Project Update

ChaCha20 Stream Cipher Hardware Implementation

May 2025

Introduction

ChaCha20 is a modern stream cipher designed by Daniel J. Bernstein. It offers strong security, high performance, and simple design, making it a popular choice in secure communications like TLS and SSH.
Poly1305 is often paired with ChaCha20 to provide message authentication (MAC), forming the ChaCha20-Poly1305 AEAD scheme widely used in internet protocols.

This is my first project update on implementing ChaCha20 in hardware using SystemVerilog. I'm focusing on building a clean, modular, and synthesizable design. I'm excited to continue improving the architecture and exploring integration with Poly1305. Looking forward to posting more updates as development progresses!

Project Overview

ChaCha20 is a stream cipher widely adopted for its simplicity, efficiency, and security—especially in constrained environments like embedded systems and mobile devices. Its resilience against timing attacks and suitability for high-speed applications have made it a cornerstone in modern cryptographic protocols.

The goal of this project is to implement the ChaCha20 encryption algorithm in hardware using SystemVerilog, with an emphasis on modularity, clarity, and synthesis readiness. This implementation showcases the real-world application of cryptographic theory in digital logic design, enabling high-throughput, low-latency encryption tailored for hardware accelerators, FPGAs, and ASICs.

This update covers the current progress on core module development, quarter-round operations, and block-level orchestration, laying the groundwork for a complete hardware ChaCha20 engine and future integration with the Poly1305 authenticator.

Key Achievements

Core Module Development

- **ChaCha State Matrix Generator**: Implemented initial state setup with 256-bit key, 96-bit nonce, and block counter input
- **Quarter-Round Operations**: Designed state machine-driven quarter-round logic with proper ARX (Add-Rotate-XOR) operations
- **Block Function**: Combined matrix generator and quarter-round operations to form the complete ChaCha20 block function
- **Block Counter Module**: Created parameterized counter as a separate component that interfaces with the block function through higher-level integration

</> Technical Implementation

- **Architecture**: Modular design with separate state initialization and transformation modules
- **State Management**: 8-state stepper FSM for quarter-round execution (S0-S7)
- **Round Control**: 8 quarter-round types (Q0-Q7) implementing both column and diagonal operations
- **Performance**: 20-round ChaCha20 implementation with optimized clocking