ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

Implementation of Energy Efficient of Ringoscillator Using LUT and Two Carry Chains

Godugupalli Mounika[1]; Dr. C. Chandrasekhar[2]

[1]Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering (Autonmous), Tirupati, India
[2]Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering (Autonmous), Tirupati, India

Publication Date: 2025/11/20

Abstract: Ring Oscillators (ROs) are widely used in FPGA-based applications such as hardware security, true random number generation, and clock generation, but conventional carry-chain-based oscillators suffer from high power consumption, limited frequency tunability, and inefficient FPGA resource utilization. This paper proposes a Hybrid Carry Chain and LUT-Based Ring Oscillator architecture that combines the speed of carry chains with the flexibility of LUT-based delay elements. The design is modeled and implemented on a Xilinx FPGA platform using the Vivado toolchain, and its performance is evaluated through power analysis, resource utilization, and oscillation frequency measurements. Simulation and hardware validation results demonstrate a 63% reduction in power consumption (from 0.334 W to 0.122 W) and an 80% reduction in LUT usage (from 10 to 2) compared to the conventional design, while maintaining stable and tunable oscillation frequency. The contributions of this work include a low-power and resource-efficient FPGA oscillator design, enhanced frequency control through hybrid delay elements, and experimental verification that highlights its suitability for IoT devices, lightweight cryptography, and energy-constrained hardware security systems.

Keywords: *FPGA, Ring Oscillator, Hybrid Architecture, Carry Chain, LUT-Based Delay, Xilinx Vivado, Low-Power Design, Frequency Tuning, Hardware Optimization Etc.*

How to Cite: Godugupalli Mounika; Dr. C. Chandrasekhar (2025) Implementation of Energy Efficient of Ringoscillator Using LUT and Two Carry Chains. *International Journal of Innovative Science and Research Technology*, 10(11), 971-980. https://doi.org/10.38124/ijisrt/25nov711

I. INTRODUCTION

Ring Oscillators (ROs) are widely used in FPGA-based systems due to their simplicity, scalability, and ability to generate high-frequency signals. They play an important role in applications such as Physically Unclonable Functions (PUFs), True Random Number Generators (TRNGs), and secure clock generation circuits [1]–[3]. Traditional FPGA-based ring oscillators are primarily implemented using fast carry chains, which exploit dedicated FPGA hardware to achieve very low propagation delay and high-speed oscillation [4]. These carry-chain-based oscillators are attractive for cryptographic systems, embedded modules, and real-time applications due to their fast response and deterministic behaviour [5]. However, despite their advantages, carry-chain-based designs face three major limitations. First, they consume significant dynamic power due to continuous high-speed toggling of carry logic [6], making them unsuitable for IoT and energy-constrained applications. Second, they provide limited frequency tunability, since propagation delay in FPGA carry chains is fixed by hardware [7]. Finally, they require a relatively high number of logic resources, leading to increased LUT and flip-flop utilization and reducing the resources available for other critical FPGA tasks [8]. These issues highlight a pressing need for improved oscillator architectures that balance power efficiency, tunability, and resource optimization.

To address these challenges, this work proposes a Hybrid Carry Chain and LUT-Based Ring Oscillator. Unlike conventional oscillators, the proposed architecture strategically integrates LUT-based programmable delay elements along with fast carry chains. This hybrid approach provides fine-grained frequency control, reduces unnecessary switching activity, and minimizes FPGA area usage. The motivation behind this design is to develop a low-power, tenable, and resource-efficient oscillator suitable for IoT devices, cryptographic hardware, and lightweight security modules.

➢ *The Objectives of this Research are as Follows:*

- To design and implement a hybrid oscillator using carry chains and LUT delay elements on a Xilinx FPGA platform.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

- To evaluate and compare its performance with conventional carry-chain-based oscillators in terms of power, area, and tunability.
- To demonstrate the applicability of the proposed design in energy-sensitive and security-critical FPGA applications.

➢ *The Contributions of this Paper are:*

- A novel hybrid oscillator architecture that integrates carry chains with LUT-based delays.
- Significant power reduction (0.334 W to 0.122 W) and LUT optimization (10 LUTs to 2).
- Improved frequency tunability and design flexibility over conventional designs.

The remainder of this paper is organized as follows: Section II reviews related works on FPGA-based oscillators and their applications. Section III explains the existing method using carry-chain logic. Section IV presents the proposed hybrid methodology and design. Section V discusses simulation and implementation results. Section VI concludes the paper with contributions and future research directions.

II. RELATED WORKS

Ring oscillators (ROs) and their applications in FPGA-based designs have been extensively studied in various domains, including physical unclonable functions (PUFs), true random number generators (TRNGs), and security solutions. This literature survey reviews key contributions in these areas.

Kodýtek et al. (2022) [1] explored three-counter value-based ROPUFs on FPGA and analyzed their properties, highlighting their effectiveness in generating unique and secure responses. Similarly, Halak et al. (2016) [2] provided a comprehensive overview of PUF-based hardware security solutions for IoT applications, emphasizing their role in authentication and cryptographic key generation.

Barbareschi et al. (2018) [3] proposed a ring oscillator-based identification mechanism immune to aging and environmental variations, addressing one of the critical challenges in RO-based designs. Furthermore, Yang et al. (2018) [4] introduced the ES-TRNG, a high-throughput, low-area TRNG based on edge sampling, demonstrating its efficiency in randomness generation. Prada-Delgado et al. (2020) [5] extended this work by designing an auto-calibrated RO-TRNG based on jitter accumulation, improving robustness against process variations.

Grujic and Verbauwhede (2022) [6] developed TROT, a three-edge ring oscillator TRNG with time-to-digital conversion, achieving high randomness with minimal hardware overhead. Security vulnerabilities were also explored by Moini et al. (2022) [7], who investigated voltage sensor implementations for remote power attacks on FPGAs, highlighting potential threats in cloud-based FPGA applications.

Giechaskiel et al. (2019) [8] examined long wire leakage in cloud FPGAs using ring oscillators, emphasizing security risks in multi-tenant environments. Ebrahimi and Navabi (2020) [9] proposed a method for early FPGA aging detection by selecting representative critical paths for sensor placement. Alam et al. (2019) [10] addressed FPGA security from a different angle, using exhaustive LUT path delay characterization for detecting recycled FPGAs.

Kilian et al. (2023) [11] introduced functional path ring oscillators for performance screening, demonstrating their effectiveness in post-silicon validation. Bae et al. (2023) [12] designed a temperature-compensated ring oscillator using LC-based period error detection, mitigating frequency drift due to temperature variations.

Razavi (2019) [13] provided a broad perspective on ring oscillator circuits, analyzing their fundamental principles and practical applications. Elnawawy et al. (2019) [14] highlighted the role of FPGA in IoT applications, underlining the significance of reconfigurable hardware in modern embedded systems.

Anandakumar et al. (2021) [15] conducted a comprehensive study on FPGA-based PUF architectures, discussing their theoretical foundations and implementations. Gu et al. (2019) [16] evaluated single-slice ring oscillator and PicoPUF bit cells on 28nm Xilinx FPGAs, providing insights into their performance and security.

