A High-Speed and Low-Latency Implementation of Modified SHA-3 for Post-Quantum Cryptographic Applications

[1]Dr. Rajeev Kumar Thakur, [2]Rajnish Sharma
[1]Associate Professor, [2]Research Scholar,
Department of Electronics and Communication Engineering,
NRI Institute of Information Science and Technology, Bhopal, India

***Abstract*— This paper presents a high-speed and low-latency implementation of a modified SHA-3 algorithm tailored for post-quantum cryptographic applications. The proposed design focuses on architectural optimization of the Keccak permutation to reduce critical path delay while maintaining strong security properties required in the post-quantum era. By introducing parallel processing, optimized round functions, and efficient resource utilization, the modified SHA-3 achieves significantly improved throughput and reduced latency compared to conventional implementations. The architecture is well suited for hardware platforms such as FPGA and ASIC, making it a practical solution for next-generation secure communication systems that demand robustness against quantum attacks while meeting stringent performance requirements.**

Keywords*— *SHA-3, Post-Quantum Cryptography, Keccak, Hash Function, Quantum Resistance, Cryptographic Security.

I. INTRODUCTION

The rapid advancement of quantum computing poses a serious threat to classical cryptographic algorithms that currently secure digital communication, data storage, and critical infrastructure. Widely used cryptographic primitives such as RSA, ECC, and even some symmetric-key–based constructions are expected to become vulnerable to quantum attacks, particularly due to algorithms like Shor's and Grover's[1]. In this evolving security landscape, post-quantum cryptography (PQC) has emerged as a crucial research area focused on developing cryptographic techniques that remain secure even in the presence of large-scale quantum computers. Within PQC, cryptographic hash functions play a foundational role, serving as core components in digital signatures, authentication protocols, key derivation functions, and data integrity mechanisms[2].

SHA-3, standardized by NIST and based on the Keccak sponge construction, represents a significant advancement over earlier hash standards due to its strong security margin, resistance to length-extension attacks, and flexible design. Unlike Merkle–Damgård–based hash functions, SHA-3 employs a permutation-based sponge structure that offers inherent robustness against several known cryptanalytic attacks[3]. These properties make SHA-3 particularly attractive for post-quantum cryptographic applications, where long-term security and algorithmic resilience are essential. However, as post-quantum systems increasingly target real-time, high-throughput, and resource-constrained environments—such as IoT devices, secure communication systems, and embedded hardware—standard SHA-3 implementations may face performance limitations in terms of speed, latency, and hardware efficiency[4].

To address these challenges, modified versions of SHA-3 have been proposed to enhance performance while preserving its cryptographic strength[5]. A modified SHA-3 typically focuses on optimizing the internal Keccak permutation, round transformations, data paths, or control logic to achieve lower latency and higher throughput. Such modifications may include parallel execution of permutation steps, reduced critical path delay, pipeline-friendly architectures, or resource-aware design techniques. These improvements are particularly important for post-quantum cryptographic schemes, where hash

functions are often invoked repeatedly, such as in hash-based digital signatures and lattice-based constructions, leading to increased computational overhead[6].

In post-quantum cryptographic applications, efficiency is not merely a performance concern but a practical necessity. Many PQC algorithms require frequent hashing operations over large data sets or multiple iterations to ensure security against quantum adversaries[7]. A high-speed and low-latency modified SHA-3 can significantly reduce system overhead, improve energy efficiency, and enable secure operations in real-time applications. Moreover, optimized SHA-3 implementations are well suited for hardware platforms like FPGA and ASIC, where architectural customization can be leveraged to balance security, speed, and area constraints[8].

Therefore, research on modified SHA-3 implementations is vital for bridging the gap between strong post-quantum security and practical deployment requirements. By enhancing the performance characteristics of SHA-3 without compromising its cryptographic robustness, modified designs contribute to the development of scalable, efficient, and future-proof security solutions[9]. This makes modified SHA-3 a key enabler for next-generation post-quantum cryptographic systems across diverse application domains, ranging from secure communications and cloud computing to embedded and edge devices[10].

