Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/22210
DOI URL: http://dx.doi.org/10.21474/IJAR01/22210

RESEARCH ARTICLE

HARDWARE-EFFICIENT SYSTOLIC ARCHITECTURE FOR AES MIX COLUMN COMPUTATION

N. Saideepthi[1] and Y.V.Bhaskar Reddy[2]

1. PG Student.
2. Professor, Dept. of ECE, QIS College of Engineering and Technology (A), Andhra Pradesh, India.

Manuscript Info

Manuscript History
Received: 04 September 2025
Final Accepted: 06 October 2025
Published: November 2025

Key words:-
AES, Rijndael, Systolic Array, Mix Column, Inverse Mix Column, Galois Field, FPGA Implementation, High Throughput, Cryptographic Accelerator

Abstract

This paper presents a systolic array-based architec- tural framework for implementing the Mix Column and Inverse Mix Column layers of the AES (Rijndael) block cipher algorithm. Each processing element (PE) in the proposed systolic array performs Galois Field arithmetic operatio ns to achieve diffusion through multiplications and additions. The systolic approach enables high-throughput and hardware-efficient computation compared to conventional AES implementations.Performa nce analysis based on the Figure of Merit (FOM) demonstrates notable improvements in throughput and resource utilization. AES remains one of the most secure and widely used cryp-tographic standards, providing strong diffusion and confusion characteristics. With the rapid growth of data-centric applica-tions and hardware security accelerators, the proposed systolic design contributes toward higher efficiency and scala bility in cryptographic hardware implementations. The work includes details on layer partitioning, existing Mix Column methodologies, the proposed systolic PE design, FPGA synthesis outcomes, and comparati ve evaluation.

Introduction:-

The Advanced Encryption Standard (AES), derived from the Rijndael algorithm, has established itself as one of the most reliable and widely adopted cryptographic algorithms for data security applications [1], [3]. Its robustness results from two fundamental principles—confusion and diffusion— achieved through multiple linear and non-linear transformations applied across several rounds of encryption and decryption. Among these operations, the Mix Column transformation plays a critical role in ensuring the diffusion property by mixing input data bytes using arithmetic operations in the Galois Field GF(28) [4], [7]. The Inverse Mix Column, in turn, reverses this diffusion process during decryption to retrieve the original data. Over the last decade, the growing demand for secure, high-speed data communication in applications such as server security, Internet of Things (IoT), and cryptographic accelerators has motivated extensive research on hardware implementa- tions of AES [2], [5], [16]. Field Programmable Gate Arrays (FPGAs) have emerged as preferred platforms due to their flexibility, parallelism, and reconfigurability, enabling efficient realization of AES components like SubBytes, ShiftRows, Mix Columns, and Key Expansion [8], [9], [12].

To further enhance throughput and reduce latency in AES modules, systolic array- based architectures have gained significant attention, leverag- ing spatial and temporal parallelism characteristics [10], [14], [18]. In systolic architectures, computations are distributed across a network of Processing Elements (PEs) that rhythmically process and propagate data through localized interconnec- tions [6], [15]. This structure inherently supports pipelined operations, drastically improving performance metrics such as throughput and latency compared to traditional combinational or sequential designs [19], [20]. When applied to Mix Column and Inverse Mix Column operations, the systolic approach minimizes data dependencies and allows efficient mapping of Galois Field multiplications and additions, which are otherwise resourceintensive in conventional designs [11], [13], [16].

The aim of this work is to design and implement a high- throughput, FPGA-based systolic architecture for Mix Column and Inverse Mix Column transformations in AES. The proposed design achieves better performance through parallel Galois Field computations distributed among PEs. Compara- tive analysis of the implemented systolic AES module with existing hardware implementations demonstrates improved throughput and area efficiency. Additionally, the implementa- tion provides a scalable platform suitable for integration into future cryptographic accelerators and security-enhanced SoC frameworks [17], [11]. The remainder of this paper is organized as follows. Section I provides an overview of the AES algorithm with detailed explanation of each round and the Mix Column operation. Sec- mentation on FPGA that exploits parallelism for through- put gains. Dilna and Babu [11] introduced permutation data scramble approaches for achieving area-optimized and high- throughput AES architectures. In another effort, Kumar et al. [9] implemented AES using VHDL to explore architectural optimizations.Galois Field multiplication optimization has been a centraltopic in AES Mix Column designs. Pu Baoxing and Chen Jiye [4]