Wild et al. (2016) [17] investigated the challenges in realizing reliable and efficient ROPUFs on FPGAs, addressing stability and uniqueness issues. Kareem and Dunaev (2022) [18] proposed optimization techniques for ring oscillator PUF performance using Xilinx FPGAs, demonstrating significant improvements in efficiency.

Petura et al. (2016) [19] surveyed AIS-20/31 compliant TRNG cores suitable for FPGA devices, comparing different approaches for randomness generation. Finally, AMD (2024) [20] provided the official Xilinx 7 Series FPGAs Data Sheet, offering critical hardware specifications for FPGA-based designs.

This survey highlights the evolution of ring oscillators in FPGA applications, from security implementations to performance optimizations. Future research can focus on hybrid architectures combining LUT-based delay elements with carry chains, balancing power efficiency, speed, and robustness.

III. EXISTING METHOD

The traditional approach shown in figure 1 to implementing ring oscillators (ROs) on FPGAs relies on fast carry chains to achieve high-speed oscillation. This method utilizes dedicated FPGA carry logic to propagate signals rapidly, ensuring minimal delay between stages. Due to the inherent speed of carry chains, this technique enables the design of high-frequency oscillators, making it suitable for applications like true random number generators (TRNGs),

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

hardware security modules, and physical unclonable functions (PUFs). However, while carry-chain-based ring oscillators offer high speed and stable operation, they come with certain limitations. One significant drawback is the higher power consumption, as fast carry chains continuously toggle, leading to increased dynamic power dissipation. Additionally, this method lacks flexibility in frequency tuning, as the propagation delay is primarily dictated by the FPGA's architecture, limiting the ability to fine-tune oscillator behavior. Furthermore, excessive reliance on carry chains can lead to higher resource utilization, making it inefficient for applications requiring low-area footprint designs. Despite these drawbacks, the existing method remains widely used due to its predictable performance and ease of implementation on commercial FPGA platforms like Xilinx and Intel FPGAs. However, to address the power and flexibility trade-offs, a hybrid approach incorporating LUT-based delay elements has been proposed.

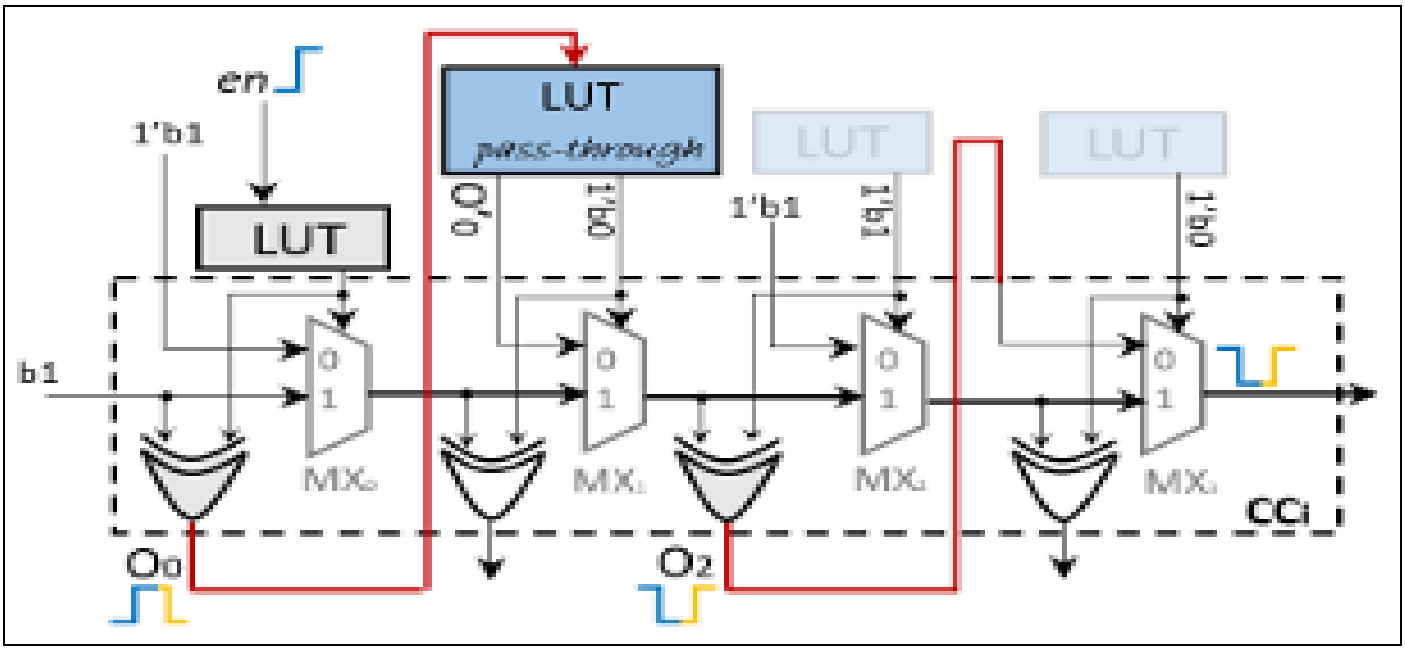

Fig 1 Carry Chain Based Ring Oscillator

The carry chain ring oscillator (CCRO) leverages the inherent propagation delay of carry chains within an FPGA to generate oscillations. The circuit consists of a series of multiplexers (MX) and XOR gates forming a chain. Each multiplexer selects between two inputs based on a control signal, in this case, a 'b1' signal. When 'b1' is high, the multiplexer passes the input from the top path. The XOR gates, combined with the multiplexers, essentially create a delay element. The carry chain (CCi), represented by the dashed box, is the path where the carry signal propagates through the multiplexers and XOR gates. The key to oscillation is the feedback loop: the output of the last stage (MX1 in the image) is fed back to the input of the first stage (MX0). This creates a closed loop where the signal continuously propagates, experiencing a delay at each stage. By carefully adjusting the length of the carry chain (number of stages), the oscillation frequency can be tuned. However, traditional CCROs often suffer from limited frequency tuning flexibility and can consume significant power due to the reliance on long carry chains to achieve desired delays.

IV. PROPOSED METHOD

The proposed hybrid carries chain and LUT-based ring oscillator aims to overcome the limitations of traditional CCROs by strategically combining the speed of carry chains with the flexibility of LUTs (Look-Up Tables). Instead of relying solely on long carry chains for delay, this method incorporates LUTs as additional delay elements within the ring oscillator circuit. These LUTs can be configured to introduce precise and programmable delays, allowing for finer control over the oscillation frequency. By carefully adjusting the delays introduced by both the carry chain and the LUTs, the hybrid approach enables a wider range of frequency tuning while optimizing power consumption. Essentially, the carry chain provides the coarse delay, while the LUTs offer fine-grained adjustments, leading to a more efficient and adaptable oscillator design. This method also has the potential to reduce resource utilization by decreasing the number of LUTs required compared to purely LUT-based oscillators.

