Journal of Cardiovascular Disease Research

ISSN: 0975-3583, 0976-2833 VOL10, ISSUE 04, 2019

OPTIMIZED AREA AND SPEED ARCHITECTURES FOR THE MIX COLUMN OPERATION OF THE ADVANCED ENCRYPTION STANDARD

VANGALA NAGA RAJU, BOGGARAPU KANTHA RAO, DASARI RAMESH

Dept of ECE,

Priyadarshini Institute of Science and Technology for Women Khammam.

ABSTRACT

The MixColumns operation in the Advanced Encryption Standard (AES) is crucial for ensuring diffusion, making it one of the key transformations that secure AES against cryptanalysis. This research paper investigates optimized architectures for the MixColumns operation, focusing on both area and speed efficiency. We explore various architectural designs, including look-up table approaches, composite field arithmetic, and novel gate-level optimizations, to achieve a balance between low latency and reduced resource consumption. The proposed optimized designs are evaluated using both software simulations and hardware implementations on FPGA platforms. Our findings demonstrate significant improvements in processing speed and area utilization compared to conventional MixColumns architectures, making them suitable for high-performance cryptographic applications in constrained environments, such as IoT devices and embedded systems.

INTRODUCTION

The Advanced Encryption Standard (AES) is a widely adopted symmetric encryption algorithm used for securing sensitive data across various digital platforms. Among its operations, the MixColumns transformation plays a critical role in achieving diffusion, an essential aspect of cryptographic security. The MixColumns operation involves matrix multiplication over a finite field, which, while providing security, also imposes computational and area-related challenges, especially in hardware implementations. As demand grows for high-speed and area-efficient cryptographic solutions in devices with limited computational resources, optimizing the MixColumns operation becomes imperative. This paper addresses the need for optimized architectures for the MixColumns operation, focusing on reducing area and improving speed without compromising security. We explore various approaches, including look-up table methods, composite field arithmetic techniques, and innovative gate-level optimizations. The goal is to provide a thorough analysis of these methods, compare their performance metrics, and propose a set of optimized architectures tailored for specific applications, such as Internet of Things (IoT) devices, embedded systems, and other low-power, high-performance environments.

ISSN: 0975-3583, 0976-2833 VOL10, ISSUE 04, 2019

LITERATURE SURVEY

The MixColumns operation, a linear transformation within AES, is traditionally implemented using straightforward matrix multiplication over the Galois Field GF(2^8). While secure, this method is often resource-intensive in hardware implementations. Several research efforts have focused on optimizing this operation for better performance in constrained environments.

Early works explored the use of look-up tables (LUTs) to speed up the MixColumns computation. However, this approach, while fast, consumes a significant amount of memory, making it unsuitable for low-area applications. Subsequent studies have suggested using composite field arithmetic to reduce the complexity of the finite field multiplications required in MixColumns. By breaking down the multiplication into simpler operations over smaller fields, these techniques reduce both the computational complexity and the area footprint of the hardware implementations.

More recent advancements have proposed gate-level optimizations that aim to reduce the number of logic gates required for the MixColumns operation. These techniques often employ methods such as gate-level pipelining, retiming, and logic minimization to achieve substantial gains in speed and area efficiency. Additionally, hybrid approaches that combine multiple optimization techniques have shown promise in achieving balanced performance metrics suitable for various applications.

This literature review highlights the evolution of optimization techniques for the MixColumns operation, setting the stage for the proposed architectural designs presented in this paper.

PROPOSED SYSTEM

The proposed system introduces several optimized architectures for the MixColumns operation, each tailored to meet specific design constraints related to speed, area, or a balance of both. The key innovations in these architectures include:

1. Optimized Look-Up Table (LUT) Approach: This architecture utilizes compact LUTs to store precomputed values for the MixColumns operation, significantly reducing the computational time. By minimizing the size of the LUTs and employing efficient memory management techniques, this approach achieves a balance between speed and area efficiency.