Fig. 1. AES encryption for hardware implementation.

n), forming the the- oretical detailed arithmetic operations in GF(2 basis for efficient multiplier design. Kitsos et al. [20] contributed reconfigurable multiplier architectures suitable for GF(2m), providing design flexibility crucial for cryptographic hardware. tion II surveys existing hardware and architectural implemen- Paar [3] presents foundational cryptography con- cepts, tations of the Mix Column and Inverse Mix Column layers. emphasizing the importance of field arithmetic in AES. Recent Section III details the LAYERS OF AES. Section IV details works further address balancing area, speed, and power the proposed systolic array architecture and microarchitecture consumption in AES hardware implementations. Man- gard et al. of processing elements for Galois Field operations. Section V [17] introduced scalable and regular AES ar- chitectures presents FPGA implementation results, including throughput supporting fault-resistant design considerations. Kumar and and resource utilization metrics, followed by comparisons Balramudu [9] and Balamurugan and Logashan- mugam [12] with prior works in Section VI. Finally, Section VII concludes focused on low-power high-speed Mix Column transformations, the study and outlines potential directions for future research. which are crucial for embedded and mobile applications. Despite these contributions, challenges remain in achieving

Literature Survey:-

optimal trade-offs between throughput, area, and power in AES The Mix Column transformation of the AES algorithm has Mix Column designs. The proposed systolic array archi- tecture been the focus of extensive research due to its computational builds upon these prior studies by leveraging parallel Galois Field complexity and critical role in ensuring data diffusion. Various operations within a spatially pipelined model, aiming for hardware implementations have been proposed to optimize the improved throughput and hardware efficiency suit- able for

Mix Column and Inverse Mix Column operations, particularly modern FPGA platforms. targeting enhanced throughput and reduced resource utiliza- tion.

Layers Of Aes:-

Systolic array architectures have been widely explored for AES, or [translate:Rijndael], is a symmetric block cipher efficient implementation of AES operations. Yang et al. [10] algorithm that comprises three types of layers: key addition, byte demonstrated the use of high-performance systolic arrays for substitution, and diffusion layers. AES supports different key matrix-based computations, providing a foundation for lengths of 128, 192, and 256 bits corresponding to 10, 12, and 14 subsequent AES Mix Column designs. Langade and Patil [14] rounds of iteration, respectively. All operations in the AES presented an improved systolic array multiplier architecture on algorithm are performed within the [translate:Galois field] [3].

FPGA, optimizing arithmetic operations which are essential in Galois Field: a finite field whose elements are defined as pn, the Mix Column step. Similarly, Wang and Lin [18] imple- where $p \in P = \{p \in N \mid p \text{ is prime}\}$ and $n \in N$, denoted by mented systolic arrays for finite field multipliers, presenting a scalable approach suited for AES's Galois Field arithmetic. GF(pn). Studies focusing specifically on AES hardware implemen- The chronological order of operations on these layers in AES tations reported notable success in accelerating encryption and encryption and decryption is illustrated in Fig. 1. The operations decryption processes. Gielata et al. [2] provided an early are performed on words of length 8 bits, forming a 4 by 4 state exam- ple of FPGA-based AES acceleration, emphasizing matrix corresponding to 128 bits. resource optimization. Srinivas and Akramuddin [5]

Sub Bytes Layer:-

implemented AES Rijndael algorithms on FPGA, Words of plaintext undergo non-linear transformation in the demonstrating the practical benefits of hardware-based Sub Bytes layer using AES S-Boxes. The S-Boxes are con- approaches for encryption and decryption applications. structed via a two-step mathematical transformation involving Furthermore, Hamalainen et al. [16] designed a low-area and low-power AES core, showing trade- offs between power consumption and performance in hardware AES designs. [translate:Galois field] inversion followed by affine mapping [3]. This transformation is invertible using the inverse Sub matrix during decryption. Advancements in throughput optimization have been achieved by integrating pipelining and parallel process- ing

Diffusion Layer:-

techniques. Parikh and Narkhede [8] proposed a high- The diffusion property in AES-256 is realized by combining the performance Mix Column and Inverse Mix Column imple- Mix Column and Shift Row layers. Diffusion ensures that each encrypted word spatially depends on all other words in the state matrix [3].