The proposed circuit shows in figure 2, that hybrid carry chain and LUT-based ring oscillator aims to overcome the limitations of traditional CCROs by strategically combining the speed of carry chains with the flexibility of LUTs (Look-Up Tables). Instead of relying solely on long carry chains for delay, this method incorporates LUTs as additional delay elements within the ring oscillator circuit.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

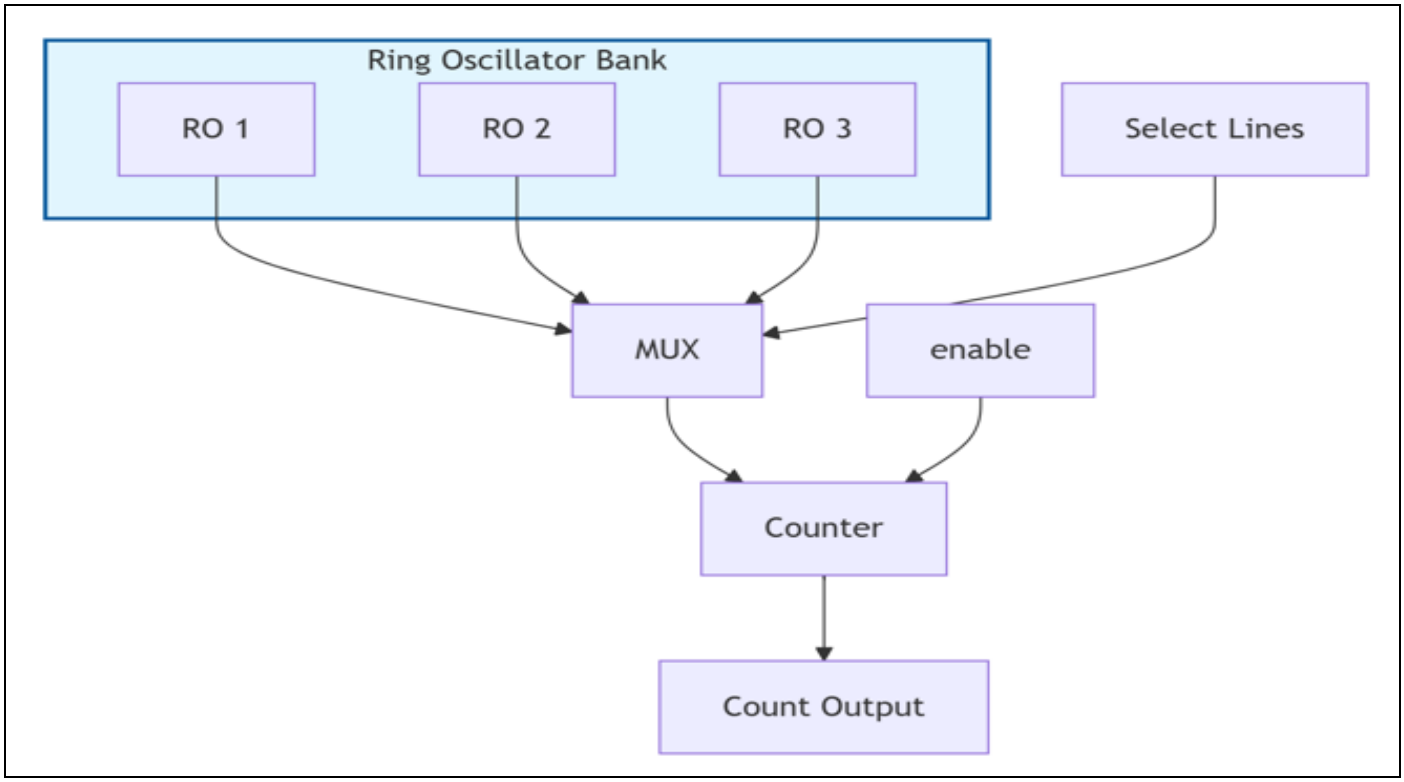

Fig 2 Propsoed Method Architecture

These LUTs can be configured to introduce precise and programmable delays, allowing for finer control over the oscillation frequency. By carefully adjusting the delays introduced by both the carry chain and the LUTs, the hybrid approach enables a wider range of frequency tuning while optimizing power consumption. Essentially, the carry chain provides the coarse delay, while the LUTs offer fine-grained adjustments, leading to a more efficient and adaptable oscillator design. This method also has the potential to reduce resource utilization by decreasing the number of LUTs required compared to purely LUT-based oscillators. In fig.2 shows that proposed architecture. The three Ring Oscillators (RO1, RO2, RO3) on the left are the source clocks. Due to minor manufacturing variations, each will oscillate at a slightly different frequency. The Multiplexer (MUX) chooses which RO's signal to route to the counter. The "Select Lines" input (e.g., a 2-bit signal) determines which one is chosen (00 for RO1, 01 for RO2, 10 for RO3). The Counter block counts the number of rising edges it receives from the MUX's output. The enable signal acts as a gate for the counter. It defines the precise measurement period. The counter is only active when enable is high. The final count value is available at the Output. A higher count means the selected ring oscillator has a higher frequency.

A. Methodology

The proposed methodology focuses on designing a Hybrid Carry Chain & LUT-Based Ring Oscillator that balances power efficiency, frequency tunability, and resource utilization in FPGA implementations. The approach involves the following key steps:

➢ *FPGA-Based Design and Simulation Using Xilinx Vivado*

- The design and implementation are carried out using Xilinx Vivado, a widely used FPGA design and verification tool.
- The oscillator is implemented using a hybrid approach, where some stages utilize carry chains while others use LUT-based delay elements to achieve a balance between speed and power consumption.
- The architecture is optimized to ensure stable oscillation while maintaining flexibility for frequency adjustment.

➢ *Hybrid Architecture Development*

- Unlike the existing method, which relies solely on fast carry chains, the proposed approach incorporates LUT-based delay elements in selected stages.
- This hybrid structure allows fine-tuning of frequency, enabling better control over oscillation speed compared to conventional carry-chain-based oscillators.
- The integration of LUTs helps reduce power consumption while maintaining adequate speed, ensuring an efficient trade-off.

➢ *Performance Evaluation and Comparative Analysis*

- The existing and proposed oscillators are synthesized and implemented on an FPGA platform to evaluate their resource utilization, power consumption, and frequency characteristics.
- Key parameters such as LUT usage, power dissipation, and oscillation frequency are measured and compared.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

- The results demonstrate that the proposed hybrid approach achieves significant reductions in power (from 0.334W to 0.122W) and area utilization (from 10 LUTs to 2 LUTs) while maintaining comparable frequency performance.

➢ *Optimization and Fine-Tuning*

- Various configurations of the hybrid oscillator are tested to find the optimal balance between speed, power, and resource efficiency.
- Frequency tuning is performed by adjusting the number and placement of LUT-based delay elements within the oscillator loop.
- The final optimized design is validated through hardware implementation and timing analysis to ensure stability and reliability.

