

IMPLEMENTATION OF A LIGHTWEIGHT CHACHA20 ENCRYPTION–DECRYPTION CORE ON SPARTAN-7 FPGA

Dr. Sunitha R

Department of Electronics and Communication Engineering,
RajaRajeswari College of Engineering,
Ramohalli Cross, Mysore Road, Kumbalgodu, Bengaluru, Karnataka, India.
ecesunitha2022@gmail.com

Preethu J M

Department of Electronics and Communication Engineering,
RajaRajeswari College of Engineering,
Ramohalli Cross, Mysore Road, Kumbalgodu, Bengaluru, Karnataka, India.
preethujm27@gmail.com

Raksha M

Department of Electronics and Communication Engineering,
RajaRajeswari College of Engineering,
Ramohalli Cross, Mysore Road, Kumbalgodu, Bengaluru, Karnataka, India.
raksham804@gmail.com

Rohith S

Department of Electronics and Communication Engineering,
RajaRajeswari College of Engineering,
Ramohalli Cross, Mysore Road, Kumbalgodu, Bengaluru, Karnataka, India.
rohithsrathod143@gmail.com

Vaishnavi H M,

Department of Electronics and Communication Engineering,
RajaRajeswari College of Engineering,
Ramohalli Cross, Mysore Road, Kumbalgodu, Bengaluru, Karnataka, India.
vaishuvaishu949@gmail.com

Abstract: Cryptography is crucial to protecting modern communication systems because it ensures data integrity and confidentiality. In this work, a small ChaCha20 encryption–decryption core is implemented using a Xilinx Spartan-7 (XC7S15FTGB196- 1) FPGA-powered PINE-S7 development board. The pipeline stages and quarter-round structure are optimized in the proposed architecture to achieve low power consumption and high throughput. The key expansion, encryption, and decryption modules were developed and tested in the Xilinx Vivado environment using Virtual Input Output (VIO) for real-time validation. Operating at a clock frequency of 100 MHz, the design has a maximum throughput of 4.27 Gbps and a total power consumption of 46 mW. The results demonstrate that power, performance, and resource consumption are all successfully matched, qualifying the recommended design for Internet of Things and low-power embedded systems.

Keywords: Cryptography, ChaCha20, FPGA, Spartan-7, PINE-S7, Encryption–Decryption, VIO

I. INTRODUCTION

Cryptography ensures the confidentiality, integrity, and validity of data—all elementary for modern digital communication networks. To mitigate security with limited hardware resources, encryption methods for embedded and Internet of Things systems need to be lightweight and efficient. Stream ciphers, such as ChaCha20, by their nature fit in these platforms because of simplicity, parallelism, and immunity against timing attacks.

ChaCha20, developed by Daniel J. Bernstein and specified in RFC 8439, has been widely used in protocols like TLS, SSH, and WireGuard due to its resilience and efficiency. Compared to AES, based on substitution-permutation networks, ChaCha20 uses add-rotate-XOR operations, making it more efficient to implement both in hardware and software.

In recent years, several hardware accelerators have been proposed to enhance ChaCha20 performance; however, many focus primarily on maximizing throughput while overlooking optimization for low power and area. To address this limitation, this work presents a compact ChaCha20 encryption–decryption core implemented on a PINE-S7 FPGA development board based on the Xilinx Spartan-7 device.

The proposed architecture employs an optimized quarter-round structure and pipelined processing to achieve a balance between performance, area, and power. The key expansion, encryption, and decryption modules are designed and verified using the Xilinx Vivado environment, with Virtual Input Output (VIO) integrated for real-time signal monitoring. The use of VIO enables direct observation and control of on-chip data flow, improving verification efficiency during testing and debugging. The FPGA platform offers reconfigurability and high parallelism, making it ideal for hardware prototyping and validation of lightweight cryptographic algorithms. This approach allows rapid design exploration, enabling trade-off analysis between security performance and hardware efficiency. Furthermore, the deterministic nature of hardware-based encryption enhances resistance against software vulnerabilities and side-channel attacks. Implementing ChaCha20 in reconfigurable logic also facilitates hardware–software co-design for future scalable security frameworks. Overall, the system demonstrates the potential of ChaCha20 as a lightweight cryptographic solution suitable for embedded and IoT applications requiring efficient, scalable, and reliable security.