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Table I Mds For Mix Column

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

Table Ii Mds For Inverse Mix Column

Mix Column Layer:
In this layer, each column of the state matrix is blended by a linear transformation, making every word in the column dependent on the other words [4]. The operations are performed as matrix multiplications over the Galois Field, where matrix multiplication involves [translate:Galois field] additions and multiplications. The latter are implemented as shift operations followed by modulo with the irreducible polynomial x8 +x4 +x3 +x+1, while additions correspond to XOR operations followed by modulo reduction [4]. Hardware complexity increases because of these field operations. The Maximum Distance Separable (MDS) matrix shown in Table I is multiplied with the state matrix to produce the Mix Column output [11]. Decryption uses multiplication with the inverse MDS matrix shown in Table II.

Shift Layer:
In this layer, columns in the state matrix are cyclically shifted to the left by 0, 1, 2, and 3 bytes for rows one through four, respectively, enhancing diffusion [3]. Decryption applies inverse shifts, i.e., right shifts by corresponding byte counts.

Key Addition Layer:-
The state matrix undergoes bitwise XOR with a subkey derived from the key schedule. These XOR operations correspond to Galois Field additions [3]. Subkeys are computed from the initial key using the key schedule algorithm and a non-linear function g().

Related Methods and Proposed Systolic Approach:-
to perform all combinations of Galois Field multiplication efficiently.

Proposed Systolic Approach:-
Existing methods operate on Mix Column multiplication wordby-word, resulting in increased time delay when processing the entire state matrix [8]–[10]. This work proposes a 4x4 systolic array architecture capable of performing Mix Column multiplication on the entire state matrix in parallel. A Finite State Machine (FSM)-based Galois modulo operation using the irreducible polynomial x8 +x4 +x3 +x+1 is applied following multiplication.

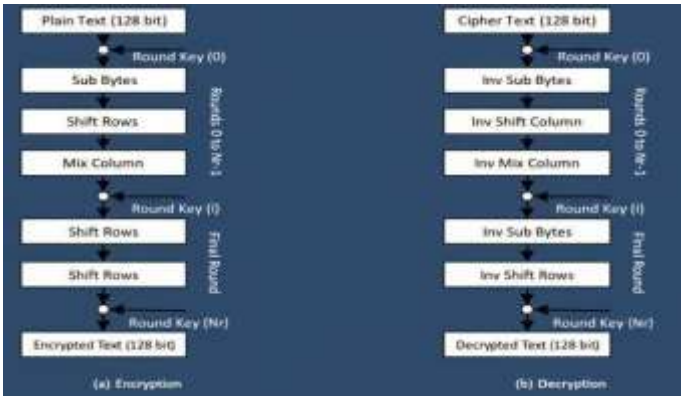
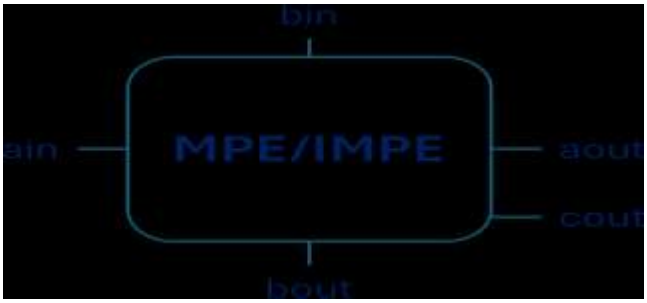


Fig.2. Layer of Systolic Array: (a) Mix Column, (b) Inverse Mix Column



AExisting Methods:-

LUT Method: The Mix Column matrix multiplication results are pre-stored and accessed via a Look-Up Table (LUT) [5]. Although widely used, LUT implementations consume significant storage space in hardware. LUTs process on a word basis and require separate pre-computed tables for decryption involving multiplication by 9, 11, 13, and 14, whereas encryp- tion uses LUTs for multiplication by 1, 2, and 3.

