Evolution in Electrical and Electronic Engineering Vol. 3 No. 2 (2022) 872-878

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

High Throughput of AES Design in FPGA Implementation

Quek Ai Xian[1], Nabihah @ Nornabihah Ahmad[1]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2022.03.02.102
Received 20 June 2022; Accepted 17 July 2022; Available online 30 October 2022

Abstract: The AES algorithm, is a network security technique that is extensively used in all forms of wireless digital communication networks to secure data transfer between two end users. The National Institute of Standards and Technology (NIST) established the AES in 2001 as an electronic data encryption specification and it can improve the security of the systems through the use of key lengths. The aim of this project is to provide high throughput of AES design in Field Programmable Gate Array (FPGA) Implementation. The AES algorithm is designed by using Verilog Hardware Description Language (HDL) utilizing Quartus II software and is synthesized on FPGA Cyclone IV EP4CE115F29I7. The result of the AES design is that it runs at 125 MHz, has a throughput of 0.2 Gbps, and uses approximately 12044 logic elements (LEs). The proposed AES design is appropriate for use in portable device applications that can provide strong security features to help protect user privacy.

Keywords: AES, Throughput, FPGA

1. Introduction

As information and communication technology (ICT) progresses, medical records are increasingly transitioning from paper to electronic forms. Therefore, writing and saving the information on an electronic device can help the user to view the information quickly. The transfer of information may lead to the leakage of the user's privacy, therefore, it is crucial to encrypt information using cryptography. Cryptography is a technique that converts raw data into an unreadable format before encrypting it with an algorithm and a key. AES is a symmetric key encryption technology that has a longer key length and hence is more secure than other encryption techniques. This method was chosen by the National Institute of Standards and Technology (NIST) as the standard encryption algorithm in 2001, it has a fixed data length of 128 bits and key lengths of 128 bits, 192 bits, and 256 bits. Implemented AES in software in result higher latency and area, which can be solved by hardware implementation in order to reduce latency and area. Hardware implementations can be more secure than software implementations [1].

*Corresponding author: nabihah@uthm.edu.my

The objective of the research is to design a high throughput of AES design. The second objective is to design a high-performance AES using FPGA technology. The AES method is implemented using Quartus II software on an FPGA (Cyclone IV E), and the data and key lengths are both 128 bits. The AES algorithm is written in Verilog HDL. AES has a high performance with a frequency limitation parameter of 125 MHz, throughput of 0.2 Gbps, and element usage of 12000 LEs. The last objective is to verify the functionality of the AES. The functionality of the AES algorithm can be verified using functional and timing simulation in the Quartus II.

The problem statement is that nowadays people rarely use paper to share information, they all use some applications to share information, but using wireless communication networks will have the risk of data leakage. Sharing information can occasionally result in the leaking of a user's privacy, hence it is critical to employ cryptography such as the AES method to encrypt patient data. In addition, data needs to be encrypted and decrypted faster to keep up with the pace of people's lives and avoid delaying users' time. As a result, hardware implementations such as Field Programmable Gate Array (FPGA) are employed to eliminate significant time delays and area, while still providing efficient security.

2. Methodology

The design specification calls for an AES algorithm with 128-bit data, a 128-bit key length, a frequency constraint parameter of 125 MHz, a throughput of 0.2 Gbps, utilized of 12000LEs. The AES algorithm is written using Verilog Hardware Description Language (HDL) and compiled it at Quartus II. AES is implemented using FPGA EP4CE115F29I7 with Cyclone IV E device. When the compilation is complete, the functionality of AES can be checked using ModelSim Altera Starter Edition 10.1d. The design outcomes, such as throughput, frequency, and element utilization were examined. These findings are then examined to determine whether the system is suitable for secure medical record access under certain conditions such as encryption and decryption speed or security.

2.1 Architecture of AES

The AES design uses 128-bit data and a 128-bit key length for encryption and decryption. 128-bit key length requires 10 rounds, as shown in Figure 1. Key Expansion is a module that creates a succession of Round Keys from the Cipher Key while the encryption and decryption procedures are running. This design encrypts 128-bit plaintext to ciphertext and decode the data to retrieve the plaintext using the encryption output. It operates by doing the AddRoundKey transformation with input and key expansion, then repeating 9 rounds of four procedures: SubBytes, ShiftRows, Mixcolumn, and AddRoundKeys. The tenth round only executes three processes: SubBytes, ShiftRows, and AddRoundKeys. The output of 128 bits is then finished.