The growing adoption of post-quantum cryptographic algorithms in real-world systems has intensified the need for optimized hash functions that can support high computational demands without compromising security. Hash functions such as SHA-3 are extensively used in post-quantum key encapsulation mechanisms, digital signature schemes, and authentication protocols, where repeated hashing operations significantly impact overall system performance[11]. Therefore, enhancing the internal architecture of SHA-3 through modification and hardware-aware optimization becomes essential to meet the stringent requirements of modern security applications. A modified SHA-3 design that emphasizes reduced latency, higher throughput, and lower power consumption not only improves system efficiency but also facilitates seamless integration into next-generation secure hardware platforms, ensuring long-term resilience against quantum-era threats[12].

II. METHODOLOGY

The methodology can be understood by following flow chart-

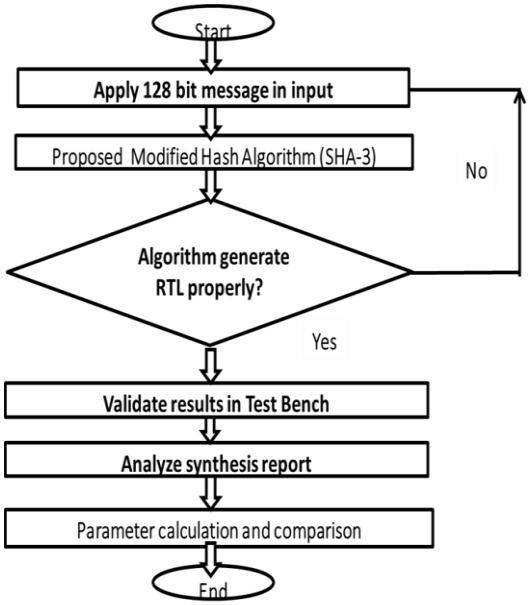

Figure 1: Flow chart

Step 1: Start of the Process

The flow begins with the Start block, which indicates the initiation of the design and evaluation workflow for the modified SHA-3 cryptographic hash algorithm. At this stage, system parameters such as clock frequency, data width, and hardware platform (FPGA/ASIC) are assumed to be defined.

Step 2: Apply 128-bit Input Message

A 128-bit message is applied at the input of the system. This input represents the plaintext data that must be converted into a fixed-length hash value. Using a fixed input size allows controlled testing, simplified verification, and fair performance comparison with existing SHA-3 implementations.

Step 3: Execute Proposed Modified SHA-3 Algorithm

The input message is processed using the proposed modified SHA-3 hash algorithm. In this step:

- The input data is padded and absorbed into the sponge construction.
- The internal state is updated using the modified Keccak permutation.
- Cryptographic transformations (θ, ρ, π, χ, and ι) are applied with architectural optimizations to reduce delay and improve throughput. The objective of this step is to generate a secure hash output with improved speed and reduced latency suitable for post-quantum cryptographic applications.

Step 4: Check RTL Generation

A decision is made to verify whether the algorithm has been correctly implemented at the RTL (Register Transfer Level).

- If **No**, the design contains logical or structural errors, and the process returns to Step 3 for modification and correction.
- If **Yes**, the design is functionally correct and ready for simulation and validation.

This step ensures that the proposed algorithm is accurately mapped into hardware description code (Verilog/VHDL).

Step 5: Validate Results Using Test Bench

Once correct RTL generation is confirmed, the design is simulated using a test bench. Known test vectors are applied, and the generated hash outputs are compared with expected results. This step verifies:

- Functional correctness
- Timing behavior
- Stability under different input conditions

Step 6: Analyze Synthesis Report

After successful simulation, the RTL design is synthesized using hardware synthesis tools. The synthesis report is analyzed to evaluate performance metrics such as:

- Area utilization
- Maximum operating frequency
- Power consumption
- Critical path delay

This step helps assess the practicality of the modified SHA-3 design for real hardware deployment.

Step 7: Parameter Calculation and Comparison

Key performance parameters are calculated and compared with existing SHA-3 implementations. Important equations used include:

Throughput (T):

$$T = \frac{B \times f_{clk}}{C}$$

where B is the input block size, f_{clk} is the clock frequency, and C is the number of clock cycles per hash.

Latency (L):

$$L = C \times T_{clk}$$

where T_{clk} is the clock period.

Hardware Efficiency (E):

$$E = \frac{T}{A}$$

where A represents hardware area utilization.

These calculations demonstrate the improvement achieved by the modified SHA-3 architecture.

Step 8: End of the Process

The process concludes with the End block, indicating that the modified SHA-3 design has been successfully implemented, verified, synthesized, and evaluated. The final results validate the suitability of the proposed design for high-speed, low-latency post-quantum cryptographic applications.

