Design and Feasibility Analysis of a UART Core Based on the AES Algorithm

Yuzhi Huang*

Department of Microelectronics, Tianjin University, Tianjin, China

*Corresponding author: 3023232030@tju.edu.cn

Abstract. This paper designs a Universal Asynchronous Receiver/Transmitter (UART) core integrated with an Advanced Encryption Standard (AES) algorithm encryption module, aiming to enhance the security of serial communication. By adopting the AES symmetric encryption algorithm, the UART core serves as the fundamental communication module, responsible for serializing and parallelizing data, while the encryption module performs operations such as XOR and byte substitution based on the principles of the AES algorithm. These modules not only ensure that the data is not distorted but also improve data security during transmission. With these measures, the system achieves real-time encryption and decryption of transmitted data. Then, this system successfully completes the entire process from plain-text data input and encrypted data transmission, after which it comes to reception and decryption through testbench verification. The test results demonstrate that the system can effectively improve the security of serial communication and hold broad potential and space for development in embedded systems.

Keywords: UART; AES algorithm; serial communication; data security; hardware description language.

1. Introduction

UART is an asynchronous serial communication protocol widely used in embedded systems and microcontrollers, known for its ease of hardware implementation and robust communication capabilities. However, traditional UART communication lacks built-in data security measures, a flaw that has become increasingly evident in the current era. With the rising frequency of modern cyberattacks, the importance of data encryption and privacy protection has become significantly prominent across various industries. This is especially critical in sensitive fields such as financial services, industrial control, smart homes, and medical devices, where the integrity and confidentiality of information are paramount. Data breaches in these areas could severely impact personal privacy, corporate core databases, and national economic security [1]. Therefore, designing a UART core integrated with an encryption module holds exceptional practical value, as it enhances data transmission security without compromising communication efficiency or incurring excessive costs.

In recent years, with the advancement of information technology, secure data transmission has become a key design consideration. Strengthening security inevitably involves cryptography, which enables the storage or transmission of sensitive information in a manner that unauthorized personnel cannot decipher [2]. Large-scale data streams can be securely exchanged in this encrypted form. To achieve cryptographic encoding, reprogrammable devices such as Field-Programmable Gate Arrays (FPGAs) offer a powerful solution. Consequently, much recent research has focused on hardware implementations of encryption algorithms to enhance overall system security [3]. However, as integration scales continue to grow, substantial resource consumption and memory usage have become critical factors that must be considered. Overly complex or energy-intensive encryption modules can significantly degrade the performance of large-scale integrated circuits. By adopting lightweight encryption algorithms such as AES-128, secure communication channels can be achieved at a relatively low cost.

This paper aims to implement an AES encryption module integrated with a UART core in hardware and analyze its performance and reliability in practical applications.

2. Preliminary

2.1. UART Recommendation

UART is an asynchronous serial communication protocol [4], which transmits and receives data by combining starting bits, data bits, parity bits, and stopping bits [5]. Its main characteristic is its simple hardware structure, requiring only TX and RX signal lines to complete communication. Furthermore, UART operates based on an asynchronous communication mechanism, eliminating the need for a shared clock line; the transmitter and receiver only need to use the same baud rate for synchronized operation. Due to its simple design and ease of maintenance, UART is widely used in embedded systems, such as microcontrollers, sensor nodes, and industrial control devices.

Although UART holds advantages in communication efficiency and implementation cost, it lacks a data encryption mechanism, posing significant security risks in applications sensitive to data. In the era of big data and against the backdrop of growing demand for secure communication, much research has been dedicated to enhancing UART security. One approach is software-based encryption, where encryption and decryption are performed using software algorithms at the transmitter and receiver. This method offers high flexibility and facilitates switching between different encryption algorithms. However, it demands higher processor performance and may introduce significant latency in resource-constrained embedded systems. Another approach is hardware-based encryption, which integrates an encryption module within the UART core to enable real-time encryption and decryption. This method provides high efficiency, low power consumption, and stronger security, but it increases design complexity and requires careful coordination of timing and logic with the UART module. A third approach adopts a "hardware + software" hybrid method, where the hardware handles core encryption operations, and the software manages protocol handling and key updates. This approach offers a solution that balances performance and flexibility and is currently regarded as one of the most effective methods in encryption strategies.