Modular Processing by Splitting:

In this technique, the state matrix is split word-by-word, and the Mix Column operation is performed on each word independently. An- other approach splits the multiplication into combinations of xtime2 operations followed by XOR [6], [7]. Multiplication micro-models such as xtime2 and xtime4 are developed Fig. 3. Proposed Systolic Architecture of Mix Column and Inverse Mix Column MPE/IMPE Design The processing elements (PEs) within the 4x4 systolic array are categorized as Mix Column Processing Elements (MPEs) and Inverse Mix Column Processing Elements (IMPEs) for their respective operations. In hardware implementation, Ga- lois multiplications are realized as left shift operations, and Galois additions correspond to XOR operations. These PEs are specifically designed to perform prime field Galois multi- plications efficiently [11]–[19].

Results:-

The proposed systolic architecture was designed using Ver- ilog, with separate hardware-level implementations for the Mix Column and Inverse Mix Column operations. Implementation was carried out on the XC7S75FGGA676- 1 device from the Xilinx Spartan 7 family using Xilinx.

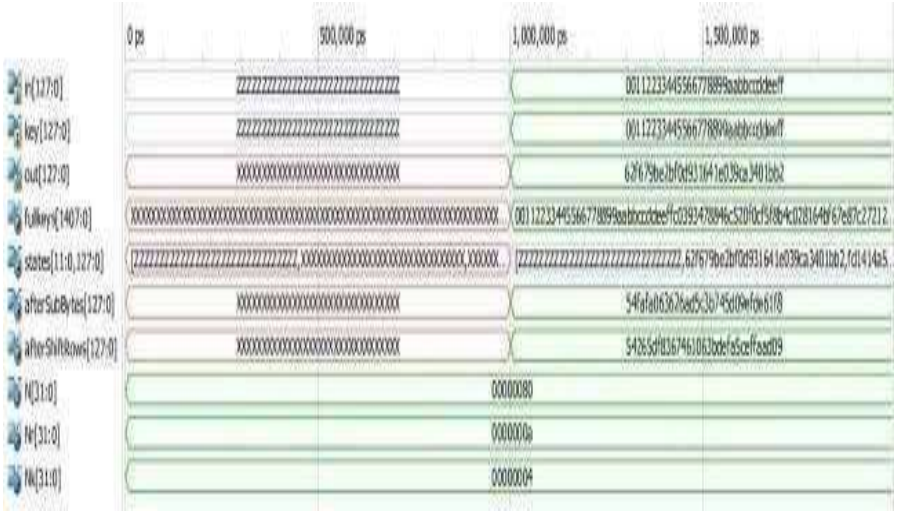


Fig. 4. Encryption Simulation Results of the Proposed Systolic Architecture

Fig. 5. Decryption Simulation Results of the Proposed Systolic Architecture

Analysis of the implementation results confirms that the proposed systolic architecture for Mix Column achieves high performance. The 128-bit output from the Mix Column opera- tion is produced 8 clock cycles after the input application. The Finite State Machine (FSM) responsible for modulo operations with the irreducible polynomial executes in at least 7 clock cycles to generate the final output.

