International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 4 (2025)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 4, 2025

Peer Reviewed, Referred & Indexed Journal

Research Paper

DESIGN AND ANALYSIS OF AES CRYPTOGRAPHIC CORE USING VERILOG HDL

Gowni Sai Mounika[1], Mohana Dandi[2]

[1]M. Tech Student, Department of Electronics and Communication Engineering, Golden Valley Integrated Campus, Madanapalle, Annammayya District, Andhra Pradesh 517352

[2]Assistant professor, Department of Electronics and Communication Engineering, Golden Valley Integrated Campus, Madanapalle, Annammayya District, Andhra Pradesh 517352

Abstract *The Advanced Encryption Standard (AES) is a symmetric block cipher algorithm widely used for securing digital communication and data storage. This project focuses on the design and implementation of AES using Verilog HDL. The AES algorithm operates on 128-bit data blocks with variable key lengths of 128, 192, or 256 bits. The design includes the essential operations of Sub Bytes, Shift Rows, Mix Columns, and Add RoundKey, executed across multiple rounds depending on the key size. Verilog HDL is used to model the internal architecture at the Register Transfer Level (RTL) for improved modularity and synthesis efficiency. Functional verification is performed using simulation tools like Xilinx ISE to validate encryption and decryption processes. The proposed design achieves optimized speed, area, and power consumption, making it suitable for real-time security applications. The hardware-based AES design offers enhanced data protection for communication networks, IoT devices, and embedded systems. Overall, the work demonstrates that Verilog HDL provides a flexible and reliable platform for implementing cryptographic algorithms in modern digital security solutions.*

Received: 05-11-2025 Accepted: 15-12-2025 Published: 22-12-2025

1. Introduction

In this research work, we propose the design and implementation of a real-time FPGA based application, which demonstrates the creation of real-time process tasks in FPGA systems for successful real-time communication between multiple FPGA systems. We have chosen the RSA based encryption and decryption algorithm for this implementation, as security is one of the most important need for data communication. The recent development of Field-Programmable Gate Array (FPGA) architectures, with soft core (Micro Blaze) and hard core (PowerPC) processors, embedded memories and IP cores, offers the potential for high computing power. Presently FPGAs are considered as a major platform for high performance embedded applications as it provides the opportunity for reconfiguration as well as good clock speed and design resources.

As the complexities in the embedded applications increase, use of an operating system brings in a lot of advantages. In present day application scenarios most embedded systems have real-time requirements that demand the use of Real-time operating systems (RTOS), which creates a suitable environment for real time applications to be designed and expanded easily. In an RTOS the design process is simplified by splitting the application code into separate tasks and then the scheduler executes them according to a specific schedule, meeting the real-time deadline. In this research work, we propose the design and implementation of a real-time FPGA based application, which demonstrates the creation of real-time process tasks in FPGA systems for successful

real-time communication between multiple FPGA systems.

We have chosen the RSA based encryption and decryption algorithm for this implementation, as security is one of the most important need for data communication. At first we demonstrate the realtime execution of multiple process tasks in a single FPGA system for the encryption and decryption of data. Next we describe the most challenging part of our work, where we establish the realtime communication between two FPGA systems, each running the encryption engine and decryption engine respectively and communicating with one another via an RS232 communication link. The results show that our design is better in terms of execution speed in comparison with the existing research works.

At first we demonstrate the real time execution of multiple process tasks in a single FPGA system for the encryption and decryption of data. Next we describe the most challenging part of our work, where we establish the real time communication between two FPGA systems, each running the encryption engine and decryption engine respectively and communicating with one another via an RS232 communication link. The results show that our design is better in terms of execution speed in comparison with the existing research works.It achieves the real time secured information between the systems implemented in multiple FPGA's by using RTOS (Real Time Operating System). This information sharing is based on RSA algorithm (encryption and decryption). Very large Scale Integrations in The Recent trends of design. Network Security in the Techniques of Very large Scale Integrations Plays Very Vital Role. FPGA, logic circuits, operating systems (computers), Micro Blaze FPGA architectures, embedded memory, multiple FPGA systems and soft core processors.