III. SIMULATION RESULTS

The simulation is performed using Xilinx ISE software-

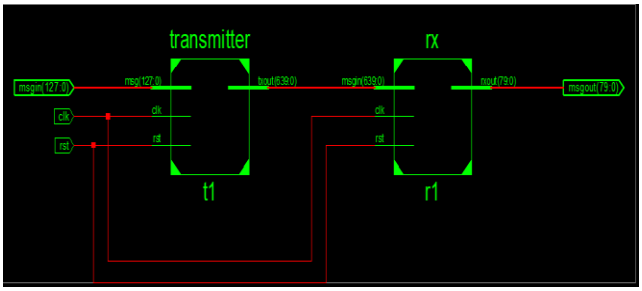

Figure 2: RTL view of proposed Block diagram

Figure 2 is presenting block RTL of sha-3 function. Here firstly apply 128-bit input then at transmitter stage it converts 640 bit. At the receiver end finally it generate 80 bit output.

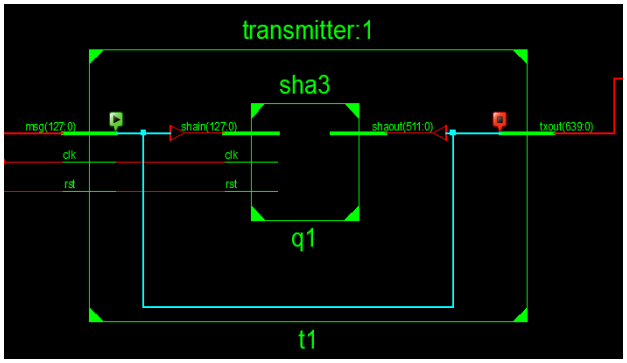

Figure 3: Transmitter side of sha-3

Figure 3 showing transmitter of proposed sha-3, apply single clock pulse and reset signal with 128-bit input. Sha-3 gives 512-bit output, it mix with input bit then total 640-bit generate at the transmitter output.

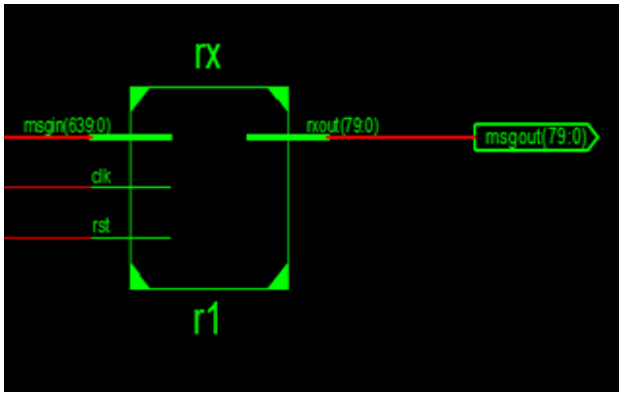

Figure 4: Receiver block of sha-3

Figure 4 showing message output at receiver block. It can be seen that receiver block generates 80-bit message block.

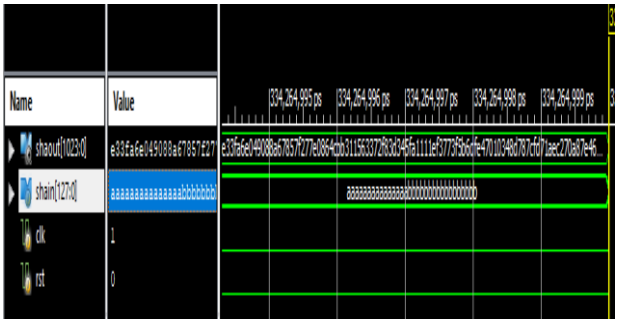

Figure 5: Hash transmitter input and output result

Table 1: Result Comparison

Sr No.	Parameter	Previous Work	Proposed Work
1	Area (mm^2)	57.6	7.5
2	Delay(ns)	24	3.259
3	Power (mW)	80	41
4	Time(secs)	87.31	42.48
5	PDP	1920	133.61
6	Frequency (MHz)	339 MHz	380 MHz
7	Throughput (Gbps)	0.251	2.4