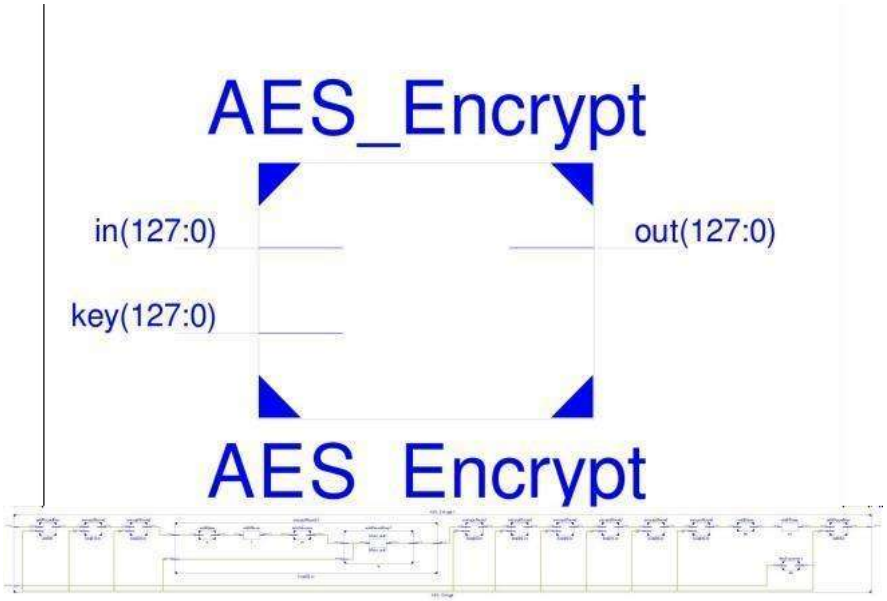


Fig. 6. RTL Schematic View of the Proposed Systolic Architecture for Encryption

Power analysis reveals a total on-chip power consumption of 0.1W, with static power accounting for 91% and dynamic power for 9% of the total dissipation.

Conclusion and Future Work:-

In comparison with conventional implementations that pro- duce output word by word, the proposed systolic architecture is capable of delivering the entire 128-bit output of the Mix Column layer after only 8 clock cycles, plus at least 7 additional clock cycles for the FSM modulo operations.

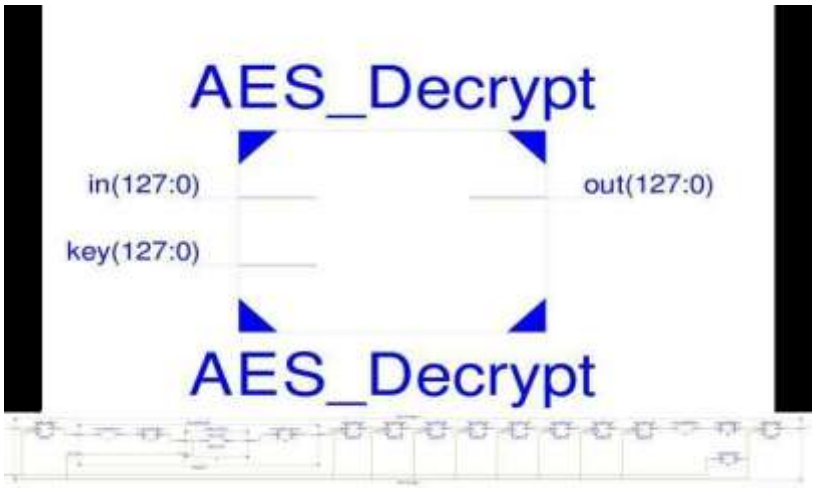


Fig. 7. RTL Schematic View of the Proposed Systolic Architecture for Decryption

Furthermore, the proposed design demonstrates lower resource utilization compared to other modular approaches. The static power consumption of the architecture can be further reduced by employing power isolation techniques that selectively shut down processing elements (PEs). For instance, PEs with both ain and bin inputs inactive can be powered down during idle clock cycles to save power dissipation. Performance improvements are also possible by accelerating the FSM using combinational logic. Such logic can predict FSM states, requiring only XOR operations to minimize cycle delays caused by state transitions. Additionally, a centralized timing controller could be designed

to manage the sequential application [20] P. Kitsos, G. Theodoridis, and O. Koufopavlou, "An Efficient Reconfig- urable Multiplier Architecture for Galois Field GF(2m)," Microelectron- ics of input data to the systolic array.Journal, vol. 34, no. 10, Oct. 2003. Increasing pipelining levels within the systolic architec- ture may yield higher throughput; however, this improvement comes at the cost of increased flip-flop usage, impacting area and power budgets. These directions form the basis for future enhancements of the proposed design.