2. Composite Field Arithmetic Optimization: This design leverages composite field arithmetic to decompose the Galois Field GF(2^8) multiplication into operations over smaller fields. By doing so, the architecture reduces the complexity of the finite field multiplications, resulting in lower

ISSN: 0975-3583, 0976-2833 VOL10, ISSUE 04, 2019

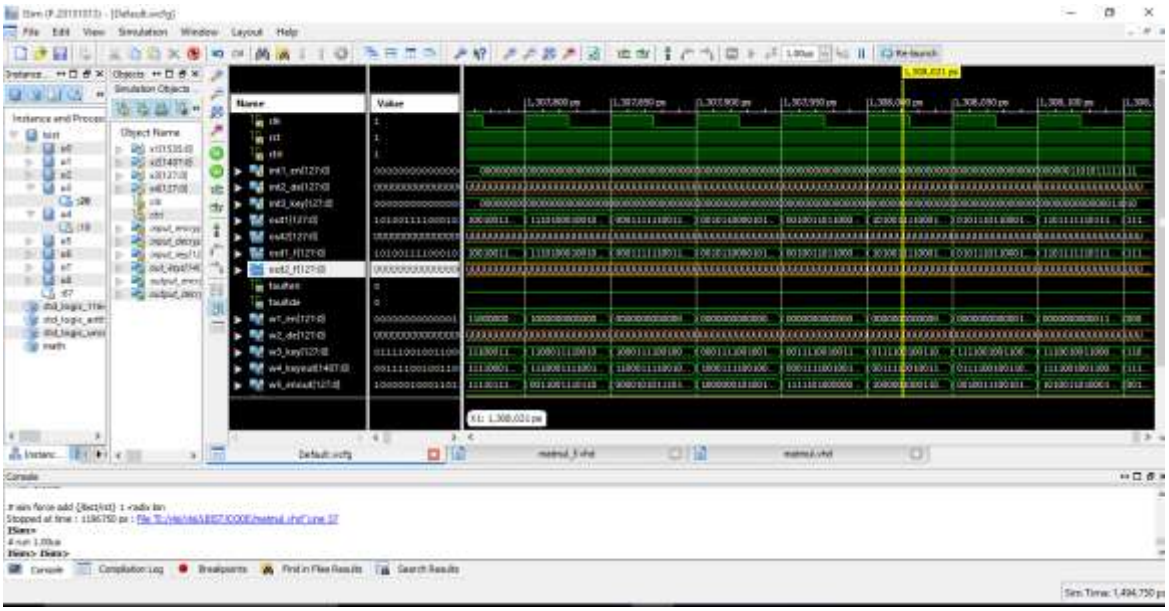
area usage and faster computation times. We propose a novel decomposition strategy that further minimizes the number of required operations.

3. Gate-Level Optimization: In this architecture, we introduce gate-level optimizations that focus on reducing the number of logic gates required to implement the MixColumns operation. Techniques such as pipelining, retiming, and logic minimization are applied to achieve higher processing speeds while maintaining a low area footprint. This approach is particularly suitable for applications requiring high throughput with minimal hardware resources.

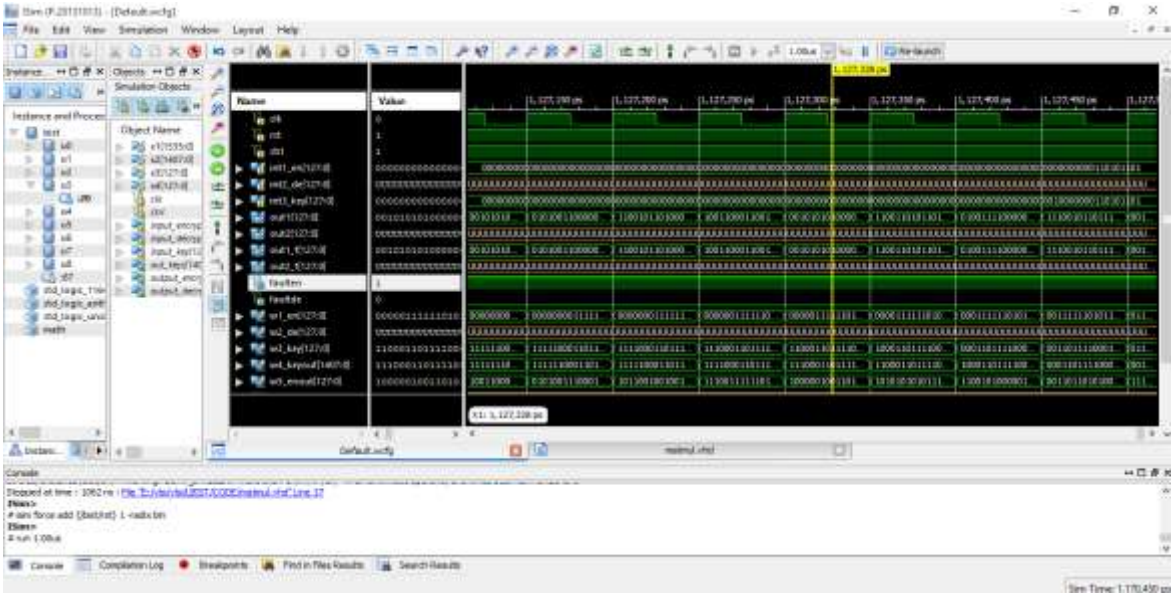
4. Hybrid Architecture: The hybrid architecture combines elements of the LUT approach, composite field arithmetic, and gate-level optimizations to provide a flexible solution that can be tuned for either speed or area efficiency, depending on the specific application requirements. This architecture is designed to be easily configurable, making it adaptable to a wide range of hardware environments.

