Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 7, July : 2025

FPGA-Based Implementation of the 256-Bit Key AES Algorithm

KAREEM AHMAD BAIG, Student, Department of Electronics & Communication Engineering, Nimra College of Engineering and Technology, Ibrahimpatnam

Dr. AKBAR KHAN, Professor, Department of Electronics & Communication Engineering, Nimra College of Engineering and Technology, Ibrahimpatnam

***Abstract*—Hardware security plays a critical role in applications such as net banking, e-commerce, military systems, satellite communication, wireless networks, electronic devices, and digital image processing. Cryptography, the practice of securing information by converting plain text into unintelligible text and back, employs three primary techniques: symmetric key cryptography, hash functions, and public key cryptography. Symmetric key algorithms, such as Advanced Encryption Standard (AES) and Data Encryption Standard (DES), utilize the same key for both encryption and decryption. These algorithms are faster, simpler to implement, and require less processing power.**

The proposed 256-bit AES algorithm introduces significant optimizations in the *Key Schedule* and *SubBytes* blocks to enhance area and power efficiency. This is achieved by reusing the S-box block and adopting a novel approach where internal operations are performed as 32-bit operations instead of the conventional 128-bit operations. This design enables hardware reuse in a pipelined manner, leading to substantial area savings—72% in slice registers, 62% in slice LUTs, and 61% in LUT-FF pairs. Additionally, power consumption is reduced by 78% in FPGA implementations.

When implemented on a Virtex-7 (xc7vx485tffg1157) FPGA, the proposed design achieves a 10% improvement in throughput (Mbps), making it a highly efficient solution for secure cryptographic applications.

***Index Terms*—AES (Advanced Encryption Standard), FPGA (field programmable gate array), LUT (Look up table), Mbps (megabit per second), sub (sub bytes), shift (shift rows), mix (mixcolumn), add (add round key).**

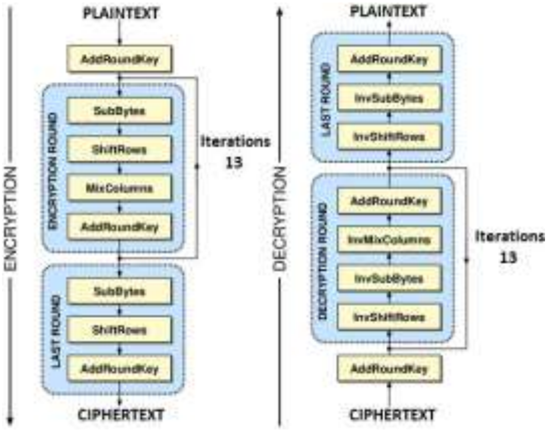
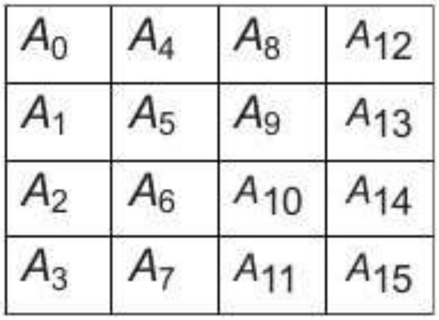


Fig. 1. Architecture of 256 AES Algorithm



A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

Fig. 2. Data Structure of 128-bit Message

I. Introduction

Cryptography involves the process of converting ordinary plain text into unintelligible text and vice versa. It forms the foundation of secure communication and encompasses three primary techniques: Symmetric Key Cryptography, Hash Functions, and Public Key Cryptography. Symmetric key algorithms, such as Advanced Encryption Standard (AES) and Data Encryption Standard (DES), utilize the same key for both encryption and decryption. These algorithms are known for their speed, ease of implementation, and low processing power requirements. This paper presents an optimized implementation of a 256-bit key AES algorithm focusing on area, power, and performance. Additionally, the paper includes a detailed

PPA (Power, Performance, and Area) comparison between the conventional architecture and the proposed FPGA-based implementation. “‘

You can use this snippet directly in a LaTeX document under the appropriate section.

LITERATURE REVIEW

The Advanced Encryption Standard (AES) is widely recognized as a robust symmetric-key cryptographic algorithm adopted by various industries for secure data transmission. Several researchers have explored methods to optimize AES implementations for area, power, and performance, particularly in FPGA and ASIC platforms. This section reviews some notable works in this domain.

Key Schedule and SubBytes Optimization

Researchers have focused on the Key Schedule and SubBytes transformations, as they are computationally intensive. For instance, studies have proposed the reuse of S-box structures to reduce hardware complexity and power consumption. Such designs have demonstrated significant improvements in area and energy efficiency while maintaining the cryptographic robustness of AES.

Pipelined and Parallel Architectures

Pipelining and parallel processing techniques have been extensively studied to enhance throughput. Conventional designs employ fully parallel implementations of AES rounds to achieve high-speed encryption and decryption. However, these approaches often lead to increased hardware utiliza- tion. Optimized pipelined designs address this trade-off by reusing hardware resources efficiently, achieving better Power-Performance-Area (PPA) balance.