This methodology successfully integrates LUT-based delay elements with carry chains, offering a more power-efficient and flexible alternative to traditional ring oscillator designs. By leveraging Xilinx Vivado for simulation and implementation, the proposed approach provides a scalable and efficient solution for FPGA-based applications requiring tunable oscillators

B. Implementation:

The implementation of the Hybrid Carry Chain & LUT-Based Ring Oscillator is carried out using Xilinx Vivado on an FPGA platform. The design process shown in figure 3 that involves the following steps:

➢ *FPGA Platform Selection and Tool Setup*

- The design is implemented on an FPGA, specifically targeting Xilinx 7-Series FPGAs, using the Vivado Design Suite.
- The hardware description is written in Verilog/VHDL, and synthesis, placement, and routing are performed using Vivado's integrated toolchain.

➢ *Hybrid Ring Oscillator Design*

- The ring oscillator is constructed using a combination of carry chains and LUT-based delay elements.
- The carry chains are implemented using dedicated FPGA carry logic, ensuring high-speed propagation.
- The LUT-based delay elements introduce controlled delays, allowing fine-tuned frequency control and power optimization.
- The hybrid oscillator structure is designed to allow parameter adjustments for different trade-offs between speed, power, and area utilization.

➢ *HDL Coding and Simulation*

- The oscillator circuit is written in Verilog, defining the hybrid structure and ensuring correct logic propagation.
- Testbenches are created for functional verification using Vivado's built-in simulation tools.
- The simulation verifies correct oscillation behavior, stability, and expected frequency output.

➢ *FPGA Synthesis and Implementation*

- The HDL design is synthesized, and resource utilization (LUTs, FFs, and carry chains) is analyzed.
- Placement and routing are performed using Vivado's implementation flow to optimize circuit performance.

Timing analysis ensures that the oscillator meets the required frequency and stability constraints.

➢ *Power and Performance Analysis*

- Power consumption is analyzed using Vivado Power Analyzer, comparing the proposed hybrid design against the traditional carry-chain-based oscillator.
- The impact of LUT-based delay elements on power efficiency is evaluated, demonstrating a reduction from 0.334W to 0.122W.
- Frequency tuning is tested by varying the number of LUT stages, confirming improved flexibility compared to the conventional design.

➢ *Hardware Validation on FPGA*

- The design is deployed on an FPGA board, and real-time measurements are taken using oscilloscopes and logic analyzers.
- The observed frequency, power characteristics, and overall stability are validated against simulation results

➢ *Implementation Flow Chart*

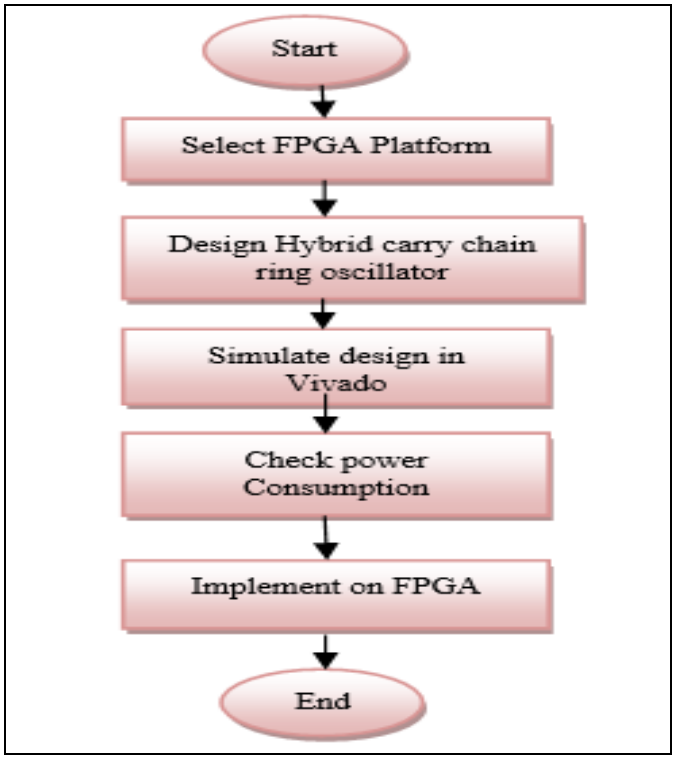

Fig 3 Implementation Flow Chart

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

V. SIMULATION RESULTS

➢ *Existing Method*

The RTL schematic of the existing ring oscillator illustrates in figure 4 the carry-chain-based architecture, where fast carry chains are used for delay generation. The circuit consists of carry logic elements, flip-flops, and input/output connections for generating a stable oscillation

