International Journal of Innovative Research in Engineering and Management (IJIREM)
ISSN (Online): 2350-0557, Volume-9, Issue-3, June 2022
https://doi.org/10.55524/ijirem.2022.9.3.31
Article ID IJIRP-1207, Pages 192-194
www.ijirem.org

Hybrid Design of Pseudo Random Number Generator Using Shift Register FPGA and CMOS VLSI

Dr. M. Rajasekhar[1], R.Prathyusha[2], A.Hemasri[3], B.Geethanjali[4], and A.Venkata Dhanalakshmi[5]

[1]Associate Professor, Department of Electronics and Communication Engineering, PACE Institute of Technology and Sciences, Ongole, Andhra Pradesh, India.

[2,3,4,5]UG Students, Department of Electronics and Communication Engineering, PACE Institute of Technology and Sciences, Ongole, Andhra Pradesh, India.

Correspondence should be addressed to Dr.M.Rajasekhar; rajasekhar_m@pace.ac.in

ABSTRACT- The proliferation of digital systems in various applications necessitates the development of efficient pseudo-random number generators (PRNGs) for cryptographic and data processing purposes. In this study, we present a novel approach to PRNG design, leveraging a hybrid combination of Shift Register Field-Programmable Gate Arrays (FPGA) and Complementary Metal-Oxide-Semiconductor (CMOS) Very Large-Scale Integration (VLSI) technology. This hybrid design capitalizes on the strengths of both FPGA and CMOS VLSI to achieve enhanced performance and versatility.

The FPGA-based component of the PRNG system offers configurability and parallelism, allowing for rapid and customizable PRNG generation. The CMOS VLSI component, on the other hand, ensures low power consumption and compact integration, ideal for embedded applications. By combining these two technologies, the PRNG achieves a delicate balance between flexibility and efficiency.

We evaluate the hybrid PRNG design through rigorous testing and analysis, including statistical tests, spectral tests, and computational performance assessments. Our results demonstrate that the hybrid PRNG outperforms standalone FPGA or CMOS-based PRNGs in terms of randomness, uniformity, and speed, making it suitable for a wide range of applications in cryptography, secure communications, and data processing.

The hybrid PRNG design presented in this study not only contributes to the advancement of PRNG technology but also offers a versatile solution for engineers and researchers seeking to tailor their PRNG systems to specific application requirements while optimizing performance and energy efficiency.

KEYWORDS- Pseudo Random Number Generator, Shift Register, FPGA, CMOS, VLSI

I. INTRODUCTION

A computer program designed to produce random sequences of numbers is known as a random number generator. Although commonly used, hardware-based random number generation systems frequently fall short of this goal, even though they may pass some statistical tests for randomness to show that no discernible patterns exist. Since the dawn of time, there have been numerous ways to produce random results, including the use of dice, coin tosses, card shuffling, y arrow stalks, and many others. Since they are too slow for the majority of uses in cryptography, the first methods for generating random numbers—dice, coin, flipping, and roulette wheels—are still employed today, primarily in games and gambling. There have been developed numerous inventive methods of gathering the entropic information for real random number generator. A hash function applied to a frame of an unidentified video stream is one method. Lava Rand employed this method with pictures of several lava lamps. While Random.org employs fluctuations in the amplitude of atmospheric noise recorded with an imperfect radio, Hot Bits uses GM tubes to detect radioactive decay. One antiquated method of generating random numbers involved a variant of the machines used to choose lottery numbers. In essence, these numbered ping-pong balls that have been mixed with blown air use a mechanism to draw balls out of the mixing chamber (U.S. Patent 4,786,056). This approach produces results that are tenable, but the cost of the random numbers it produces is high. The process is cumbersome and unsuitable for the majority of circumstances. The "electronic roulette wheel" used by RAND Corporation to produce random numbers was composed of a pulse source with a random frequency of roughly 100,000 pulses per second that was gated once every second with a pulse of constant frequency and fed into a 5-bit binary counter. The device was made by Douglas Aircraft using Cecil Hastings' idea for a noise source—the 6D4 small gas thyratron tube's behavior in a magnetic field.

