**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

# FPGA Acceleration of AES Algorithm for High-Performance Cryptographic Applications

**Abdullah Farhan Siddiqui[1*], Prof. P. Chandra Sekhar[2]**

*[1*]Student, Department of Electronics and Communication Engineering, Osmania University, Hyderabad, India.*
*[2]Professor, Department of Electronics and Communication Engineering, Osmania University, Hyderabad, India.*

*Corresponding Email: [1*]afs.farhans@gmail.com*

**Received:** 02 February 2024 **Accepted:** 19 April 2024 **Published:** 03 June 2024

***Abstract:*** ***This research paper presents the FPGA implementation of the AES-128 Algorithm as an accelerator tailored for high-performance cryptographic applications. Leveraging the capabilities of the Virtex-7 evaluation kit, the AES algorithm is meticulously coded using Xilinx Vivado software. The results of the implementation reveal a resource-efficient design, utilizing 588 Look-Up Tables (LUTs) and 353 Flip Flops. This implementation showcases the efficacy of FPGA technology, specifically the Virtex-7 device, in achieving a fine balance between algorithmic complexity and resource utilization for cryptographic acceleration. The abstract underscores the significance of this research in advancing the field of hardware-accelerated cryptographic applications, offering a scalable solution with promising resource efficiency on the FPGA platform.***

***Keywords: Field Programmable Gate Array, Advanced Encryption Standard, Look up Table.***

## 1. INTRODUCTION

The field of information security has become increasingly critical with the widespread use of cryptographic algorithms to protect sensitive data and ensure secure communication.
The Internet of Things (IoT) is completely revolutionizing our society by enabling communication anywhere and anytime. Increasingly deployed to enhance public infrastructures and improve citizens' living environments, IoT devices play vital roles in monitoring and controlling the spread of COVID-19. By 2023, the worldwide market for enterprise IoT deployment is estimated to reach a total of 24 billion U.S. dollars, marking a 23% increase from the global market of 19 billion U.S. dollars in 2022. Moreover, the

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

number of IoT device deployments worldwide is expected to reach approximately 27 billion by the end of 2025 and is projected to further increase to approximately 30.2 billion by 2029.

Established by the U.S. National Institute of Standards and Technology (NIST) in 2001, the Advanced Encryption Standard (AES) stands as a cornerstone in contemporary cryptography. It serves as a widely adopted symmetric key encryption algorithm, replacing the aging Data Encryption Standard (DES) due to its vulnerability to brute-force attacks. AES employs a symmetric key, meaning the same key is used for both encryption and decryption, ensuring a high level of security while preserving computational efficiency.

In the realm of contemporary cybersecurity, the importance of hardware acceleration in cryptography stands as a critical facet shaping the landscape of secure data protection. As the volume and complexity of digital information surge, the conventional reliance on software-based cryptographic implementations encounters inherent limitations in speed and computational efficiency. Hardware acceleration, employing specialized processors like cryptographic co-processors or application-specific integrated circuits (ASICs), becomes paramount in surmounting these challenges. This integration not only enhances the swiftness of cryptographic operations but also fortifies overall security by facilitating the implementation of more robust algorithms. In the face of escalating cyber threats, this research paper delves into the pivotal role of hardware acceleration in cryptographic applications, exploring its profound impact on performance, energy efficiency, and resilience against sophisticated adversaries. The examination of this symbiotic relationship aims to contribute valuable insights to the ongoing discourse on fortifying digital systems in an era where the safeguarding of sensitive data is of paramount importance.

The primary purpose of this research paper is to investigate and demonstrate the efficacy of Field-Programmable Gate Array (FPGA) implementation in enhancing the performance of the Advanced Encryption Standard (AES) algorithm. In the landscape of cryptographic applications, where the demand for both security and speed is paramount, FPGA technology holds immense promise due to its reconfigurable nature and parallel processing capabilities. The significance of this research lies in its potential to bridge the gap between cryptographic security and computational efficiency by leveraging the unique features of FPGAs. As the digital world grapples with increasingly sophisticated cyber threats, the exploration of FPGA acceleration for the AES algorithm becomes crucial in designing cryptographic solutions that not only provide robust protection but also meet the high-performance requirements of modern computing environments. The research aims to contribute valuable insights into the practical implementation of FPGA-accelerated AES, offering a nuanced understanding of its advantages and paving the way for the development of high-performance cryptographic systems.