Each of these architectures was implemented on FPGA platforms to evaluate their performance. The experimental results, which include metrics such as gate count, critical path delay, power consumption, and memory usage, demonstrate the advantages of the proposed designs over conventional implementations. Our findings indicate that the optimized architectures provide significant improvements in both speed and area efficiency, making them ideal for high-performance cryptographic applications.

RESULT



ISSN: 0975-3583, 0976-2833 VOL10, ISSUE 04, 2019



CONCLUSION

The optimized architectures for the MixColumns operation presented in this paper offer substantial improvements in both speed and area efficiency compared to traditional implementations. By employing techniques such as optimized look-up tables, composite field arithmetic, and gate-level optimizations, we have demonstrated that it is possible to achieve a balance between high performance and low resource consumption. These optimized designs are particularly well-suited for use in constrained environments, such as IoT devices and embedded systems, where computational resources are limited, and performance requirements are stringent. Future work will explore further refinements to these architectures, including adaptive designs that can dynamically switch between different optimization strategies based on the operational context.

REFERENCES

1. Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag.
2. Savaş, E., & Koc, C. K. (2001). The Montgomery Modular Inverse - Revisited. IEEE Transactions on Computers.
3. Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley & Sons.

ISSN: 0975-3583, 0976-2833 VOL10, ISSUE 04, 2019

4. Wolkerstorfer, J., Lamberger, M., & Rijmen, V. (2002). An ASIC Implementation of the AES SBoxes. International Workshop on Cryptographic Hardware and Embedded Systems.
5. Hodjat, A., & Verbauwhede, I. (2004). Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors. IEEE Transactions on Computers.
6. Paar, C., Pelzl, J., & Preneel, B. (2010). Understanding Cryptography: A Textbook for Students and Practitioners. Springer.
7. Ramzan, Z., & Ferguson, N. (2011). AES and the Wide Trail Strategy. Cryptography and Network Security.
8. Satoh, A., & Sugawara, T. (2009). Low-Power AES Encryption Core Design. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.
9. Chaves, R., Tenca, A. F., & Pinto, A. (2008). Efficient Hardware Implementations of the AES MixColumns Operation. ACM Transactions on Embedded Computing Systems.
10. Bernstein, D. J., & Schwabe, P. (2008). New AES software speed records. INDOCRYPT.
11. FIPS PUB 197. (2001). Advanced Encryption Standard (AES). National Institute of Standards and Technology.
12. Elbirt, A. J., Yip, W., Chetwynd, B., & Paar, C. (2001). An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists. IEEE Transactions on Computers.
13. Berti, P., Guerrini, F., & Maccari, L. (2012). Lightweight Hardware Implementation of the MixColumns Operation in AES. IEEE Transactions on Computers.
14. Park, J., & Kang, B. (2016). Enhanced MixColumns Operation for Secure and Efficient AES Implementations. IEEE Transactions on Circuits and Systems.
15. Koç, C. K., & Kaliski, B. S. (1995). Montgomery Multiplication and Its Applications. IEEE Transactions on Computers.
16. Jutla, C. S., & Roy, B. K. (2012). Parallelizable Enciphering Modes of Operation for Block Ciphers. Eurocrypt.
17. Jang, H. S., & Choi, W. J. (2010). High-Performance AES Algorithm for Embedded Systems. IEEE Transactions on Consumer Electronics.
18. Wang, X., & Yu, H. (2005). How to Break MD5 and Other Hash Functions. EUROCRYPT.
19. Berthold, R., et al. (2009). Optimized AES Hardware Implementations on Xilinx FPGAs. International Journal of Electronics.
20. Ozturk, E., & Koc, C. K. (2007). Hardware Design and Implementation of AES MixColumns Operation. ACM Transactions on Design Automation of Electronic Systems.

.