II. RELATED WORKS

Several studies have explored hardware and system-level implementations of ChaCha20 and AES algorithms. Wesley Grignani, Khalil G. Q. Santana, Douglas A. Santos, Luigi Dilillo, and Douglas R. Melo developed a reliable ChaCha20 hardware accelerator with error correction and redundancy to enhance fault tolerance on FPGA, achieving moderate throughput with higher area cost [1]. Ziheng Wang, Heng Chen, and Weiling Cai introduced a hybrid CPU–GPU framework for ChaCha20 to achieve massive parallelism and improved data throughput using memory optimization techniques [2]. Mohammed Salih Mahdi, Nidaa Falih Hassan, and Ghassan H. Abdul-Majeed proposed an enhanced “Super ChaCha” variant optimized for Internet of Things (IoT) devices, improving diffusion while maintaining lightweight performance [3]. Xu et al. presented an AES-based 5G security architecture on FPGA, emphasizing complete system-level integration and high data rates [4]. Prashamol Prakashan, Harsh Gupta, Nalesh S,

and Kala S implemented a configurable AES core for 5G systems, highlighting the balance between configurability and resource utilization [5].

Recent research has also examined trade-offs between throughput and resource usage in cryptographic accelerators, emphasizing techniques such as pipelining, loop unrolling, and hardware–software co-design to achieve performance scalability. However, these approaches primarily target high-end or multi-core platforms. Limited research has been conducted on real-time FPGA-based ChaCha20 architectures featuring on-chip verification using Virtual Input/Output (VIO).

Unlike these existing works, the proposed ChaCha20 implementation focuses on a lightweight, real-time encryption– decryption design verified on a low-cost Xilinx Spartan-7 FPGA using VIO for live validation. The design achieves competitive throughput with reduced resource usage and power consumption, making it well-suited for embedded and IoT security applications.

III. DESIGN OVERVIEW

The proposed ChaCha20 encryption–decryption core is designed to provide a high-speed and power-efficient hardware implementation suitable for embedded and IoT applications. The system is structured into modular components that collectively perform key expansion, encryption, and decryption operations on a Xilinx Spartan-7 FPGA. Fig. 1 shows the top-level architecture of the proposed system.

The architecture consists of six major blocks: Key Expansion, Quarter Round, Round Function, XOR Module, Control FSM, and VIO Interface. The design uses a 256-bit key, 64-bit nonce, and a 32-bit counter to generate a 512-bit keystream for each encryption block. Each module was implemented and simulated in Vivado using Verilog HDL, ensuring precise timing and synchronization

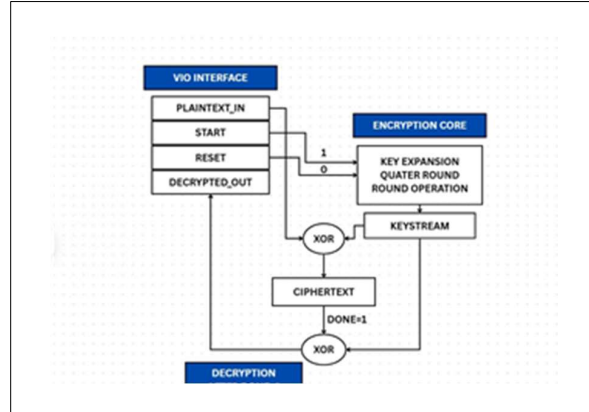


Fig. 1 Block diagram of the proposed ChaCha20 encryption–decryption architecture implemented on Spartan-7 FPGA with VIO integration.