**Background**

The Advanced Encryption Standard (AES) algorithm is a cornerstone in modern symmetric key cryptography. It operates on fixed-size blocks of data, typically 128 bits, and supportskey

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

lengths of 128, 192, or 256 bits. The algorithm comprises several key components that collectively contribute to its security and efficiency.

**Sub Bytes Transformation**: This step involves substituting each byte in the data block with another byte from a predefined substitution table (S-box). The non-linear nature of this substitution provides a crucial layer of confusion in the encryption process.

**Shift Rows Transformation**: In this phase, the rows of the data block are cyclically shifted, ensuring that each byte influences multiple rows. This step contributes to the diffusion of the data, spreading the impact of individual bytes across the entire block.

**Mix Columns Transformation**: MixColumns operates on columns of the data block, linearly combining the bytes within each column. This process further enhances diffusion and ensures that each byte's contribution affects the entire block.

**Add Round Key Transformation**: This step involves bitwise XOR (exclusive OR) of the data block with a round key derived from the original encryption key. The key addition provides a dynamic and key-dependent layer, reinforcing the security of the algorithm.

**Key Expansion**: AES employs a key schedule algorithm to expand the original key into a set of round keys used in successive encryption rounds. This process involves key mixing and substitutions to create a diversified set of keys for each round.
These key components collectively form the basis of the AES algorithm, which has demonstrated resilience against various cryptographic attacks. Understanding the intricacies of these components is essential for the research paper, as it lays the foundation for exploring the subsequent FPGA acceleration of the AES algorithm and its implications for high-performance cryptographic applications.

Field-Programmable Gate Arrays (FPGAs) represent a transformative technology in the realm of hardware acceleration, playing a pivotal role in enhancing the performance of various computational tasks, including cryptographic algorithms like AES. FPGAs are reconfigurable integrated circuits that allow users to customize their digital circuits even after manufacturing. Their unique architecture comprises an array of configurable logic blocks and programmable interconnects, enabling the implementation of bespoke digital circuits tailored to specific applications. The advantages of FPGAs in hardware acceleration are manifold. Unlike fixed-function ASICs, FPGAs offer a high degree of flexibility, enabling rapid prototyping and iterative development. Their parallel processing capabilities allow for concurrent execution of multiple tasks, resulting in significant speedup for applications that demand intensive computation, such as cryptographic algorithms. Additionally, FPGAs are energy-efficient, as they can be optimized for specific algorithms, minimizing power consumption compared to general-purpose processors. This research paper delves into the transformative potential of FPGAs in accelerating the AES algorithm, exploring how their flexibility and parallelism can be harnessed to achieve high-performance cryptographic applications.

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

## 2. RELATED WORK

Several implementations of the Advanced Encryption Standard (AES) Algorithm on various Field Programmable Gate Array (FPGA) devices have been explored, focusing on performance and evaluation metrics. In the study conducted by [1], the AES Algorithm was implemented on the Xilinx XC5VLX50 Virtex-5 FPGA, revealing the utilization of 1338 Slice Look Up Tables (LUTs), 922 flip flops, and a total memory footprint of 20kb. Another investigation, documented in [2], delved into the implementation of AES on the Xilinx Virtex 4 FPGA, showcasing the use of 1468 LUTs, 764 flip flops, and a total memory consumption of 30kb. Moreover, [3] explored the implementation on Xilinx XCV600 Virtex 2.5V FPGA, reporting 3645 LUTs, 512 Flip Flops, and a total memory utilization of 40kb. Notably, these studies provide valuable insights into the specific resource allocations and memory footprints associated with FPGA-based AES implementations, shedding light on the trade-offs between speed and area optimization. The collective findings contribute to the ongoing efforts to enhance the efficiency of AES on FPGA platforms.