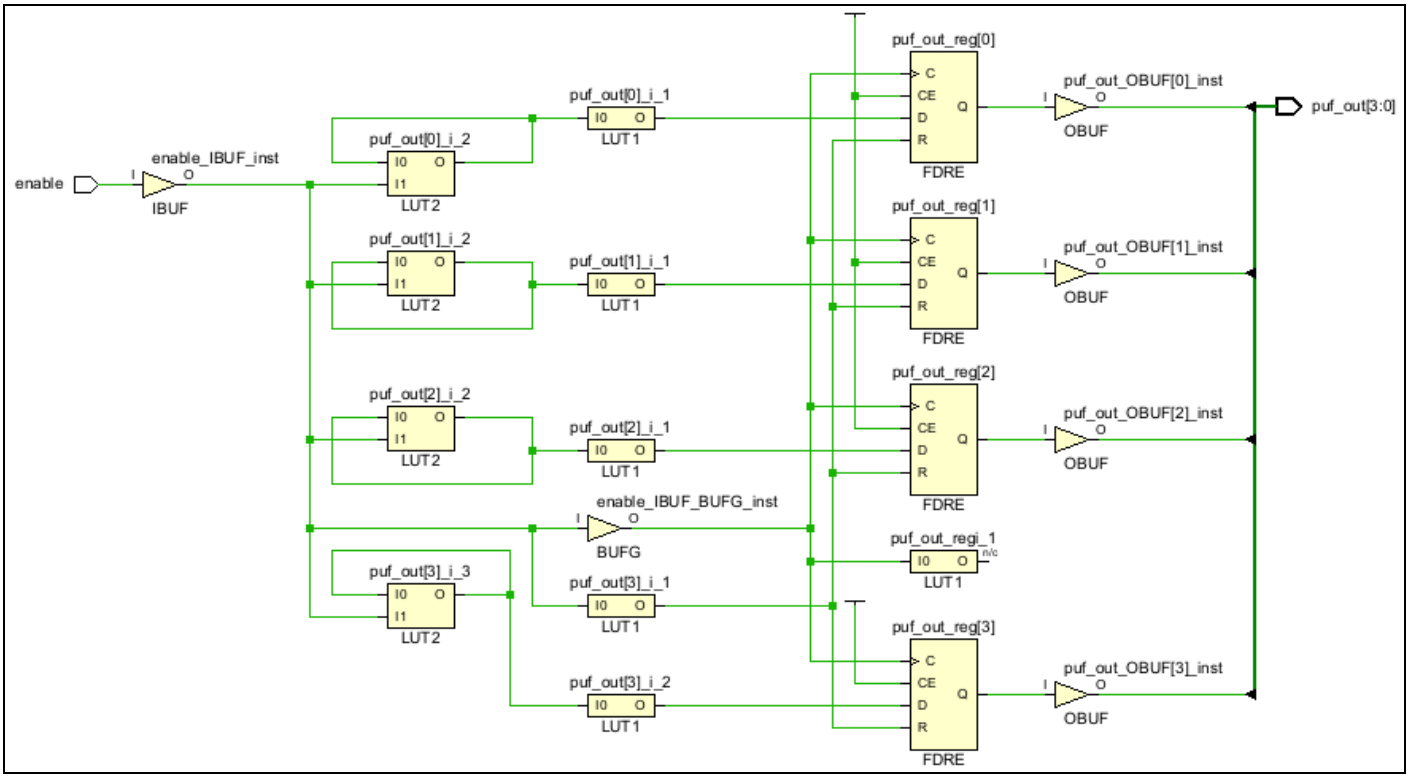

Fig 4 RTL Schematic

The power analysis shows in figure 5 that the existing method consumes 0.334W, which is relatively high due to the continuous toggling of carry chains. The dynamic power component is significant as carry chains are optimized for speed but not necessarily for power efficiency.

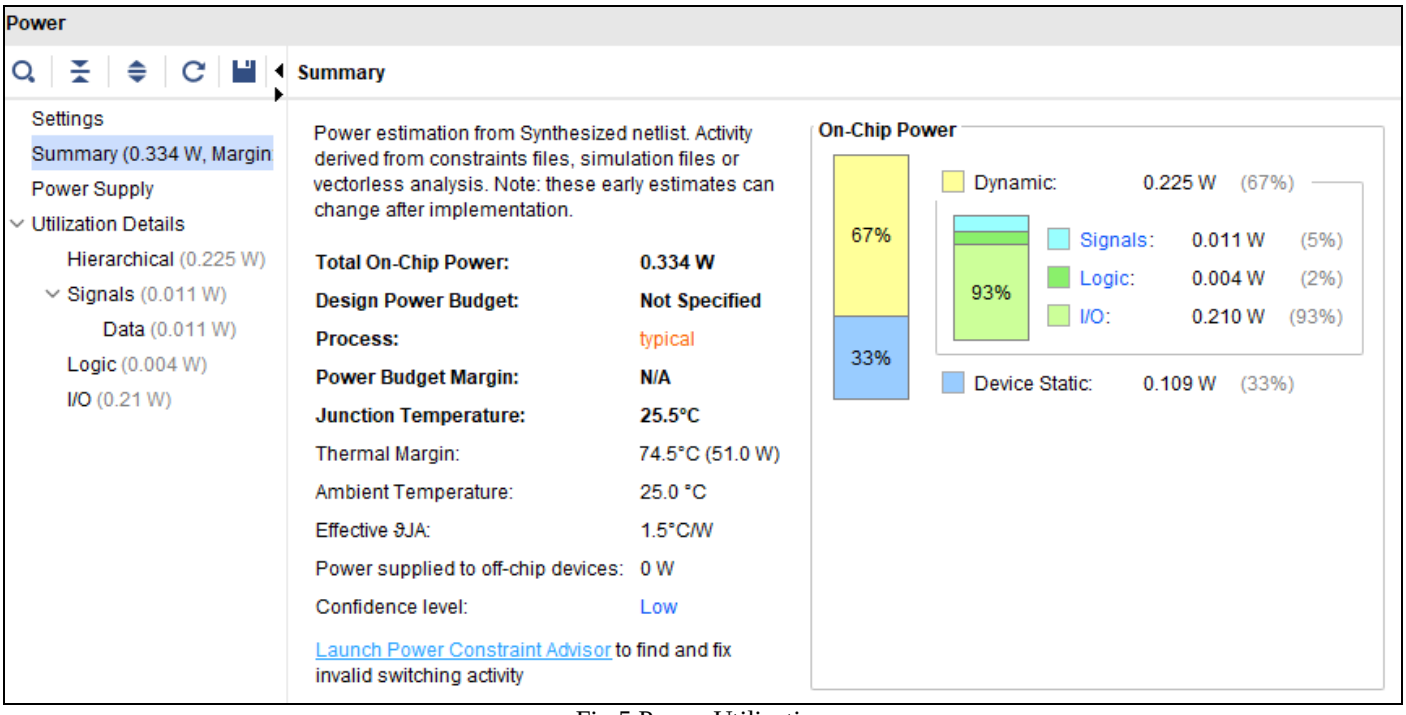

Fig 5 Power Utilization

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

The area utilization analysis shown in figure 6 that indicates that the existing oscillator occupies 10 LUTs, 4 Flip-Flops (FF), and 5 IOs. The large LUT consumption is due to the extensive use of carry chains, which increases area utilization and limits flexibility in FPGA resource allocation.

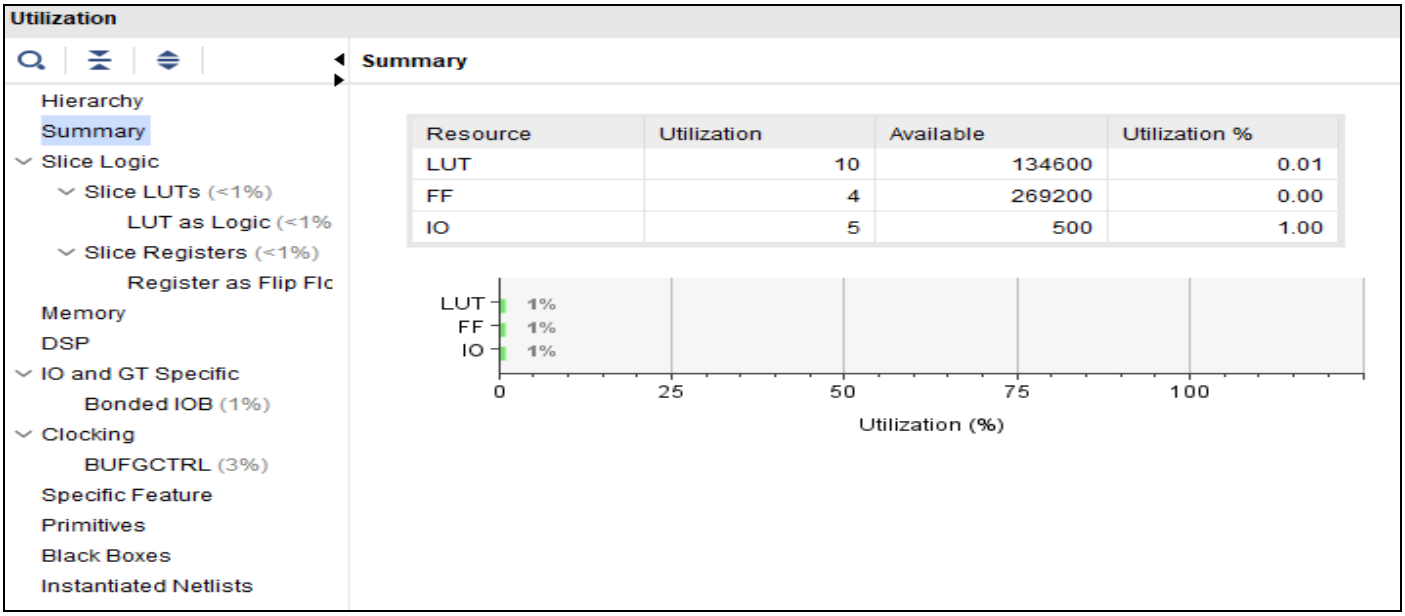

Fig 6 Area Utilization

➢ *Proposed Hybrid Ring Oscillator*

The RTL schematic for the proposed hybrid design shown in figure 7 that a combination of carry chains and LUT-based delay elements. This design allows controlled frequency tuning while reducing excessive power consumption.