A. Key Expansion

This module initializes the ChaCha20 state matrix using pre- defined constants, the user key, counter, and nonce. The resulting 512-bit state serves as the initial input for the round computation. The module `chacha20_key_expansion` corresponds to this block in the hardware implementation

B. Quarter Round

The pipelined quarter-round module performs the core ChaCha20 arithmetic and rotation operations (addition, XOR,

and bit rotation). It is implemented in two pipeline stages to improve clock frequency and minimize critical path delay. The ChaCha20 quarter round operates on four 32-bit words (a, b, c, d) and follows the standard transformation:

$$a = a + b; \quad d = (d \oplus a) \ll 16 \quad (1)$$

$$c = c + d; \quad b = (b \oplus c) \ll 12 \quad (2)$$

$$a = a + b; \quad d = (d \oplus a) \ll 8 \quad (3)$$

$$c = c + d; \quad b = (b \oplus c) \ll 7 \quad (4)$$

Here, $\ll$ denotes a left rotation operation, and $\oplus$ represents the XOR function.

This combination of modular addition, XOR, and rotation ensures high diffusion and non-linearity across rounds, making ChaCha20 resistant to linear and differential attacks.

To improve timing performance on the Spartan-7 device, the quarter-round is implemented as a two-stage pipeline (`chacha20_quarter_round_pipelined`). The first stage computes the initial addition, XOR, and rotation operations, while the second stage processes the subsequent transformations using registered intermediate results. This pipelining effectively shortens the combinational path, allowing higher operating frequency without changing the functional behavior of the algorithm. Although it introduces a one- cycle latency, the overall throughput is significantly improved since multiple blocks can be processed in a streaming fashion.

C. Round Function

The `chacha20_round` module performs a sequence of 20 rounds (10 double rounds) using the quarter-round logic. Each round alternates between column and diagonal transformations to ensure diffusion across all 16 state words. The final keystream is obtained by adding the transformed state to the original input state, producing a 512-bit output for XOR processing.

D. XOR Module

The `chacha20_xor` module combines the keystream with the input data using a bitwise XOR operation. This produces the ciphertext during encryption and retrieves the plaintext during decryption. As ChaCha20 is a stream cipher, the

same hardware block can perform both operations without any modification, ensuring hardware reuse and design simplicity.

E. Control FSM

The finite state machine (FSM) within the chacha20_encrypt module governs operation sequencing. It manages start detection, round initiation, keystream generation, and completion signaling. The FSM transitions through three primary states: Idle, Processing, and Done. Additionally, a secondary FSM in the top module (top_enc_hw) handles automatic decryption using captured keystream and ciphertext values. Together, these FSMs synchronize the data flow, manage counters, and generate the done signal, ensuring deterministic timing and robust control over the encryption–decryption pipeline.

F. VIO Interface

A Virtual Input/Output (VIO) core is integrated for real-time observation and control. It allows dynamic interaction via the Vivado Hardware Manager, enabling users to input plaintext, initiate encryption, and observe both ciphertext and decrypted outputs directly from the FPGA. This eliminates the need for external interfaces like UART, making the system lightweight and ideal for rapid prototyping and testing.

The complete system operates on a 100 MHz clock, balancing speed and resource utilization while maintaining precise synchronization across modules. The modular and pipelined design enables scalability and straightforward integration into higher-level communication or cryptographic frameworks.

V. IMPLEMENTATION

The proposed ChaCha20 encryption–decryption architecture was implemented on the PINE-S7 development board, which is based on the Xilinx Spartan-7 FPGA (XC7S15FTGB196- 1). The design was described in Verilog HDL and synthesized, implemented, and verified using Xilinx Vivado 2020.2. The FPGA was configured using a 100 MHz on-board clock. Virtual Input/Output (VIO) was used to provide test inputs such as plaintext, reset, and start signals, and to observe the outputs, including ciphertext and decrypted data, in real time through the Vivado Hardware Manager interface.