## 3. METHODOLOGY

The methodology for this research commences with the Verilog language-based development of code for the Advanced Encryption Standard (AES) Algorithm. The code is organized into modular components, each serving a specific function in the AES encryption process. The primary modules include AES-main, g_function, key_schedule, last_round, mixcolumn, round, sbox, shiftrows, and subbytes, each encapsulating crucial aspects of the algorithm. Subsequently, the development transitions to the creation of test bench modules essential for simulation and result validation. These test bench modules, namely Aes_main_tb, f_function_tb, key_schedule_tb, mixcolumn_tb, shiftrows_tb, and subbytes_tb, facilitate a comprehensive evaluation of the algorithm's functionality using Xilinx Vivado software.
Following the code development and test bench creation, the simulation phase is executed, generating waveforms to assess the performance and correctness of the implemented modules. The next step involves the crucial process of synthesis, wherein the algorithm's code is translated into a hardware description language that lays the foundation for obtaining the schematic representation of the design. This schematic is vital for visualizing the structure and connectivity of the algorithm within the FPGA.
In the final stages of the methodology, the designed algorithm is implemented on the FPGA. This involves the assignment of Input/Output (I/O) ports and pins through the hardware manager, ensuring seamless integration of the algorithm with the FPGA hardware. This comprehensive methodology, spanning code development, simulation, synthesis, and hardware implementation, establishes a robust framework for realizing and evaluating the AES Algorithm on FPGA platforms.

### Aes Overview

The AES Encryption algorithm, also recognized as the Rijndael algorithm, operates as a symmetric block cipher with a fixed block size of 128 bits. Through the utilization of keys spanning 128, 192, and 256 bits, it processes individual blocks of data.

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

Fundamentally grounded on a substitution-permutation network, abbreviated as SP network, AES entails a sequence of interconnected operations. These operations encompass the replacement of input values with designated outputs, termed substitutions, along with manipulations involving the rearrangement of bits, known as permutations. After encrypting these blocks, they are amalgamated to produce the ciphertext.

**Aes: Feautures**

SP Network: Unlike the Feistel cipher structure employed in the DES algorithm, the AES encryption algorithm operates on an SP network configuration.

Key Expansion: Initially, a single key is utilized, which is subsequently expanded into multiple keys employed across individual rounds of encryption.

Byte Data: In the AES encryption process, operations are conducted on byte data rather than bit data. Consequently, the 128-bit block size is interpreted as 16 bytes throughout the encryption operation.

Key Length: The number of encryption rounds executed is contingent upon the length of the key employed. Specifically, ten rounds are performed for a 128-bit key, twelve rounds for a 192-bit key, and fourteen rounds for a 256-bit key.

**Aes Working.**

To grasp the functioning of AES, it's essential to comprehend the flow of information across its various stages. Given that a block consists of 16 bytes, the data is organized within a 4x4 matrix structure, where each cell accommodates one byte of information.

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |

Fig: 1

The depicted matrix is referred to as a state array. Likewise, the initial key undergoes expansion to yield (n+1) keys, where n represents the number of encryption rounds. For

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

instance, with a 128-bit key, comprising 16 rounds, the total number of keys generated amounts to 11, including the original key.

**Aes: Steps**

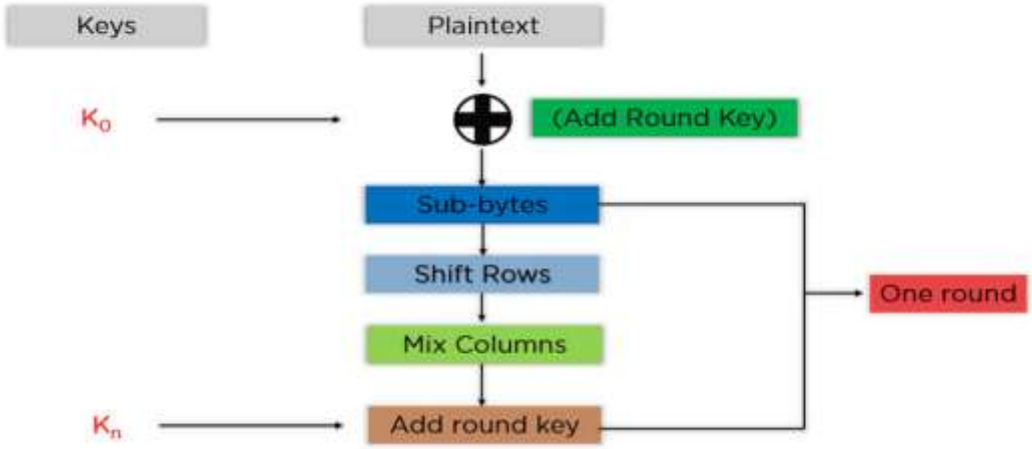


Fig: 2

These outlined steps are executed sequentially for each block. Once the individual blocks are successfully encrypted, they are concatenated to compose the final ciphertext. The process proceeds as follows:

**Add Round Key:** The block data stored in the state array undergoes XOR operation with the first generated key (K0), yielding a resultant state array that serves as input for the subsequent step.