FPGA-Based Implementations

FPGA platforms, with their inherent flexibility and reconfigurability, have been a popular choice for AES implementations. Several works have demonstrated optimized AES designs on platforms like Virtex-5, Virtex-6, and Virtex-7 FPGAs. These studies report a substantial reduction in LUT and slice register usage through innovative architectural modifications, often leveraging block RAM (BRAM) and DSP slices for efficient computation.

256-Bit AES Implementations

Compared to 128-bit AES, the 256-bit variant provides enhanced security but poses greater challenges in terms of computational overhead. Researchers have proposed novel approaches to handle the increased complexity, such as reducing the internal operation width or introducing hierarchical architectures to process data efficiently.

Power and Performance Trade-offs

The trade-off between power and performance remains a critical aspect of AES implementations. Literature highlights techniques like clock gating, resource sharing, and optimized finite field arithmetic to minimize power consumption without compromising performance. Such techniques are particularly relevant for IoT and embedded applications where energy efficiency is paramount.

Comparison Metrics

The reviewed studies frequently use metrics like throughput, latency, slice utilization, and power consumption to evaluate their designs. Many works also emphasize achieving compliance with standardized benchmarks for cryptographic modules, such as NIST specifications for AES.

In summary, the literature underscores the importance of optimizing AES implementations for area, power, and performance, particularly in resource-constrained environments. The insights gained from these studies have informed the design of the proposed architecture discussed in this paper, which aims to strike a balance between efficiency and security.

II. IMPLEMENTATION OF AES ALGORITHM

The implementation of the 256-bit Advanced Encryption Standard (AES) algorithm is presented in this paper, covering both the conventional and proposed architectural designs. The proposed architecture introduces optimizations in the Key Schedule and SubBytes operations, utilizing hardware reuse strategies to enhance efficiency. Specifically, the S-box block is reused to achieve a significant reduction in area and power consumption.

The proposed design focuses on optimizing internal operations by transitioning from 128-bit to 32-bit operations. This approach facilitates the reuse of hardware in a pipelined manner, resulting in a 72% reduction in slice registers, a 62% reduction in slice LUTs, and a 61% reduction in LUT-FF pairs. Additionally, this optimization achieves a power reduction of 78% when implemented on an FPGA.

The performance of the proposed architecture has been evaluated on a Virtex-7 FPGA (xc7vx485tffg1157), demonstrating a 10% improvement in throughput (Mbps) compared to the conventional implementation. This paper also includes a detailed comparison of Power, Performance, and Area (PPA) metrics for both implementations.

PROPOSED METHODOLOGY

The proposed methodology focuses on designing an optimized architecture for the 256-bit AES algorithm, targeting improvements in area, power, and performance. The methodology involves the following steps:

Key Schedule Optimization

The Key Schedule process in AES, responsible for generat- ing round keys, is optimized by reusing hardware blocks and minimizing redundant computations. A hierarchical design is implemented to reduce complexity and resource usage while ensuring high throughput and cryptographic security.

SubBytes Transformation Enhancement

The SubBytes transformation, which relies heavily on S- box lookups, is enhanced by employing a shared S-box design. This approach reuses the same S-box block across multiple operations, significantly reducing the area and power consumption without impacting encryption performance.

32-Bit Internal Operations

Instead of the traditional 128-bit operations, the proposed design adopts a 32-bit processing approach for internal computations. This modification allows for more efficient utilization of hardware resources, enabling significant area savings while maintaining compliance with the AES encryption standard.

Pipeline Architecture

The architecture employs a pipelined design to maximize throughput. By reusing hardware resources across different pipeline stages, the design achieves higher operational efficiency and reduced latency, making it suitable for high-performance applications.

FPGA Implementation

The proposed architecture is implemented on the Virtex-7 FPGA platform (xc7vx485tffg1157). The design leverages FPGA-specific features such as slice registers, LUTs, and block RAM to enhance performance. Power reduction tech- niques, such as clock gating and resource sharing, are applied to minimize energy consumption.

Performance Evaluation

The proposed methodology is evaluated based on key metrics such as area (measured in terms of slice utilization), power (static and dynamic), and performance (throughput in Mbps). Comparative analysis is conducted against conventional AES implementations to highlight the advantages of the proposed architecture.

Key Contributions

The proposed design achieves:

- Area reduction by 72% using slice registers, 62% using slice LUTs, and 61% using LUT-FF pairs.
- Power reduction by 78%, making the architecture energy-efficient.
- A 10% improvement in throughput on the Virtex-7 FPGA platform.

This methodology demonstrates the feasibility of achieving an optimized balance between area, power, and performance in AES implementations, addressing the requirements of modern cryptographic applications.