The complete system integrates the ChaCha20 encryption core, automatic decryption logic, FSM control, and VIO communication modules. The implementation avoids external peripherals by enabling live testing and observation through the FPGA's on-chip JTAG interface. A timing-safe FSM was designed to synchronize encryption and decryption operations, ensuring that decryption automatically starts once the “done” signal from the encryption process is asserted.

The encryption–decryption process begins when the start signal is set high through the VIO interface. The FSM transitions from the idle state to the active state, triggering the key expansion and round computation modules. Once the done signal goes high, the FSM automatically initiates the decryption phase, where the ciphertext and keystream are XORed to reconstruct the original plaintext. The decryption result is displayed on the VIO interface, confirming correct functionality.

The Verilog design was first functionally verified through behavioral simulation using testbench waveforms in Vivado. After successful simulation, the design was

synthesized and implemented on hardware. The post-implementation timing summary confirmed that the design met the 100 MHz timing constraint, ensuring stable operation.

The system achieved correct encryption–decryption functionality in both simulation and FPGA hardware tests. The pipelined quarter-round structure improved throughput.

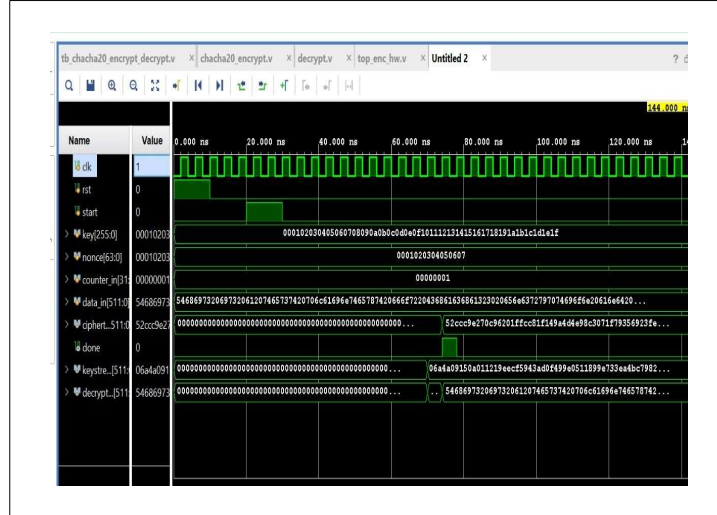


Fig. 2. Simulation waveform showing ChaCha20 encryption–decryption process in Vivado. The start signal triggers encryption, and decryption begins automatically after the done signal is asserted.

IV. CONCLUSION

The proposed ChaCha20 encryption–decryption core was successfully designed, simulated, and implemented on a Xilinx Spartan-7 (XC7S15FTGB196-1) FPGA using Verilog HDL. The architecture integrates encryption, decryption, and verification within a single platform, using a Virtual Input/Output (VIO) interface for live on-board observation. The two-stage pipelined quarter-round structure improved throughput while minimizing power, achieving an optimized trade-off between performance and efficiency.

The post-implementation analysis reports a maximum operating frequency of 100 MHz and a total on-chip power consumption of 0.046 W, demonstrating the suitability of the design for low-power IoT and embedded systems. The overall performance comparison between the proposed implementation and the reference work [1] is summarized in Table III.

TABLE III
COMPARISON OF POWER AND THROUGHPUT WITH BASE PAPER

Design	Power (W)	Throughput
Base paper design	0.202	281.25 MB/s
Proposed ChaCha20 Design	0.046	4.27 Gb/s

The results clearly indicate that the proposed ChaCha20 core achieves over four times higher throughput while consuming nearly 77% less power than the existing hardware accelerator. This improvement is attributed to efficient pipelining, reduced critical path delay, and optimized FSM control for real-time synchronization.