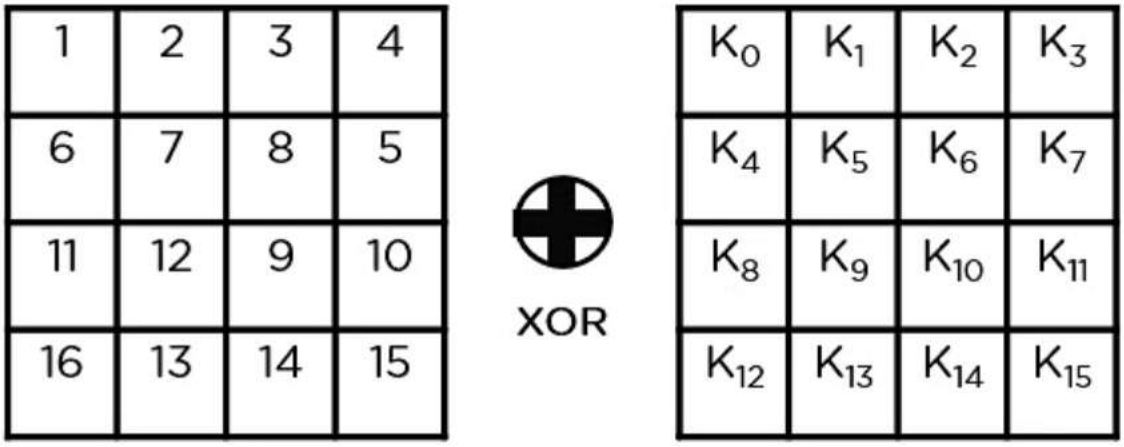


Fig: 3

**Sub-Bytes:** Each byte within the state array is converted into hexadecimal, divided into two halves representing rows and columns, and mapped to a substitution box (S-Box) to generate new values for the updated state array.

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11


Fig: 4

**Shift Rows:** Skipping the first row, row elements are interchanged. The second row shifts one position to the left, the third row shifts two positions to the left, and the last row shifts three positions to the left.

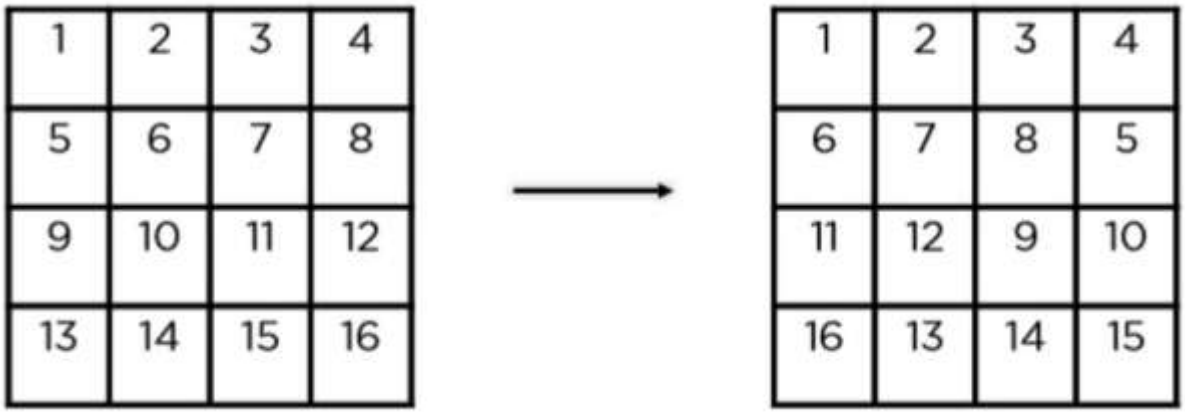


Fig: 5

**Mix Columns:** A constant matrix is multiplied with each column in the state array to produce a new column for the next state array iteration. This step is omitted in the final round.