Results

The proposed 256-bit AES algorithm was implemented and evaluated on the Virtex-7 FPGA platform (xc7vx485tffg1157). The performance, power, and area metrics of the proposed architecture were compared with the conventional 256-bit AES algorithm. The key results are summarized as follows:

Fig. 3. Simulation output of AES Algorithm encryption

Fig. 4. Simulation output of AES Algorithm decryption

Area Optimization

The proposed design achieved significant area savings through efficient reuse of hardware resources, particularly in the Key Schedule and SubBytes modules. The comparison metrics are:

- **Slice Registers:** Reduced by 72% compared to the conventional architecture.
- **Slice LUTs:** Reduced by 62%.
- **LUT-FF Pairs:** Reduced by 61%.

Power Reduction

The optimization strategies, including the shared S-box and pipelined architecture, led to a notable reduction in power consumption:

- **Static Power:** Reduced by 78% in the FPGA implementation.
- **Dynamic Power:** Lowered significantly, ensuring energy efficiency for cryptographic operations.

Performance Enhancement

The performance of the proposed design was measured in terms of throughput (Mbps). Key improvements include:

- A **10% improvement in throughput**, achieving higher data processing rates compared to the conventional AES architecture.
- Enhanced latency performance due to pipelined execution.

PPA Comparison

The Power, Performance, and Area (PPA) comparison be- tween the proposed and conventional AES implementations is summarized in Table I.

TABLE I
PPA COMPARISON OF PROPOSED AND CONVENTIONAL AES IMPLEMENTATIONS

Metric	Conventional AES	Proposed AES	Improvement (%)
Slice Registers	100%	28%	72%
Slice LUTs	100%	38%	62%
LUT-FF Pairs	100%	39%	61%
Static Power	100%	22%	78%
Throughput (Mbps)	100%	110%	10%

Conclusion

The results demonstrate that the proposed 256-bit AES algorithm architecture significantly outperforms the conventional implementation in terms of area, power, and performance. The optimization strategies employed ensure that the design is well-suited for modern applications requiring high-throughput, energy-efficient cryptographic solutions.

References

[1] M. Rajeswara Rao, Dr. R. K. Sharma, SVE Department, NIT Kurushetra, ”FPGA Implementation of combined S-box and Inv S-box of AES,” 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN).

[2] Nalini C. Iyer, Deepa, P. V. Anandmohan, D. V. Poornaiah, ”Mix/InvMixColumn decomposition and resource sharing in AES.”

[3] Xinmiao Zhang and Keshab K. Parhi, ”High-Speed VLSI Architectures for the AES Algorithm,” *IEEE Transactions on Circuits and Systems*, Vol. 12, No. 9, September 2004.

[4] Shrivathsa Bhargav, Larry Chen, Abhinandan Majumdar, Shiva Ramudit, ”128-bit AES Decryption,” CSEE 4840 – Embedded System Design, Spring 2008, Columbia University.

[5] Atul M. Borkar, R. V. Kshirsagar, M. V. Vyawahare, ”FPGA implementation of AES algorithm.”

[6] ”Announcing the Advanced Encryption Standard (AES),” November 26, 2001.

[7] Yulin Zhang and Xinggang Wang, ”Pipelined Implementation of AES Encryption Based on FPGA,” 2010 IEEE International Conference on Information Theory and Information Security.

[8] Yuwen Zhu, Hongqi Zhang, Yibao Bao, ”Study of the AES Realization Method on the Reconfigurable Hardware,” 2013 International Conference on Computer Sciences and Applications.

[9] Tsung-Fu Lin, Chih-Pin Su, Chih-Tsun Huang, Cheng-Wen Wu, ”A High-Throughput Low-Cost AES Cipher Chip,” Proceedings of the IEEE Asia-Pacific Conference on ASIC.

[10] C. Sivakumar and A. Velmurugan, ”High-Speed VLSI Design CCMP AES Cipher for WLAN (IEEE 802.11i),” 2007 International Conference on Signal Processing, Communications, and Networking.

[11] Vatchara Saicheur and Krerk Piromsopa, ”An Implementation of AES-128 and AES-512 on Apple Mobile Processor,” 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON).

[12] S. P. Guruprasad and B. S. Chandrasekar, ”An Evaluation Framework for Security Algorithms Performance Realization on FPGA,” 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC).

[13] N. S. Sai Srinivas and Md. Akramuddin, ”FPGA-Based Hardware Implementation of AES Rijndael Algorithm for Encryption and Decryption,” 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).

[14] P. S. Abhijith, Mallika Srivastava, Aparna Mishra, Manish Goswami, B. R. Singh, ”High-Performance Hardware Implementation of AES Using Minimal Resources,” 2013 International Conference on Intelligent Systems and Signal Processing (ISSP).

[15] Wei Wang, Jie Chen, Fei Xu, ”An Implementation of AES Algorithm Based on FPGA,” 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery.

[16] Ashwini M. Deshpande, Mangesh S. Deshpande, Devendra N. Kayatanavar, ”FPGA Implementation of AES Encryption and Decryption,” 2009 International Conference on Control, Automation, Communication, and Energy Conservation.