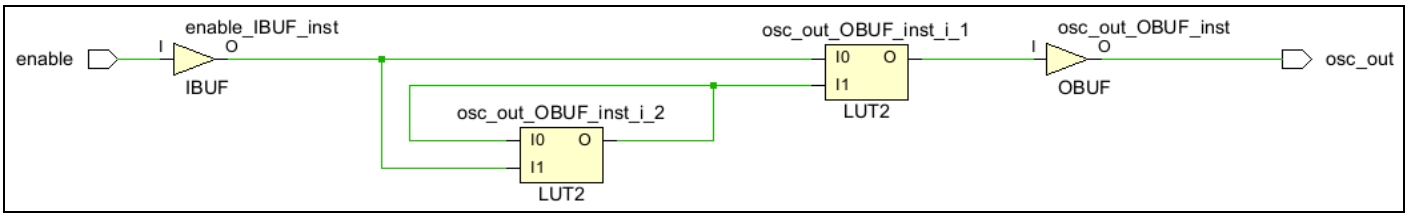

Fig 7 RTL Schematic

The proposed method shows in figure 8 that a significant power reduction to 0.122W, a 63% decrease compared to the existing method. The inclusion of LUT-based delay elements reduces switching activity, lowering dynamic power consumption.

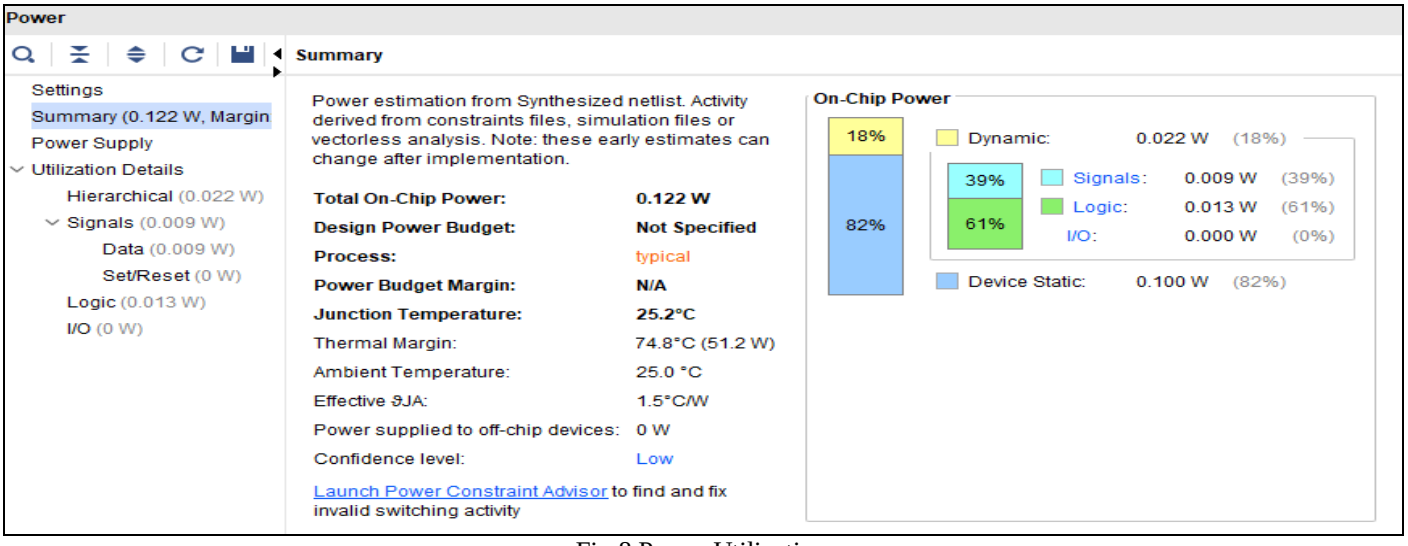

Fig 8 Power Utilization

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

The proposed oscillator utilizes only 2 LUTs and 2 IOs, significantly reducing FPGA resource usage. The elimination of unnecessary carry logic and optimization of delay elements contribute to reduced area consumption shown in figure 9.

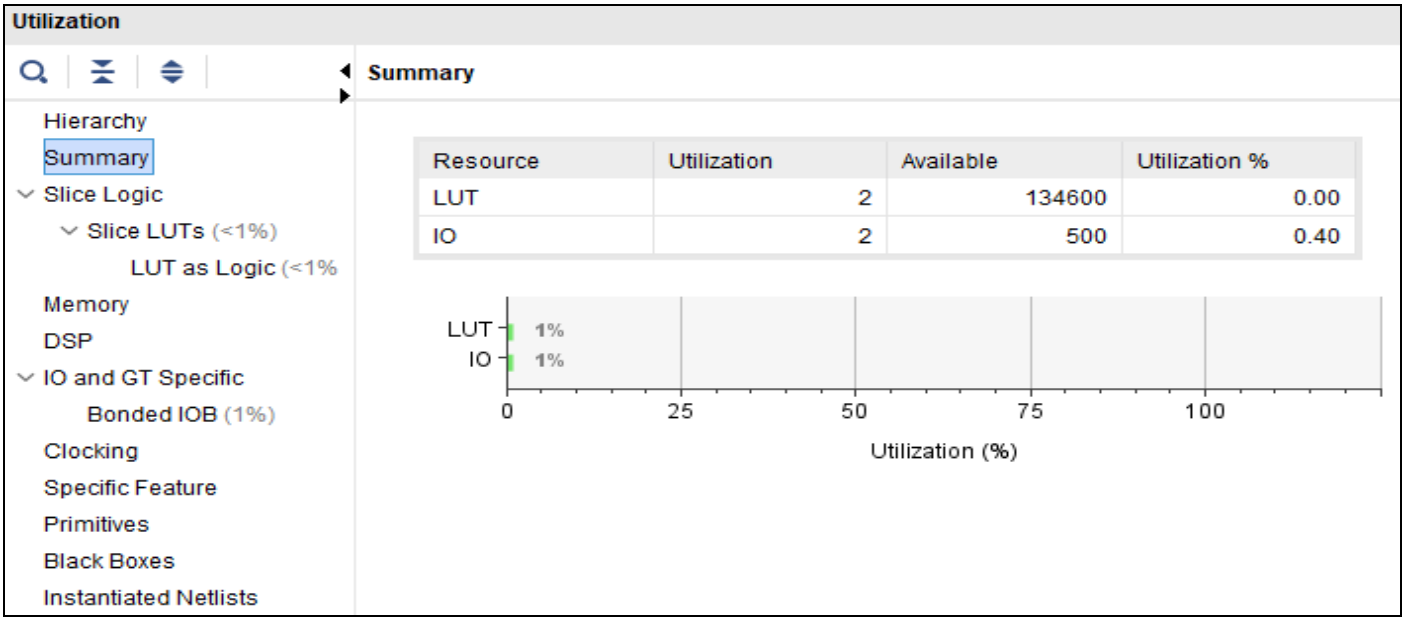

Fig 9 Area Utilization

The simulation waveform validates that the proposed oscillator produces a stable ring oscillation signal. The oscillation frequency is tunable, showcasing improved flexibility over the existing carry-chain-only design shown in figure 10.