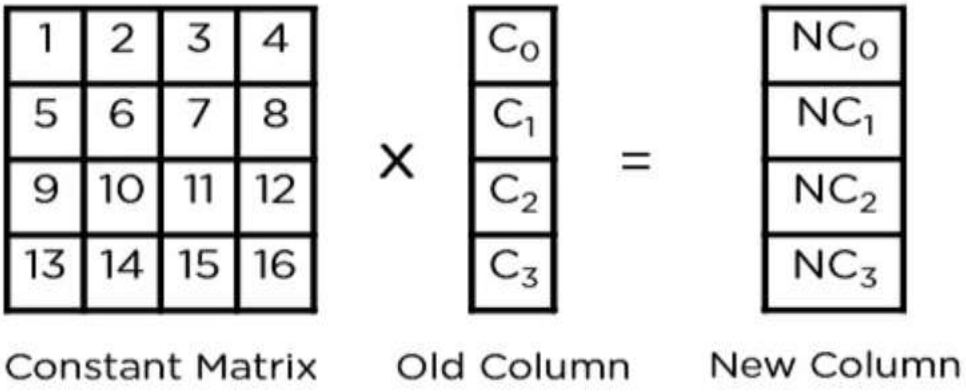


Fig: 5

**Add Round Key:** The resulting state array obtained from the previous step is XORed with the corresponding key for the round. If it's the final round, the resulting state array becomes

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

the ciphertext for the specific block; otherwise, it continues as the new state array input for the subsequent round.

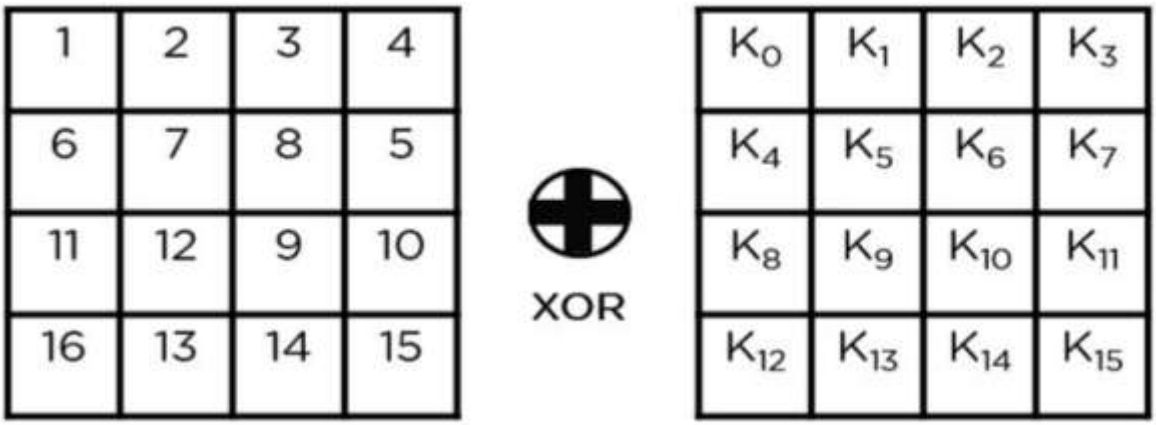


Fig: 6

**Implementation**

After developing the modules for each step of AES Algorithm, the design is simulated and synthesized. The schematic obtained shows the cells and nodes used in by the design.