2. Literature reviews

Mayada E. Mohamed, Sharief F. Babiker, 2022

This paper presents two hardware architectures for implementing AES-128 encryption on FPGA, written in Verilog. One is an iterative (basic) architecture optimized for low FPGA resource usage (uses 347 slices and 10 BRAMs) with moderate throughput (~1.3988 Gbps), and the other is a fully pipelined architecture aimed at very high speed (achieving ~31.4574 Gbps) using more resources (30 BRAMs).

Alexander Owusu-Ansah &Kwangki Ryoo, 2018

This work proposes an AES core (128/192/256-bit key sizes) implemented in Verilog HDL with emphasis on balancing throughput and area. Key features include a four-stage sub-pipelined architecture, on-the-fly key generation, and LUT-based S-Box implementations. They synthesize the design using a 180nm TSMC process. The area is reported in terms of NAND2 equivalent gates, with about 21.3K gates for the key generation unit.

3. Existed system

In the existing system, most encryption and decryption processes are implemented through software algorithms running on general-purpose processors. These software-based AES implementations are often written in high-level languages like C or Java and executed sequentially, which limits their performance when handling large volumes of data. The software approach consumes more CPU cycles and power, making it unsuitable for real-time or embedded applications where speed and energy efficiency are critical. Additionally, software implementations are more vulnerable to side-channel attacks due to predictable execution patterns. Hardware-based designs have emerged to overcome these limitations, but many traditional FPGA or ASIC AES cores still suffer from trade-offs between area, speed, and power. Conventional architectures often use lookup tables (LUTs) for S-box generation, which occupy more hardware resources. Moreover, these designs lack flexibility in key expansion and often do not fully exploit parallelism inherent in AES operations. Some implementations also fail to achieve optimal pipelining or reuse of intermediate values, resulting in lower throughput. Existing designs may not be suitable for compact IoT or edge devices due to high gate count and latency. Hence, there is a need for an optimized AES architecture implemented in Verilog HDL that can provide high-speed encryption, low area utilization, and low power consumption while maintaining standard compliance and strong security.

4. Methodology used

Our project initially aims at understanding a conventional cryptographic standard known as Advanced Encryption Standard (AES), which is the most-sought after secret-key security algorithm that is to be effectively employed in the future for the greatest security deals, in its various forms, namely simple AES (128 bit key and plain-text), APES (512 bit data using parallel 128 bit AES), ADES (512 bit data using 64 bit DES and 128 bit AES), etc., all of which have been opened up by the tremendous growth and significant breakthroughs in the recent history of conventional cryptography. Later, the Advanced Encryption Standard is discussed in all its mathematical preliminaries and scientific depiction of the approved algorithm. After this, the limitations in software implementation are analyzed and the various hardware approaches are studied exhaustively.

A highly parallelized and low cost hardware architectural solution is proposed based on relative merits of FPGA architecture; the architectural details and the functionality are fully elucidated in its top-to-bottom modular hierarchy; simulated using the Mentor Graphics VHDL simulator, ModelSim XE II v5.8C, synthesized using the Xilinx Synthesis tool, Xilinx ISE 6 (Integrated Software Environment), and finally configured and implemented in FPGA using SANDS FPGA/CPLD development platform.

Thus, in this project we ultimately aim at developing a cost-effective but highly secure and parallelized solution for implementing the AES algorithm in hardware, by effectively integrating the potential advantages, major capabilities and micro compactness of the VLSI to revolutionize the major area of Secure Data Communications through Computers Networks, which is now-a-days a major concern not only to the giant federal organizations but also to the private individuals in this strategic world.

5. Proposed system (AES Algorithm)

AES as well as most encryption algorithms is reversible. This means that almost the same steps are performed to complete both encryption and decryption in reverse order. The AES algorithm operates on bytes, which makes it simpler to implement and explain. This key is expanded into individual sub keys, a sub keys for each operation round. This process is called KEY EXPANSION, which is described at the end of this document.