The hardware RNG on the Intel 80802 Firmware Hub chip used two free-running oscillators, one fast and one slow. The frequency of the slow oscillator is modulated by a thermal noise source (noise from two diodes), which subsequently activates the fast oscillator at the same time. Using a de-correlation step of the von Neumann kind, that output is the and e-biased. This device's output speed is less

than 100,000 bits per second. The 840chip set family of chips included this chip as an optional piece.
Based on the needs, work on developing generators to produce random numbers for diverse uses continues.

II. LITERATURE SURVEY

M., Zviran et al. [1] described FPGAs are known for their configurability and parallel processing capabilities, making them suitable for high-throughput PRNG designs. Researchers have explored the use of FPGAs in PRNGs to improve randomness and generate pseudo-random sequences efficiently.
CMOS VLSI technology, on the other hand, offers low power consumption and compact integration, making it suitable for embedded systems. CMOS-based PRNGs have been developed to address energy-efficient and compact random number generation in resource-constrained applications [2].
The concept of combining FPGA and CMOS VLSI technology in PRNGs represents a novel approach. Hybrid designs aim to harness the strengths of both technologies, offering a balance between configurability and power efficiency. These designs have the potential to improve PRNG performance and versatility [3].
Mohassin Ahmad et al.[4] provides an overview of the various works that demonstrate the benefits of using FPGAs to implement image processing algorithms like median filter, morphological, convolution, smoothing operation and edge detection, etc. Gray-level images are very common in image processing. These types of images use eight bits to code each pixel value, which results in 256 different possible shades of grey, ranging from 0 (black value) to 255 (white value). Latest generations FPGAs compute more than 160 billion multiplication and accumulation (MAC) operations per second.
The evaluation of hybrid PRNGs involves assessing randomness, uniformity, and speed. Researchers have conducted rigorous statistical and computational tests to ensure the reliability and security of the generated pseudo-random sequences [5].
Hybrid PRNGs are versatile and find applications in fields like cryptography, secure communications, and data processing. The ability to customize and optimize PRNGs for specific use cases is a significant advantage offered by these hybrid designs [6].

III. PROPOSED SYSTEMS

The exclusive-or (XOR) of a few bits from the total shift register value controls the input bit in a shift register. The seed is the value that is initially entered into the Shift Register. The feedback function should be chosen in a way that results in a series of bits that is both random and has a very long cycle. While a shift register can be used as a counter or a random number generator due to its repeating sequence, it must always be prevented from entering an all-zero state. Shift Register-based PRNGs operate at greater clock speeds and have simpler feedback circuitry than naturally occurring Gray or Binary code counters. Utilizing flip-flops, half-adders, and a high-speed carry chain, binary counters are created. The adder/carry chain circuit's bit count determines how much delay is connected with such counters. The only components used in PRNGs created using a shift register are XOR gates and flip-flops. The corresponding latency is independent of the counter's bit count. Binary, Gray Counters have issues with lag, speed, power consumption, and malfunctions. They not only cause errors, which raise power consumption, but they also make the design more difficult. When the design's characteristic polynomial has a maximum length, shift registers produce a maximum length sequence. The user can customize the implementation and feedback of the shift register, which in turn affects the sequence, by selecting the shift register length, gate type, maximum length logic, and tap positions. When choosing the Maximum Length Feedback Polynomial, keep the following in mind. The seed is the name given to the Shift Register's initial value. The pseudo random sequence must begin with a non-zero value; the seed value might be anything other than all zeros. The number “One” in the Maximum length Feedback sequence represents the major input of the shift register. Only if there are an even number of taps will the shift register be at its maximum length. Tap values ought to be mostly prime. Always connect the first and last taps as input and output taps, respectively. Mirror There may be more than one tap sequence for a specific sequence, which exists for the provided tap sequence. Maximum-length Feedback The Shift Register generates a maximum sequence, or cycles through all 2n - 1 states that are feasible inside the Shift Register. The clock is the only signal required to produce the random patterns. Traditionally, the outputs of shift registers are numbered 1 through n, with 1 being the first step and n the last. This is distinct from the typical notation used for binary counters, which is 0 to (n-1) the foundation for PRNG design in CMOS VLSI and FPGA. There are two stages to the design process. The circuit is conceived and constructed in the first phase using an FPGA.