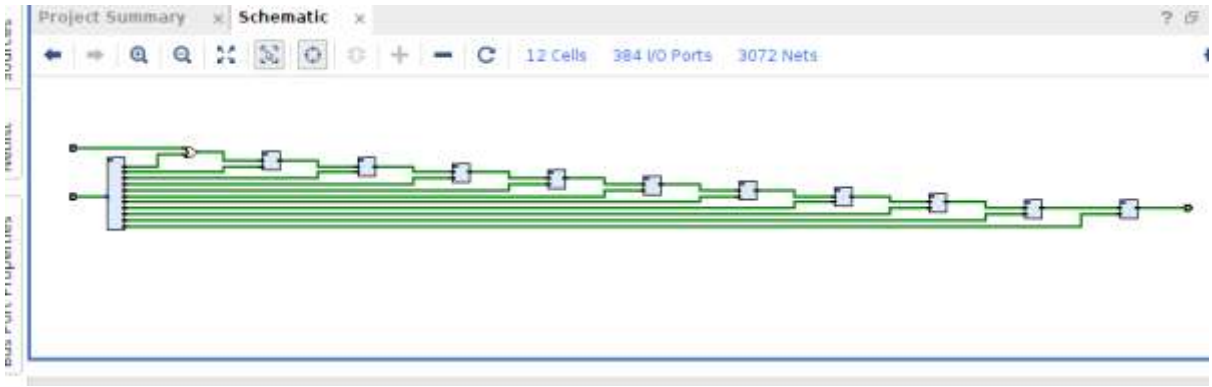


Fig: 6 This schematic is the elaborated design of the developed algorithm.

## 4. RESULT AND DISCUSSION

The result shows the wave forms obtained for plaintext, key and ciphertext.

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

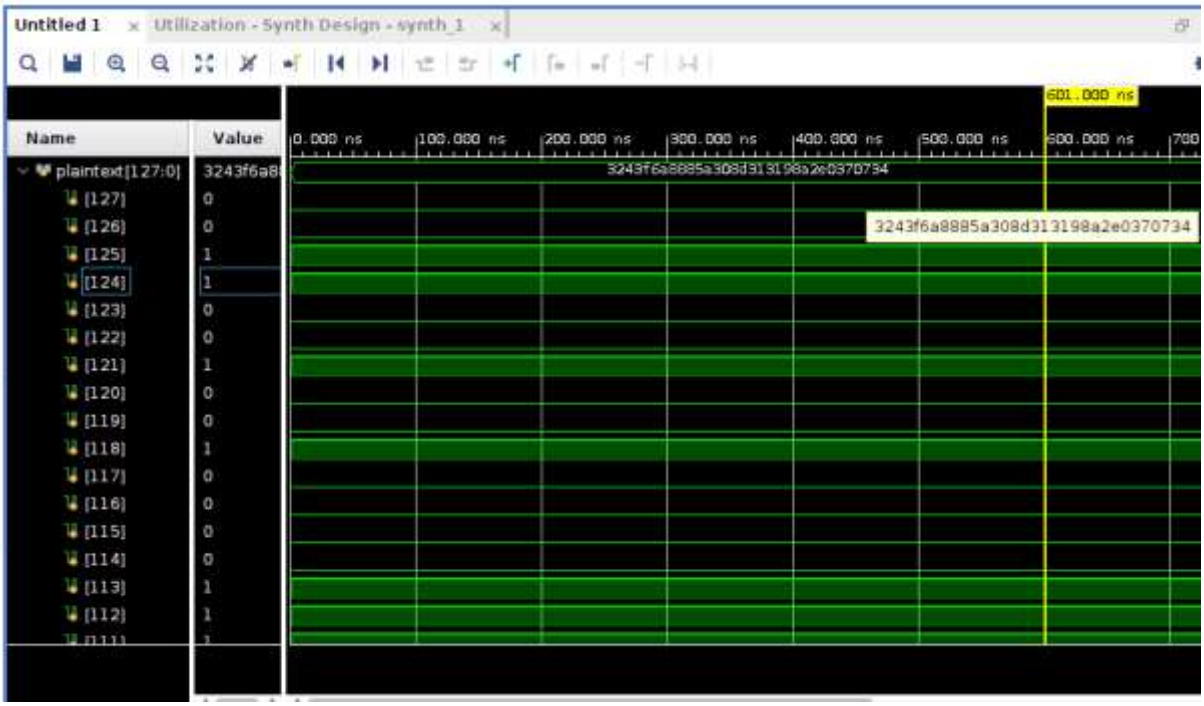

Fig: 7 The waveform of plaintext which shows the value for 128 bits of AES algorithm.