For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations:

Byte substitution using a substitution table (S-box)

1. Shifting rows of the State array by different offsets,
2. Mixing the data within each column of the State array, and
3. Adding a Round Key to the State. As mentioned before AES is an iterated block cipher.
4. All that means is that the same operations are performed many times on a fixed number of bytes.

5. Results and analysis

Simulation result of encryption:
Input:
in=128'h54776F204F6E65204E696E652054776F
clock=1
key=128'h5468617473206D79204B756E67204675
Output:
cipher =
128'h29C3505F571420F6402299B31A02D73A

Fig 1: Encryption output

Simulation result of decryption:
Inputs:
in=128'h29C3505F571420F6402299B31A02D73A
clock=1
key=128'h5468617473206D79204B756E67204675
Output:
plain txt=128'h4120452053616c 676f726974686d0d 0a

Fig 2: Decryption Output

6. Conclusions

The design and implementation of AES using Verilog HDL provide an efficient, secure, and high-performance cryptographic solution for modern digital systems. By modeling the architecture at the RTL level, the design achieves a perfect balance between speed, area, and power optimization. Simulation and synthesis confirm that the hardware-based approach outperforms traditional software implementations in both throughput and energy efficiency. Moreover, the system ensures reliable data confidentiality, integrity, and authentication, making it highly suitable for real-time and embedded applications. Overall, this project demonstrates that Verilog HDL serves as an excellent tool for implementing AES, contributing to the development of secure and efficient digital communication systems.

REFERENCES

[1] S. Sau , C. Pal and A Chakrabarti "Design and Implementation of Real Time Secured RS232 Link for Multiple FPGA Communication, Proc. Of International Conference on Communication, Computing & Security",2011, ISBN - 978- 1-4503-0464- 1.

[2] C. D. Walter. August 1999. Montgomery's Multiplication Technique: "How to Make It Smaller and Faster. Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science", Springer.No. 17 17. pp. 80-93.

[3] A Mazzeo, L. Romano, G. P. Saggese and N. Mazzocca. 2003. "FPGABased Implementation of a Serial RSA Processor". Design. Proceedings of the conference on Design, Automation and Test in Europe - Volume I. ISBN:O- 7695-1870-2 .

[4] xilkernel_v3.00.pdf on *www.xilinx .com.*

[5] R. L. Rivest et al. 1978. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM". Vol. 2 1. pp. 120- 126.

[6] "Cryptography & Network Security ByBehrouzAForouzan".

[7] "Montgomery Algorithm for Modular Multiplication Professor Dr. D. J. Guan" ,August 25, 2003.

[8] "RSA & Public Key Cryptography in FPGAs, John Fry, Martin Langhammer Altera Corporation" - Europe

[9] "A. Tenca, C. Koc. 1999. A Scalable Architecture for Montgomery Multiplication. Cryptographic Hardware and Embedded Systems", Lecture Notes in Computer Science, No. 17 17, pp. 94- 108.

[10]. "A. Tenca, G. Todorov, C. Koc. May 200 1.High-radix design of a scalable modular multiplier. Cryptographic Hardware and Embedded Systems", Lecture Notes in Computer Science, Springer. No. 2 162.pp. 185-20 1. [II] High-Speed RSA Implementation, Cetin Kaya Koc, November 1994, Version 2.0, ftp://ftp.rsa.comlpub/pdfs/tr20I.pdf.

[12]*http://csrc.nist.gov/publications/fips/fipsI97Ifips-197.pdf.*

[13] *http://www.design-reuse.comlarticlesIl3981 /fpga-implementation-ofaes- encryptio n-and-decryptio n html.*

[14]B. Schneier. 1996. "Applied Cryptography, Protocols, Algorithms, and Source Code in C",

John Wiley and Sons Inc. 2nd Edition. New York, U.S.A.