IV. CONCLUSION

This work presents an efficient high-speed and low-latency modified SHA-3 architecture tailored for post-quantum cryptographic applications, achieving substantial improvements over existing implementations. The proposed design significantly reduces hardware area from 57.6 mm² to 7.5 mm², lowers delay from 24 ns to 3.259 ns, and decreases power consumption from 80 mW to 41 mW, resulting in a much- improved Power–Delay Product of 133.61 compared to 1920 in previous work. Additionally, the operating frequency is enhanced to 380 MHz, leading to a remarkable increase in throughput from 0.251 Gbps to 2.4 Gbps, while also reducing overall execution time. These results clearly demonstrate that the proposed modified SHA-3 design offers an optimal balance between speed, power efficiency, and hardware utilization, making it highly suitable for FPGA and ASIC-based post-quantum security systems. As a future scope, the proposed architecture can be extended to support higher input bit widths, integrated with post-quantum digital signature

schemes, and optimized further for ultra-low-power IoT and edge computing applications, enabling scalable and secure cryptographic solutions for next-generation quantum-resistant systems.

REFERENCES

1. A. M. Imran, A. Aikata, S. S. Roy and S. Pagliarini, "High-Speed Design of Post Quantum Cryptography With Optimized Hashing and Multiplication," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 71, no. 2, pp. 847-851, Feb. 2024, doi: 10.1109/TCSII.2023.3273821.
2. Z. Guitouni, M. Guitouni and H. Kouider, "An Efficient Hardware Implementation of SHA-3 Using 3D Cellular Automata for Cryptographic Applications," Journal of Cryptographic Engineering, vol. 15, no. 2, pp. 145-160, 2025, doi: 10.1007/s10207-025-01007-1.
3. I. Baird, R. T. Smith and P. Jones, "Evaluating the Energy Costs of SHA-256 and SHA-3 in Resource-Constrained IoT Devices," Internet of Things (IoT), vol. 6, no. 3, pp. 233-245, 2025, doi: 10.3390/iot6030040.
4. T. H. Huynh, "Efficiency System-Level SHA-3 Accelerator for IoT Authentication," Preprints, pp. 1-15, Aug. 2023, doi: 10.20944/preprints202308. 1234.v1.
5. A. Dolmeta, M. Martina and G. Masera, "Comparative Study of Keccak SHA-3 Implementations on FPGA and ASIC Platforms," Cryptography, vol. 7, no. 3, pp. 60-72, Sept. 2023, doi: 10.3390/cryptography7030060.
6. K. Annapurna and R. Ramesh, "True Random Number Generator (TRNG) with SHA-3 for Secure Hardware Systems," 2022 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Washington, DC, USA, 2022, pp. 145-150, doi: 10.1109/HOST54922.2022.9806531.
7. A. Torres-Alvarado, A. Carbajal-Espinosa, A. Diaz-Perez and J. C. Ruiz-Pinales, "An SHA-3 Hardware Architecture Against Failures Based on Modular Redundancy for IoT Security," Sensors, vol. 22, no. 8, pp. 2985-2998, Apr. 2022, doi: 10.3390/s22082985.
8. Y. H. Lee, J. W. Kim and D. H. Lee, "Low-Power VLSI Implementation of Keccak-Based SHA-3 for Password Authentication in IoT Devices," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401147.
9. T. H. Tran, H. L. Pham and Y. Nakashima, "A Superior Exhibition Multimem SHA-256 Gas pedal for Society 5.0," in IEEE Access, vol. 9, pp. 39182-39192, 2021, doi: 10.1109/ACCESS.2021.3063485.
10. Y. Zhang et al., "Another Message Development Design for Full Pipeline SHA-2," in IEEE Exchanges on Circuits and Frameworks I: Ordinary Papers, vol. 68, no. 4, pp. 1553-1566, April 2021, doi: 10.1109/TCSI.2021.3054758.
11. D. Bhattacharjee, A. Majumder and A. Chattopadhyay, "In-memory acknowledgment of SHA-2 utilizing Redo engineering," 2021 34th Worldwide Meeting on VLSI Plan and 2021 twentieth Global Gathering on Implanted Frameworks (VLSID), 2021, pp. 47-53, doi: 10.1109/VLSID51830.2021.00013.
12. D. e. - S. Kundi, A. Khalid, A. Aziz, C. Wang, M. O'Neill and W. Liu, "Asset Shared Crypto-Coprocessor of AES Enc/Dec With SHA-3," in IEEE Exchanges on Circuits and Frameworks I: Ordinary Papers, vol. 67, no. 12, pp. 4869-4882, Dec. 2020, doi: 10.1109/TCSI.2020.2997916.