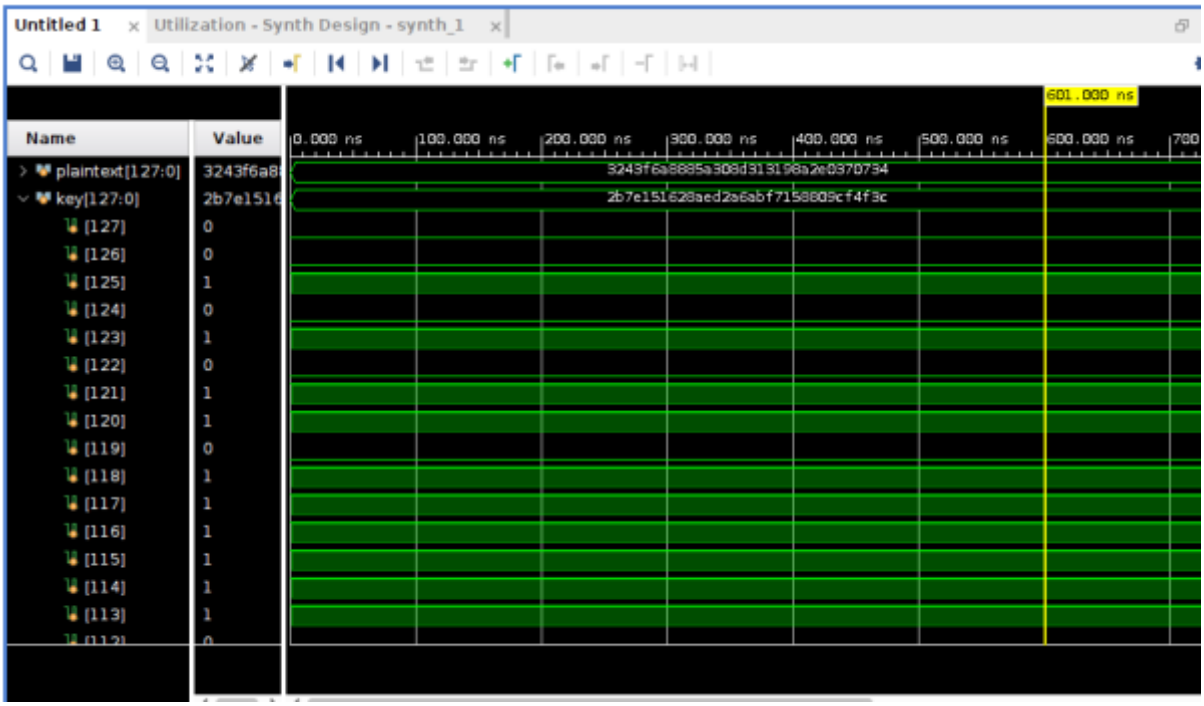

Fig: 8 The waveform of key which shows the value for 128 bits of AES algorithm.