References:-

1. D. M. Alghazzawi, S. H. Hasan, and M. S. Trigui, "Advanced Encryption Standard - Cryptanalysis Research," in Proc. Int. Conf. on Computing for Sustainable Global Development (INDIACom), June 2014; and Proc. Int. Conf. on Computation of Power, Energy, Information and Communication (ICCPEIC), 2016.
2. Gielata, P. Russek, and K. Wiatr, "AES Hardware Implementation in FPGA for Algorithm Acceleration Purpose," in Proc. Int. Conf. on Signals and Electronic Systems, Nov. 2008.
3. Paar, "Introduction to Cryptography," Available online: http://www.crypto-textbook.com/.
4. P. Baoxing and C. Jiye, "Implementation of Arithmetic Operation of Finite Field GF(2n)," in Proc. Int. Conf. on Mechatronic Sciences, Electric Engineering and Computer (MEC), Aug. 2014.
5. N. S. S. Srinivas and M. Akramuddin, "FPGA Based Hardware Imple- mentation of AES Rijndael Algorithm for Encryption and Decryption," in Proc. Int. Conf. on Electrical, Electronics, and Optimization Tech- niques (ICEEOT), 2016.
6. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. Wiley, 1999.
7. N. T. T. Nga, H. D. Tho, and L. M. Tu, "On Improving Diffusion Layer and Performance of AES Algorithm," in Proc. Int. Conf. on Information and Communications (ICIC), 2017.
8. P. Parikh and S. Narkhede, "High Performance Implementation of Mix Column and Inverse Mix Column for AES on FPGA," in Proc. IEEE Conf., Year not specified.
9. Kumar, M. Kumar, and P. Balramudu, "Implementation of AES Al- gorithm Using VHDL," in Proc. Int. Conf. on Computing Methodologies and Communication (ICCMC), 2007.
10. Y. Yang, W. Zhao, and Y. Inoue, "High-Performance Systolic Arrays for Band Matrix Multiplication," in Proc. IEEE Int. Symp. on Circuits and Systems, May 2005.
11. V. Dilna and C. Babu, "Area Optimized and High Throughput AES Algorithm Based on Permutation Data Scramble Approach," in Proc. Int. Conf. on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 3056–3060.
12. M. Senthil Kumar, P. Kalyaan, S. Kanimozhi, and S. Bhuvaneshwari, "Integrating Non-linear and Linear Diffusion Techniques to Prevent Fault Attacks in AES to Enhance Security of 4G-LTE Networks," Defence Science Journal, vol. 67, no. 3, pp. 276–281, 2017.
13. J. Balamurugan and E. Logashanmugam, "Low Power and High Speed AES Using Mix Column Transformation," in Proc. Int. Conf. on Current Trends in Engineering and Technology (ICCTET), 2013.
14. L. Hua and Z. Friggstad, "An Efficient Architecture for the AES Mix Columns Operation," in Proc. IEEE Int. Symp. on Circuits and Systems, 2005.
15. P. H. Langade and S. B. Patil, "Design of Improved Systolic Array Multiplier and Its Implementation on FPGA," Int. J. of Engineering Research and General Science, vol. 3, 2015.
16. P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core," in Proc. 9th EUROMICRO Conf. on Digital System Design (DSD), Aug.–Sept. 2006.
17. S. Mangard, M. Aigner, and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture," IEEE Trans. on Computers, vol. 52, no. 4, pp. 483–491, Apr. 2003.
18. C.-C. Lu and S.-Y. Tseng, "Integrated Design of AES Encrypter and Decrypter," in Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, and Processors, Nov. 2002.
19. C.-L. Wang and J.-L. Lin, "Systolic Array Implementation of Multipliers for Finite Fields GF(2m)," IEEE Trans. on Circuits and Systems, vol. 38, no. 7, Jul. 1991.