In summary, integrating an encryption module into the UART core to achieve hardware-level encryption is a key direction for enhancing data security. The introduction of a hardware encryption module can significantly improve efficiency and security, with this advantage being particularly evident in resource-constrained embedded systems [6].

2.2. AES Encryption Algorithm Introduction

This study primarily employs the typical symmetric encryption algorithm, AES, to construct the encryption module. Symmetric encryption algorithms, which use the same key for both encryption and decryption, are well-suited for hardware implementation, enabling fast computation while reducing resource consumption [7]. Currently, the Advanced Encryption Standard (AES) is the most widely used symmetric encryption algorithm, actively applied in various types of secure communication systems. AES supports key lengths of 128, 192, or 256 bits, and uses these keys to process plaintext, significantly enhancing ciphertext security. The fundamental steps of its multi-round encryption operations include: SubBytes, Shift Rows, Mix Columns, and Add Round Key. Among these, the S-box (SubBytes) is the core step of the AES algorithm, responsible for non-linearly substituting bytes to strengthen the algorithm's resistance against external attacks [8]. This step, along with its inverse (InvSubBytes), can be implemented using Look-Up Tables (LUTs) [9]. In this study, hardware description languages (such as Verilog or VHDL) are used to implement the AES encryption module, which not only significantly reduces encryption latency but also facilitates the replication of the encryption principles.

Various hardware architectures for AES have been explored in the literature, including pipeline structures, parallel structures, area-optimized structures, and high-throughput/gate Feistel Network (FN) AES-OTR hardware architectures [10], making them suitable for different application scenarios.

3. Searching Methods

To investigate the feasibility and reliability of implementing an encryption algorithm at the link layer of a UART IP core, this experiment mainly designs a simplified UART core and an encryption module based on the AES algorithm's S-box for encryption operations. The XOR algorithm is utilized to enhance the rigor and uniqueness of the S-box. Correspondingly, an inv_sbox library is implemented in the decryption module to ensure that the encrypted data remains intact and not distorted after decryption. All of these operations are conducted at the hardware level, primarily using the ModelSim platform and Verilog language for behavioral simulation.

As shown in Figure 1, the main module (top module) consists of five submodules: the baud rate generator module, the encryption module, the decryption module, the transmitter module (tx), and the receiver module (rx).

The external input signals include the clock signal (clk), reset signal (rst), data to be transmitted (data_in), and a start signal (tx_start). The parallel data to be transmitted enters the encryption module first. After encryption, it is passed to the transmitter module, which converts it into a serial signal when the start signal is active. This serial data is then transmitted to the receiver module at the same baud rate. Once the transmission is complete, the receiver module sends the data in parallel to the decryption module. After this entire process is finished, both the transmitter and receiver modules output a completion signal (done). Finally, the decryption module processes the data and outputs the decrypted result (data_out).