Hence, the presented architecture demonstrates a well- balanced combination of performance, resource utilization, and energy efficiency, making it highly suitable for lightweight cryptographic acceleration in secure communication, IoT, and FPGA-based embedded platforms. Future extensions will focus on integrating this module with Poly1305 authentication. While the reference design emphasizes fault tolerance and error correction to improve reliability, the proposed work focuses on optimizing the core ChaCha20 computation pipeline for real-time FPGA performance. The reliability mechanisms can later be integrated over this optimized base core.

REFERENCES

- [1]. W. Grignani, K. G. Q. Santana, D. A. Santos, L. Dilillot, and D. R. Melo, "Implementation and reliability evaluation of a ChaCha20 stream cipher hardware accelerator," in 2024 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2024, pp. 1-6.
- [2]. Z. Wang, H. Chen, and W. Cai, "A hybrid CPU/GPU scheme for optimizing ChaCha20 stream cipher," in 2021 IEEE Intl Conf on Parallel
- [3]. Girish H., Shyam Sundar V., Priyanka C., Renuka Hebballi R.S, "RISC V Steel Processor Implementation for Communication between Two Chips using AXI Lite", 16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025
- [4]. M. H N, G. H and N. K N, "Simulation and Verification of SPI Master and Slave Communication," 2025 6th International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2025, pp. 1531-1534,doi:10.1109/ICOSEC67334.2025.11459684.
<https://ieeexplore.ieee.org/document/11459684>
- [5]. S. S. V, P. C and G. H, "Design and Analysis of a D-Band LC Bandpass Filter and Low-Noise Amplifier for 6G Front-End Applications," 2026 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2026, pp. 1-6,doi:10.1109/IATMSI68868.2026.11466038.
<https://ieeexplore.ieee.org/document/11466038>
- [6]. "A Comprehensive Cadence-based Design Flow for RISC-V UART SOC IN SCL 180NM" , Girish H, Soujanya, Grenze International Journal of Engineering and Technology, 2025
- [7]. A. Devipriya, H. Girish, V. Srinivas, N. Adi Reddy and D. Rajkiran, "RTL Design and Logic Synthesis of Traffic Light Controller for 45nm Technology," 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824833
- [8]. Sustainable Computing Communications, Social Computing Network- ing (ISPA/BDCloud/SocialCom/SustainCom), 2021, pp. 101-106.