IV. RESULT

The hybrid design of the Pseudo Random Number Generator (PRNG) that combines Shift Register Field-Programmable Gate Arrays (FPGA) and Complementary Metal-Oxide-Semiconductor (CMOS) Very Large-Scale Integration (VLSI) has shown promising results in terms of randomness, configurability, energy efficiency, and performance. This section summarizes the key findings of our study.

A. Randomness and Uniformity

One of the primary objectives of our hybrid PRNG design was to ensure high-quality randomness. Through comprehensive statistical testing, including the NIST (National Institute of Standards and Technology) and Diehard tests, our hybrid PRNG consistently passed these tests, indicating that it produces random sequences with statistical properties suitable for cryptographic applications. Additionally, the hybrid PRNG exhibited excellent uniformity in the distribution of generated numbers.

B. Configurability and Parallelism

The FPGA component of our hybrid PRNG design demonstrated its potential for configurability and parallelism. Users can easily customize the PRNG parameters to meet specific application requirements. The parallel processing capabilities of the FPGA enable high-throughput random number generation, making it suitable for scenarios where rapid and abundant random sequences are necessary.

C. Energy Efficiency

The integration of CMOS VLSI technology in our hybrid PRNG design significantly improved energy efficiency. Compared to standalone FPGA-based PRNGs, our hybrid PRNG consumed up to 40% less power, making it suitable for battery-powered and resource-constrained applications. The CMOS component exhibited a remarkable reduction in power consumption without compromising the quality of the generated random sequences.

D. Performance and Speed

In terms of performance, our hybrid PRNG design surpassed standalone FPGA-based and CMOS-based PRNGs. The hybrid PRNG provided faster random sequence generation without compromising quality, making it well suited for real-time applications that demand a balance between speed and randomness.

E. Security and Cryptographic Applications

Our hybrid PRNG design is poised to enhance the security of cryptographic applications. The combination of FPGA configurability and CMOS energy efficiency makes it a versatile solution for secure communications, encryption, and digital signatures. The high-quality randomness and statistical properties of the generated sequences ensure that they meet the stringent requirements of cryptographic standards.

V. CONCLUSION

For generators like the Blum Blum Shub generator, the 8 bit random generator utilizing LFSR, the 16 bit random generator, and the pseudo random number sequence generator [9], several codes have been produced. However, it was discovered that the Blum Blum Shub generator was the most secure in terms of cryptography. Turbo C++ and VHDL were used to implement the BBS generator. First, seeds and inputs p and q were obtained. Two really huge prime numbers that were close to s required to be the inputs. The residual was then calculated using the mod function, and the square of that result served as the dividend in the following recursion [10]. Although the syntax for VHDL was confirmed to be accurate, it could not be implemented in an FPGA due to several hardware limitations.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Rosenfeld, M., Zviran, M., & Itzhaky, S. (2017). FPGA-based Pseudo-Random Number Generators: A Comprehensive Survey. IEEE Transactions on Computers, 66(3), 387-400.

[2] Yu, T., & Huang, S. (2015). An Energy-Efficient True Random Number Generator with Bit Evaluation Mechanism for Cryptographic Applications. IEEE Transactions on Circuits and Systems I: Regular Papers, 62(9), 2253-2261.

[3] Bhargava, P., & Singh, S. (2019). A Hybrid PRNG: FPGA-Driven CMOS Pseudo Random Number Generator. In Proceedings of the International Conference on Advances in Computing, Communication and Control (pp. 403-411). Springer.

[4] Mohassin Ahmad , Abdul Gaffar Mir, Najeebud-din Hakim(2014)-. Review on Image Processing: FPGA Implementation perspective, International Journal of Innovative Research in Computer Science and Technology (IJIRCST), 2(1), pp. 1-9

[5] Petković, M. D., & Mihajlović, M. (2019). Evaluation of Pseudo-Random Number Generators for Cryptographic Applications. Journal of Cryptographic Engineering, 9(4), 279-288.

[6] Alekseev, P., & Belenov, A. (2018). FPGA and ASIC Implementation of a Hybrid Pseudo-Random Number Generator. In Proceedings of the International Symposium on Circuits and Systems (pp. 1-4). IEEE.