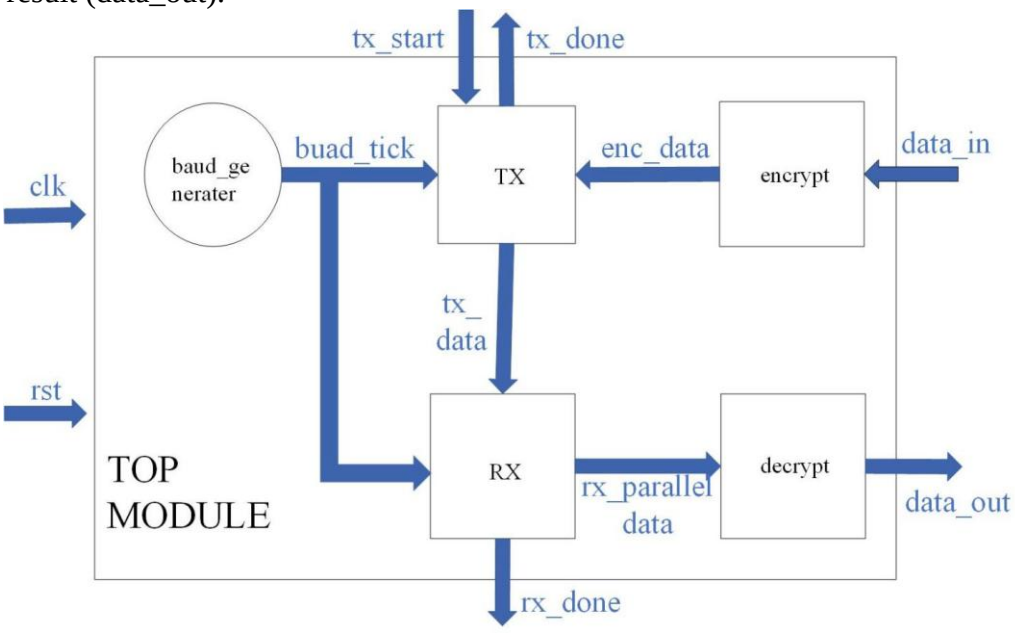

Fig. 1 the Top Module of a UART core with the AES algorithm module

During the code development process, to address the issue of mismatched data widths and types between different modules, multiple intermediate variables of register type were used within each module to adjust the data width accordingly. At the same time, the assignment algorithm in the receiver module was improved to prevent incorrect or missed assignments of tx_data.

4. Experiment Design

The ultimate goal of this project is to explore the feasibility and practicality of implementing an encryption algorithm on a UART core. Therefore, a simplified UART core and a behavior simulation experiment based on a simplified AES algorithm were designed. By comparing the data deviation before and after transmission and analyzing the encrypted data, the aim is to determine whether data distortion occurs before and after encryption and decryption, as well as to assess how clearly the encryption performance is reflected at the link layer.

In the baud rate generator module, this project utilizes a four-stage counter divider to achieve frequency division through a state machine approach. The divided signal is then sent to both the transmitter and receiver modules to ensure that the baud rate is consistent.

In the encryption module, an XOR algorithm is applied to replace the original data (which in this experiment is an 8-bit binary number) with data from the sbox, thereby achieving an encryption effect. The format of the data is preserved during this process, allowing it to be serialized by the transmitter module.

The transmitter and receiver modules are also implemented using state machines. When the reset signal is active, the initial state is set. Upon the falling edge of the reset signal and when the start signal is active, the system enters the data reading state. In this state, the transmitter module stores the data into a register and sequentially loads it into tx_data at the generated baud rate. Meanwhile, the receiver module reads the tx_data at the same frequency, sequentially storing each bit into a register. Finally, this register is assigned to rx_parallel_data, converting the data back into a parallel format.

The decryption module employs the inverse sbox (inv_sbox), which is derived by inverting the operation of the sbox. It takes the rx_parallel_data from the receiver module as input and produces the final output data.

5. Results And Analysis

As shown in Figure 2, Figure 3 and Figure 4, the data_out matches the data_in exactly, indicating that no distortion occurred during the encryption and decryption process. The encrypted data ec_data has no linear relationship with the original data data_in, which demonstrates good confidentiality performance and makes it less susceptible to attacks or tampering. At the same time, the UART core operates normally, and tx_data is fully output.

The initial baud rate was set to 25,000,000 Bd, and the total transmission time was 385 ns. After changing the baud rate to 12,500,000 Bd, the total transmission time became 745 ns. Finally, when the baud rate was adjusted to 8,333,333 Bd, the transmission time increased to 1,105 ns. The tx_data enters the preparation state starting at 25 ns and transitions into the data reading state after one symbol period. Calculations confirm that the data input and transmission process is only related to the baud rate and not affected by the encryption or decryption modules. This indicates that the modules operate independently with no interference from irrelevant variables.