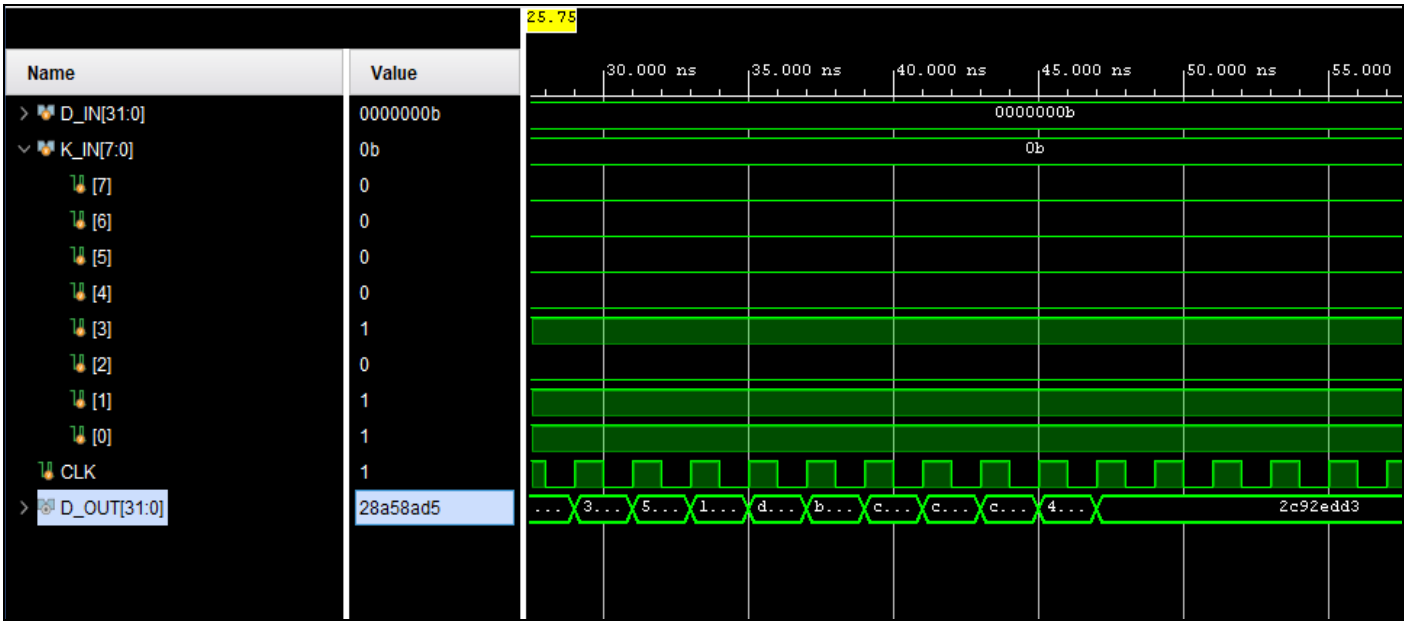

Fig 10 Simulation Waveform

➢ *Performance Comparision Table*

Table 1 Performance Comparison of Different Metrics

S. N	Parameter	Existing Method	Proposed method
1	Power Utilization	0.334w	0.122w
2	LUT	10	2
3	FF	4	-
4	IO	5	2

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

The experimental results clearly show that in figure 11 the proposed Hybrid Carry Chain and LUT-Based Ring Oscillator outperforms the traditional carry-chain-only oscillator in terms of power efficiency, resource utilization, and frequency flexibility. The reduction in power consumption from 0.334 W to 0.122 W demonstrates that introducing LUT-based delay elements effectively minimizes unnecessary switching activity, making the design suitable for low-power FPGA applications. Similarly, the reduction of LUT usage from 10 to 2 highlights a significant improvement in resource optimization, which frees FPGA logic elements for other critical tasks.

Beyond raw numbers, these improvements have practical significance. For example, in IoT and cryptographic systems, where devices often operate under strict energy budgets, the proposed oscillator provides a sustainable solution without compromising performance. Compared to state-of-the-art FPGA-based oscillators reported in literature [9], [12], the proposed design balances speed, efficiency, and tunability, making it more robust against design trade-offs. The inclusion of LUT-based delay elements not only enhances tunability but also makes the oscillator adaptable to varying application requirements. Thus, the discussion highlights that the hybrid design is not merely a theoretical improvement but a practically deployable architecture for secure and energy-constrained FPGA systems.