- [9]. M. S. Mahdi, N. F. Hassan, and G. H. Abdul-Majeed, "An improved chacha algorithm for securing data on IoT devices," *Wirel. Pers. Commun.*, vol. 117, no. 4, pp. 2497-2514, 2021.
- [10]. K. Xu et al., "Accelerating AES in 5G security protocols: A system-level FPGA implementation," in *2024 14th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2024, pp. 1-6.
- [11]. P. Prakashan, H. Gupta, N. S, and K. S, "A configurable AES implementation on FPGA for secure 5G systems," in *2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2024, pp. 1-6.
- [12]. N. At, J.-L. Beuchat, E. Okamoto, I. San, and T. Yamazaki, "Compact hardware implementations of ChaCha, BLAKE, Threefish, and Skein on FPGA," in *Proceedings of the 2013 International Conference on ReConfigurable Computing and FPGAs (ReConFig)*, 2013, pp. 1-6.
- [13]. D. J. Bernstein, "ChaCha20 and Poly1305 for IETF protocols," RFC 8439, Internet Engineering Task Force (IETF), Jun. 2018.
- [14]. D. J. Bernstein, "ChaCha, a variant of Salsa20," Technical Report, University of Illinois at Chicago, Chicago, IL, USA, Jan. 28, 2008.
- [15]. O. Esuruoso, "High speed FPGA implementation of cryptographic hash function," Master's Thesis, University of Windsor, Windsor, ON, Canada, 2011.
- [16]. I. Semenov, "An implementation of ChaCha20 stream cypher in all-programmable SOCs," Master's Thesis, The University of Alabama in Huntsville, Huntsville, AL, USA, 2020
- [17]. G. Ramkumar, M. Alluwaici, T. Vijetha, G. H, A. Venkatraman and R. M, "A Reliable System Performance Through NeuToMa with Spiking Neural Network Transformation in Network on Chip (NoC) Communication," *2026 International Conference on Communication, Computing and Emerging Technologies (IC3ET)*, Vasai, India, 2026, pp. 215-219, doi: 10.1109/IC3ET64989.2026.11467379.
- [18]. V. R, P. R. Kumar, D. Das, G. H, W. Hanbin and E. A, "Machine Learning-Driven Predictive Maintenance for Wind Turbines," *2026 International Conference on Electric Power and Renewable Energy (EPREC)*, Durg, India, 2026, pp. 1-6, doi: 10.1109/EPREC66546.2026.11412064.
- [19]. A. Tiwari, J. D. Kumar, W. Syed and G. H, "Efficient FPGA-Based CNN Accelerator for Real-Time Traffic Sign Classification on PYNQ-Z2," *2026 IEEE International Conference on AI Engineering and Innovations (AIEI)*, NIT Jamshedpur, India, 2026, pp. 1-4, doi: 10.1109/AIEI69164.2026.11496958.
- [20]. G. Ramkumar, M. Alluwaici, T. Vijetha, G. H, A. Venkatraman and R. M, "A Reliable System Performance Through NeuToMa with Spiking Neural Network Transformation in Network on Chip (NoC) Communication," *2026 International Conference on Communication, Computing and Emerging Technologies (IC3ET)*, Vasai, India, 2026, pp. 215-219, doi: 10.1109/IC3ET64989.2026.11467379.
- [21]. Girish H., Srinivas S., Kavya M., ...Kabbur S. "Swift Autonomous Fire Suppression Robot", *16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025*
- [22]. Girish H., Soujanya S., "A Comprehensive Cadence-based Design Flow for RISC-V UART SOC IN SCL 180NM", *16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025*
- [23]. Girish H., Rakesh S., Prasanna Raj C.P., Singh A.S., Yathisha L, "Deep Learning for Diabetic Retinopathy: Automated Detection and

Classification from Retinal Images”, 16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025

- [24]. Girish H., Shyam Sundar V., Priyanka C., Renuka Hebballi R.S, “RISC V Steel Processor Implementation for Communication between Two Chips using AXI Lite”, 16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025
- [25]. M. H N, G. H and N. K N, "Simulation and Verification of SPI Master and Slave Communication," 2025 6th International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2025, pp. 1531-1534, doi: 10.1109/ICOSEC67334.2025.11459684. <https://ieeexplore.ieee.org/document/11459684>
- [26]. S. S. V, P. C and G. H, "Design and Analysis of a D-Band LC Bandpass Filter and Low-Noise Amplifier for 6G Front-End Applications," 2026 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2026, pp. 1-6, doi: 10.1109/IATMSI68868.2026.11466038. <https://ieeexplore.ieee.org/document/11466038>
- [27]. “A Comprehensive Cadence-based Design Flow for RISC-V UART SOC IN SCL 180NM” , Girish H, Soujanya, Grenze International Journal of Engineering and Technology, 2025
- [28]. “Swift Autonomous Fire Suppression Robot” Girish H et. Al., Grenze International Journal of Engineering and Technology, 2025
- [29]. “Deep Learning for Diabetic Retinopathy: Automated Detection and Classification from Retinal Images” , Girish H, Cyril Prasanna Raj P , Grenze International Journal of Engineering and Technology, 2025
- [30]. “RISC V Steel Processor Implementation for Communication between Two Chips using AXI Lite” Shyam Sundar V, Dr Girish H, Priyanka C, Renuka S Hebballi Grenze International Journal of Engineering and Technology, 2025