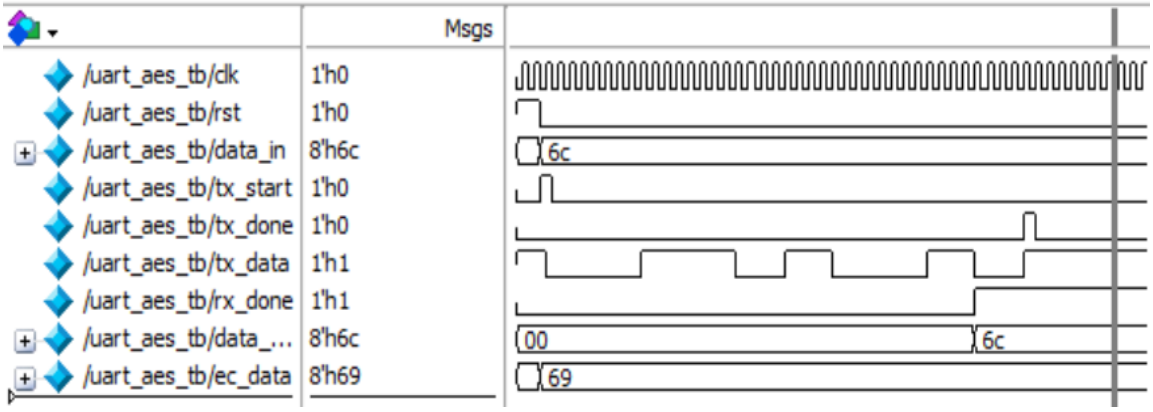

Fig. 2 When data_in is 6c and it is XOR'd with 05, data_out comes to 6c

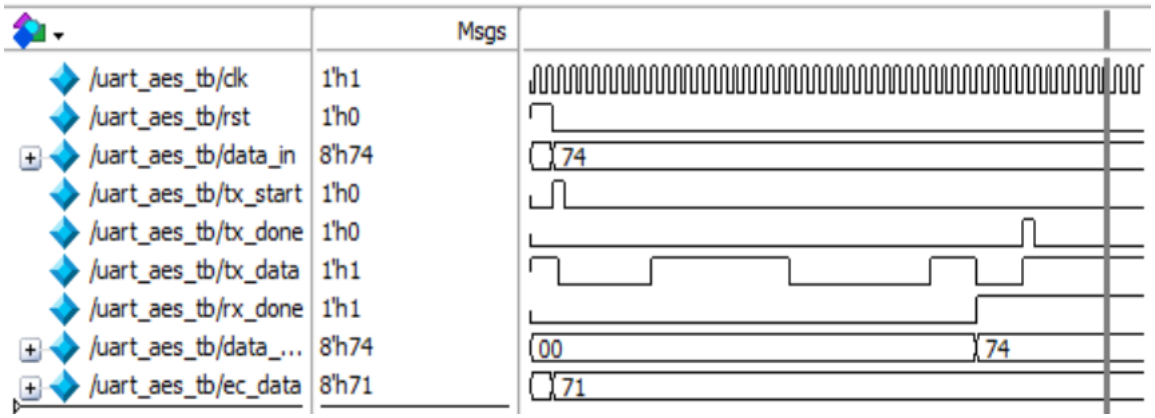

Fig. 3 When data_in is 74 and it is XOR'd with 05, data_out comes to 74

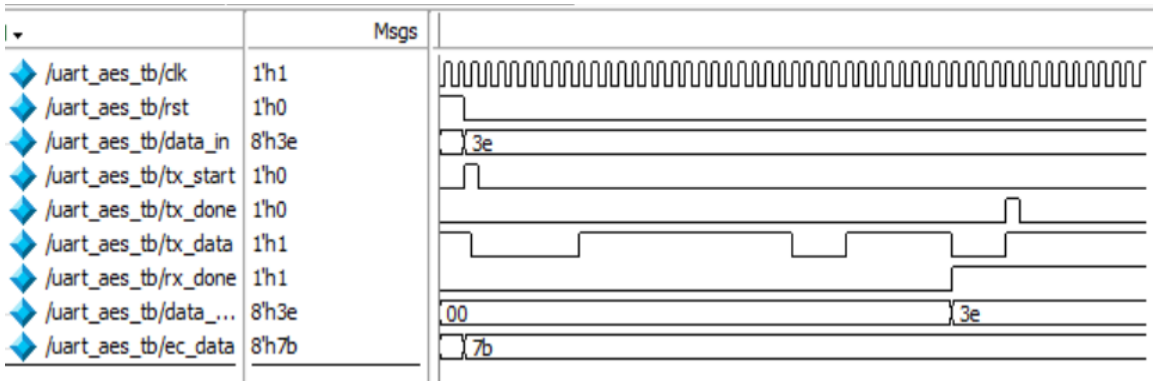

Fig. 4 When data_in is 3e and it is XOR'd with 45, data_out comes to 3e

These test results show that the system can effectively enhance the security of serial communication, providing a reliable solution for data transmission in embedded systems.

Future work may further optimize the implementation of the encryption algorithm to improve both the communication speed and security. For example, more advanced lightweight algorithms can be adopted to reduce power consumption. Additionally, by using software configuration to periodically change the encryption key on the hardware link layer, security can be further enhanced.

6. Conclusion

This paper designs a UART core integrated with an AES algorithm module, aiming to examine the feasibility of implementing data encryption at the hardware level. Through behavioral simulation by Verilog, it is concluded that the system can effectively enhance the security of serial communication and also demonstrates significant potential for further development. In the future, based on this foundation, chips that utilize internet-connected big data and artificial intelligence algorithms could be developed to analyze existing attack methods and autonomously generate the optimal encryption algorithm. This would enable wide application in areas such as national defense, finance, and healthcare.

References