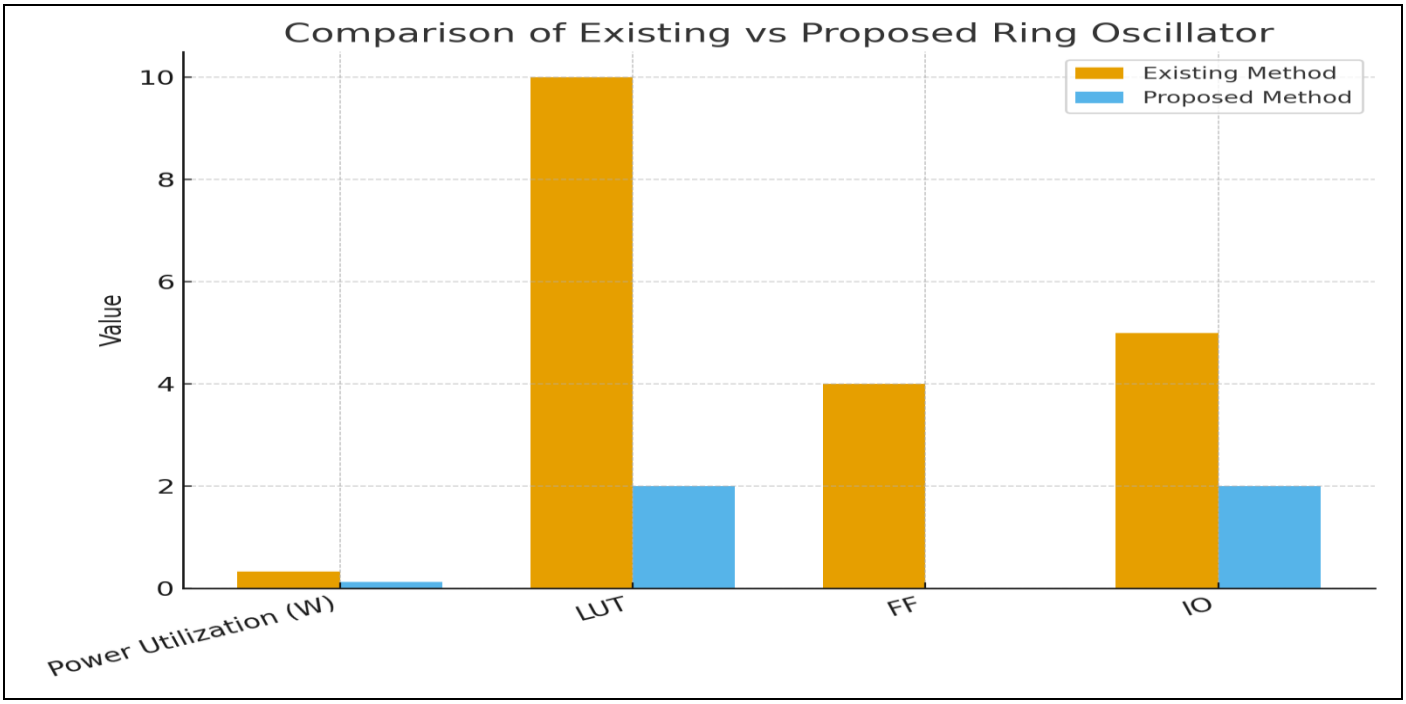

Fig 11 Performance Comparision Graph

VI. CONCLUSION AND FUTURE SCOPE

The implementation of the proposed Hybrid Carry Chain and LUT-Based Ring Oscillator on an FPGA demonstrates a significant improvement over traditional carry-chain-only designs by achieving both power efficiency and enhanced frequency tunability. Experimental results confirm that the design reduces power consumption from 0.334 W to 0.122 W and lowers LUT usage from 10 to 2, while maintaining stable oscillation performance. These optimizations make the architecture highly suitable for low-power, resource-constrained applications such as IoT devices, cryptographic modules, and hardware security systems. Furthermore, the integration of LUT-based delay elements provides flexible tuning capabilities that conventional designs cannot offer. Looking ahead, the scope of this work can be extended by incorporating dynamic power management techniques, adaptive frequency scaling, and exploring scalability across next-generation FPGA platforms, thereby broadening its applicability in future 6G, embedded, and secure computing environments.

REFERENCES

[1]. Kodýtek, R. Lórencz, and J. Buˇ cek, “Three counter value based ROPUFs on FPGA and their properties,” Microprocessors Microsystems, vol. 88, Feb. 2022, Art. no. 104375.

[2]. B. Halak, M. Zwolinski, and M. S. Mispan, “Overview of PUF-based hardware security solutions for the Internet of Things,” in Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS), Abu Dhabi, UAE, Oct. 2016, pp. 1–4.

[3]. M. Barbareschi, G. Di Natale, L. Torres, and A. Mazzeo, “A ring oscillator-based identification mechanism immune to aging and external working conditions,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 2, pp. 700–711, Feb. 2018.

[4]. B. Yang, V. Rožic, M. Grujic, N. Mentens, and I. Verbauwhede, “ES TRNG: A high-throughput, low-area true random number generator based on edge sampling,” IACR Trans. Cryptograph. Hardw. Embedded Syst., vol. 10, pp. 267–292, Aug. 2018.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov711

[5]. M. A. Prada-Delgado, C. Martínez-Gómez, and I. Baturone, " Auto calibrated ring oscillator TRNG based on jitter accumulation," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Seville, Spain, Oct. 2020, pp. 1– 4.
[6]. M. Grujic 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, Jun. 2022.
[7]. S. Moini et al., " Voltage sensor implementations for remote power attacks on FPGAs," ACM Trans. Reconfigurable Technol. Syst., vol. 16, no. 1, pp. 1– 21, Dec. 2022.
[8]. I. Giechaskiel, K. B. Rasmussen, and J. Szefer, " Measuring long wire leakage with ring oscillators in cloud FPGAs," in Proc. 29th Int. Conf. Field Program. Log. Appl. (FPL), Barcelona, Spain, Sep. 2019, pp. 45– 50.
[9]. M. Ebrahimi and Z. Navabi, " Selecting representative critical paths for sensor placement provides early FPGA aging information," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 39, no. 10, pp. 2976– 2989, Oct. 2020.
[10]. M. Alam, M. Tehranipoor, and D. Forte, " Recycled FPGA detection using exhaustive LUT path delay characterization and voltage scaling," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 12, pp. 2897– 2910, Dec. 2019.
[11]. T. Kilian, D. Tille, M. Huch, M. Hanel, and U. Schlichtmann, " Perfor mance screening using functional path ring oscillators," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 31, no. 6, pp. 711– 724, Jun. 2023.
[12]. S. Bae, M. Lee, S.-M. Yoo, and J.-Y. Sim, " A temperature compensated ring oscillator with LC-based period error detection," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 31, no. 12, pp. 2152– 2156, Dec. 2023, doi: 10.1109/TVLSI.2023.3322709.
[13]. B. Razavi, " The ring oscillator [a circuit for all seasons]," IEEE Solid StateCircuits Mag., vol. 11, no. 4, pp. 10– 81, Jun. 2019.
[14]. M. Elnawawy, A. Farhan, A. A. Nabulsi, A. R. Al-Ali, and A. Sagahy roon, " Role of FPGA in Internet of Things applications," in Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT), Ajman, United Arab Emirates, Dec. 2019, pp. 1– 6
[15]. N. N. Anandakumar, M. S. Hashmi, and M. Tehranipoor, " FPGA based physical unclonable functions: A comprehensive overview of theory and architectures," Integration, vol. 81, pp. 175– 194, Nov. 2021.
[16]. C. Gu, C. H. Chang, W. Liu, N. Hanley, J. Miskelly, and M. O' Neill, " A large scale comprehensive evaluation of single-slice ring oscillator and PicoPUF bit cells on 28 nm Xilinx FPGAs," in Proc. 3rd ACM Workshop Attacks Solutions Hardw. Secur. Workshop, London, U.K., Nov. 2019, pp. 101– 106.
[17]. A. Wild, G. T. Becker, and T. Güneysu, " On the problems of realizing reliable and efficient ring oscillator PUFs on FPGAs," in Proc. IEEE Int. Symp. Hardw. Oriented Security Trust (HOST), McLean, VA, USA, 2016, pp. 103– 108.
[18]. H. Kareem and D. Dunaev, " Towards performance optimization of ring oscillator PUF using Xilinx FPGA," in Proc. 17th Iberian Conf. Inf. Syst. Technol. (CISTI), Madrid, Spain, Jun. 2022, pp. 1– 6.
[19]. O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, " A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices," in Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL), Lausanne, Switzerland, Aug. 2016, pp. 1– 10.
[20]. AMD. Xilinx 7 Series FPGAs Data Sheet: Overview DS180 (v2.6.1). Accessed: May 4, 2024. [Online]. Available: https://docs.amd.com/v/u/en-US/ds180_7Series_Overview