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

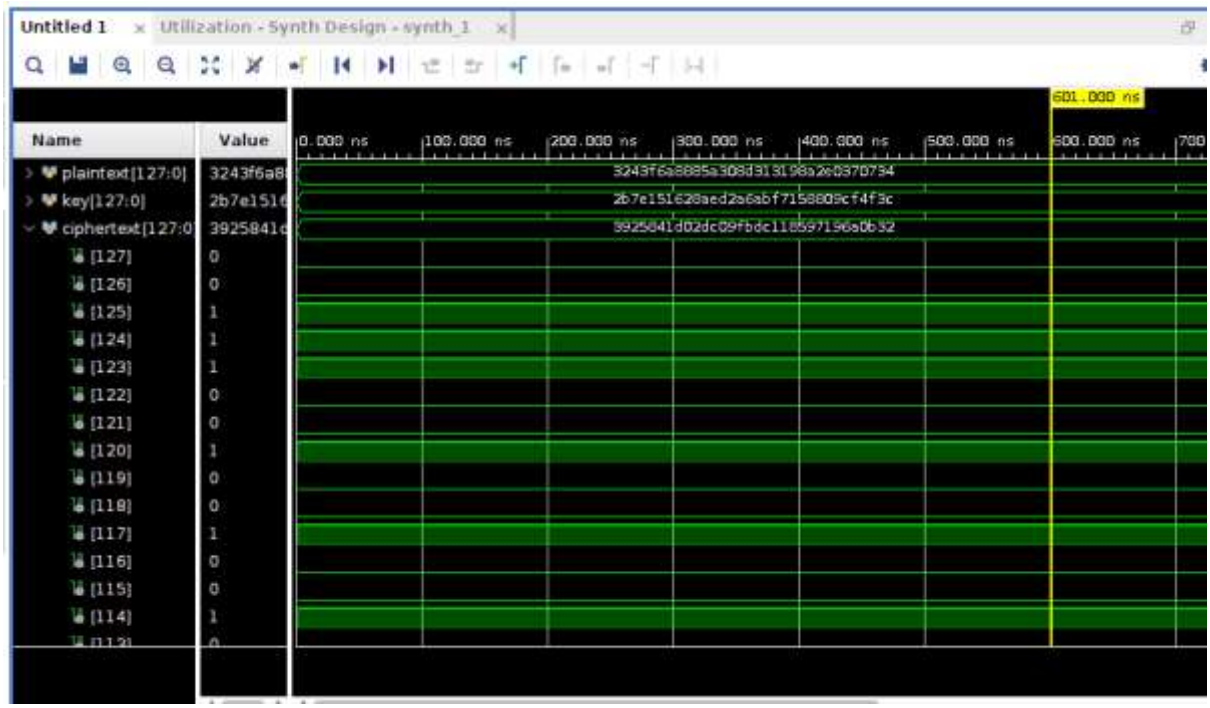


Fig: 9 The waveform of ciphertext which shows the value for 128 bits of AES algorithm.

## 5. CONCLUSION

In conclusion, the development and synthesis of the AES-128 Algorithm accelerator for high-performance cryptographic applications on the FPGA Virtex-7 device yield promising results. The implementation utilizes a resource-efficient configuration, employing 588 Look-Up Tables (LUTs) and 353 Flip Flops. Notably, these resource requirements signify a judicious use of the available hardware, demonstrating an effective balance between algorithm complexity and FPGA resource utilization. The synthesis report further provides insightful metrics, indicating that the accelerator harnesses only a fraction of the available resources on the Virtex-7 device, with 303,600 LUTs and 607,200 Flip Flops at its disposal. This underscores the scalability and efficiency of the developed AES-128 accelerator, affirming its potential for integration into high-performance cryptographic systems while leaving substantial resources for additional functionalities. The results showcase not only the successful realization of the accelerator but also its adaptability to the capabilities of the Virtex-7 FPGA, offering a promising avenue for further exploration and optimization in the realm of hardware-accelerated cryptographic applications.

## 6. REFERENCE

1. S. Suhaili, Rene Brooke Fredrick, Zainah Binti Md. Zain, N. Julai, Design and Implementation of Advanced Encryption Standard Using Verilog HDL, 2021, Lecture Notes in Electrical Engineering.

**International Journal of Information technology and Computer Engineering**
**ISSN: 2455-5290**
Vol: 04, No.04, June-July 2024
http://journal.hmjournals.com/index.php/IJITC
**DOI:** https://doi.org/10.55529/ijitc.44.1.11

2. Sunil, J., S, S. H., K, S. B., & Santhameena, S. (2020). Implementation of AES Algorithm on FPGA and on software. 2020 IEEE International Conference for Innovation in Technology (INOCON).
3. G. Shet, Jamuna V, Shravani S, Nayana H G, Pramod Kumar S, Implementation of AES Algorithm using Verilog, 2020, JNNCE Journal of Engineering and Management.
4. Jonwal, S. U., & Shingare, P. P. (2017). Advanced Encryption Standard (AES) implementation on FPGA with hardware in loop. 2017 International Conference on Trends in Electronics and Informatics (ICEI).
5. Sheetal U. Jonwal, P. Shingare, Advanced Encryption Standard (AES) implementation on FPGA with hardware in loop, 2017 International Conference on Trends in Electronics and Informatics (ICEI).
6. N. Shaji, P. L. Bonifus, Design of AES Architecture with Area and Speed Tradeoff, 2016, Procedia Technology.
7. N. Shaji, L. BonifusP., Area Optimized Architecture for AES Mix Column Operation, 2015, International journal of engineering research and technology.
8. Rao, M., Newe, T., & Grout, I. (2015). AES implementation on Xilinx FPGAs suitable for FPGA based WBSNs. 2015 9th International Conference on Sensing Technology (ICST).
9. El Maraghy, M., Hesham, S., & Abd El Ghany, M. A. (2013). Real-time efficient FPGA implementation of aes algorithm. 2013 IEEE International SOC Conference.
10. Rizk, M. R. M., & Morsy, M. (2007). Optimized Area and Optimized Speed Hardware Implementations of AES on FPGA. 2007 2nd International Design and Test Workshop.