[1] Su Zhan, Wang Hejian, Wang Huanjuan and Shi Xin. A Financial data security sharing solution based on blockchain technology and proxy re-encryption technology. 2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI), 2020, pp. 462-465.

[2] Borkar A M, Kshirsagar V R and Vyawahare V M. FPGA implementation of AES algorithm. 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 401-405.

[3] Standaert -X O, Peeters E, Rouvroy G and Quisquater -J J. An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays. Proceedings of the IEEE, vol. 94, no. 2, 2006, pp. 383-394.

[4] Huang Weilun and Sheng Guolun. Analysis and Research on UART Communication Protocol. 2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), 2024, pp. 768-771.

[5] Gupta K A, Raman A, Kumar N, and Ranjan R. Design and Implementation of High-Speed Universal Asynchronous Receiver and Transmitter (UART). 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020.

[6] Alkamil A and Perera G D. Towards Dynamic and Partial Reconfigurable Hardware Architectures for Cryptographic Algorithms on Embedded Devices. IEEE Access, vol. 8, 2020, pp. 221720-221742.

[7] Jhansi J, Yadav CHT T K, Bharathi M, Madhu G C, Peroumal K V and Mitra R. Software Based Performance Evaluation of Data Encryption Algorithms. 2025 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC), 2025, pp. 1-5.

[8] Daemen J and Rijmen V. The Design of Rijndael: The Advanced Encryption Standard (AES). Springer, 2002.

[9] Masoumi M and Mohammadi S. A new and efficient approach to protect AES against differential power analysis. 2011 World Congress on Internet Security (WorldCIS-2011), 2011, pp. 59-66.

[10] Ueno R, Homma N, Iida T and Minematsu K. High Throughput/Gate FN-Based Hardware Architectures for AES-OTR. 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019, pp. 1-4.