "A Robust Chaos-Based True Random Number Generator Embedded in Reconfigurable Switched-Capacitor Hardware," in Proc. of 17th International Conference Radioelektronika, pp. 1-6, Apr. 2007.

[2] M. Majzoobi, F. Koushanfar, and S. Devadas, “FPGA-based true random number generation using circuit metastability with adaptive feedback control,” in Proc. Crypt. Hard. Embedded Syst. (CHES), 2011, pp. 17–32.

[3] H. Hata, and S. Ichikawa, “FPGA Implementation of Metastability-Based True Random Number Generator,” IEICE Trans. Inf. & Syst., vol.E95-D, no. 2, pp. 426-436, Feb 2012.

[4] R. Della Sala, D. Bellizia and G. Scotti, "A Novel Ultra-Compact FPGACompatible TRNG Architecture Exploiting Latched Ring Oscillators," in IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 69, no. 3, pp. 16721676, March 2022.

[5] N. N. Anandakumar, S. K. Sanadhya, and M. S. Hasmi, “FPGA-based True Random Number Generation Using Programmable Delays in Oscillatorrings,” IEEE Trans. Circuits Syst. II,

Exp. Briefs, vol. 67, no. 3, pp. 570- 574, March 2020.

[6] H. Martin, P. Peris-Lopez, J. E. Tapiador, and E. San Millan, “A New TRNG Based on Coherent Sampling With Self-Timed Rings,” IEEE Trans. on Industrial Informatics, vol. 12, no. 1, pp. 91–100, Feb. 2016.

[7] X. Wang et al., “High-Throughput Portable True Random Number Generator Based on Jitter-Latch Structure,” IEEE Trans. Circuits Syst. I: Regular Papers, vol. 68, no. 2, pp. 741–750, Feb. 2021.

[8] K. Demir, and S. Ergün, “Random Number Generators Based on Irregular Sampling and Fibonacci– Galois Ring Oscillators,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 66, no. 10, pp. 1718–1722, Oct. 2019.

[9] X. Li, P. Stanwicks, G. Provelengios, R. Tessier and D. Holcomb, "Jitterbased Adaptive True Random Number Generation for FPGAs in the Cloud," in Proc. Intern. Conf. on Field-Programmable Technology (ICFPT), 2020, pp. 112-119

[10] D. Schellekens, B. Preneel, and I. Verbauwhede, “FPGA Vendor Agnostic True Random Number Generator,” in Proc. Int. Conf. Field Program. Logic Appl. (FPL), pp 1-6, Aug. 2006.

[11] A. Cherkaoui, V. Fischer, L. Fesquet, A. Aubert, A, “A Very High Speed True Random Number Generator with Entropy Assessment. Cryptographic Hardware and Embedded Systems,” in Proc. Crypt. Hard. Embedded Syst. (CHES), 2013, pp. 1–18.

[12] M. Grujić and I. Verbauwhede, "TROT: A Three-Edge Ring Oscillator Based True Random Number Generator with Time-to-Digital Conversion," IEEE Trans. Circuits Syst. I - Reg. Papers, vol.69, no. 6, pp. 2435-2448, June 2022.

[13] J. Cui et al., "Design of True Random Number Generator Based on MultiStage Feedback Ring Oscillator," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 69, no. 3, pp. 1752- 1756, March 2022.

[14] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadyay, “An improved DCM-based tunable true random number generator for Xilinx FPGA,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 64, no. 4, pp. 452– 456, Apr. 2017.

[15] N. Fujieda, M. Takeda, and S. Ichikawa, “An Analysis of DCM-Based True Random Number Generator,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 6, pp. 1109–1113, Apr. 2020.

[16] Y. Liu, R. C. C. Cheung, and H. Wong, “A bias-bounded digital true random number generator architecture,” IEEE Trans. Circuits Syst. I - Reg. Papers, vol. 64, no. 1, pp. 133–144, Jan. 2017.

[17] L. B. Carreira et al., “Low-latency reconfigurable entropy digital true random number generator with bias detection and correction,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 5, pp. 1562–1575, May 2020.

[18] 7 Series FPGAs Clocking Resources User Guide, UG472, Xilinx, July 2018.

[19] Cyclone IV Device Handbook, Vol. 1, Altera Corporation, March 2016.

[20] L. E. Bassham, III et al., “A statistical test suite for random and pseudorandom number generators for cryptographic applications, Rev. 1a,” U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Rep. SP 800–22, 2010.