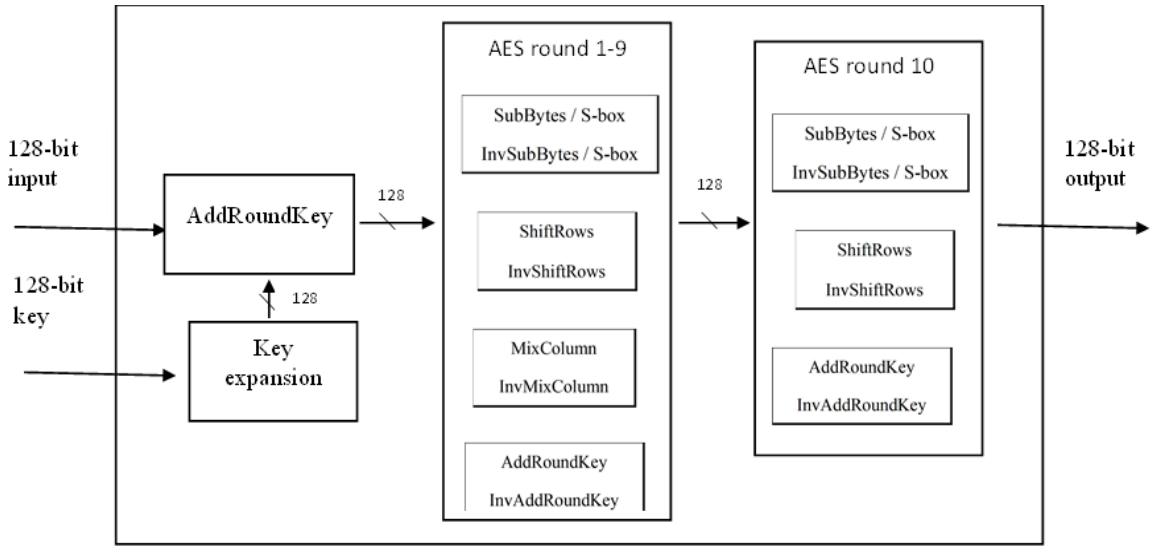

Figure 1: Block diagram of AES

In order to increase the throughput, a technique called pipelining is used as shown in Figure 2. To achieve high throughput, the register is inserted after each transformation. The data is only allowed into the register if the valid signal is active. The valid out signal will thereafter be active following the data out signal.

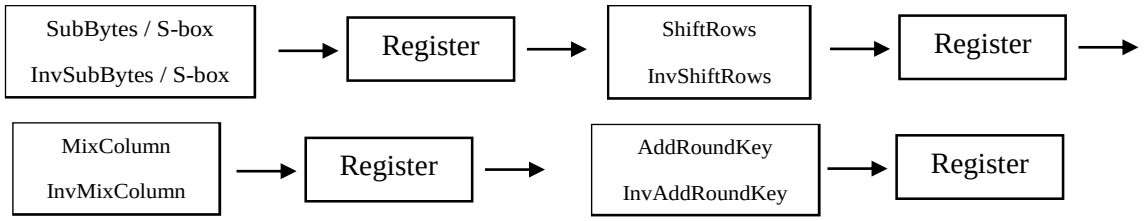

Figure 2: Technique to increase the throughput.

Figure 3 depicts the AES structure, which incorporates both encryption and decoding. To protect the system from being congested, multiplexers are used to choose an input signal from among numerous inputs and then execute the required output. Table 1 contains all of the pin descriptions. The AES's functionality was tested using the functional and timing simulation.

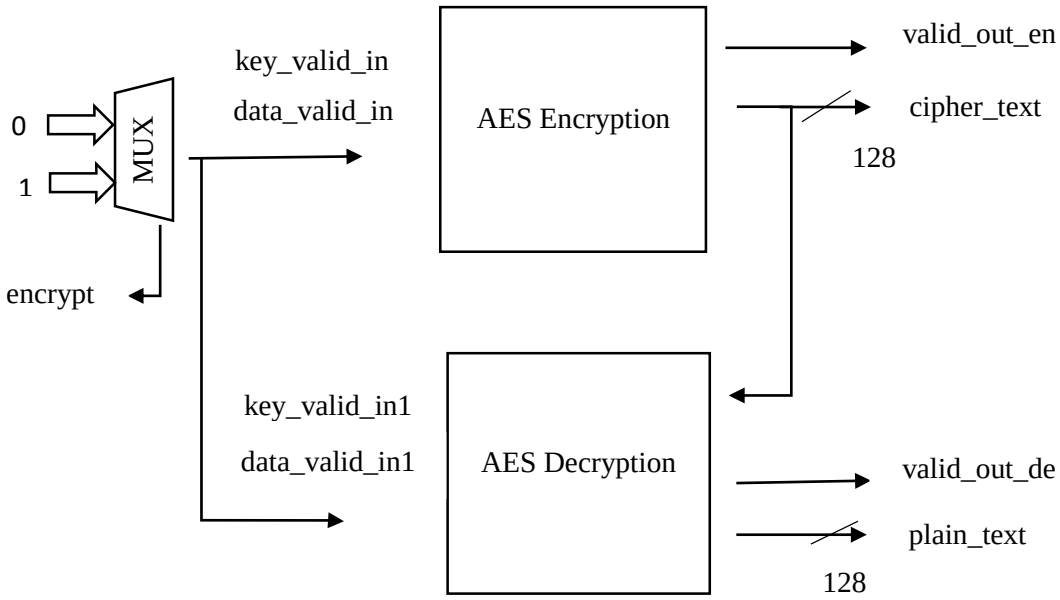

Figure 3: AES structure with both encryption and decryption.

Table 1: Description of the pin

Pin	Direction	Description
clk	Input	clock
reset	Input	Asynchronous Reset
Data_in	Input	Input data for plaintext or ciphertext
key_in	Input	Key used for encrypt and decrypt
data_valid_in	Input	Control signal: when active high, data in for encrypt.
data_valid_in1	Input	Control signal: when active low, data in for decrypt.
key_valid_in	Input	Control signal: when active high, key in for decrypt.
key_valid_in1	Input	Control signal: when active low, key in for decrypt.
valid_out_en	Output	Encrypt data is done when active high.
valid_out_de	Output	Decrypt data is done when active high.
cipher_text	Output	Output data for encrypting.
plain_text	Output	Output data for decrypt.

encrypt	Input	Control signal: when active high, start to encrypt, on the contrary, decrypt.

3. Results and Discussion

A control signal called encrypt governs the AES system. Begin encrypting when the active level is high. When the output signal rises to "1," the output of the encrypted or decrypted data is shown. The system may be reset once more by setting the reset to "1" to erase all data. Figure 4 shows the structure of the AES design. The usage of AES design is 12044 LEs less than 11%, showing that the FPGA area used is quite modest.

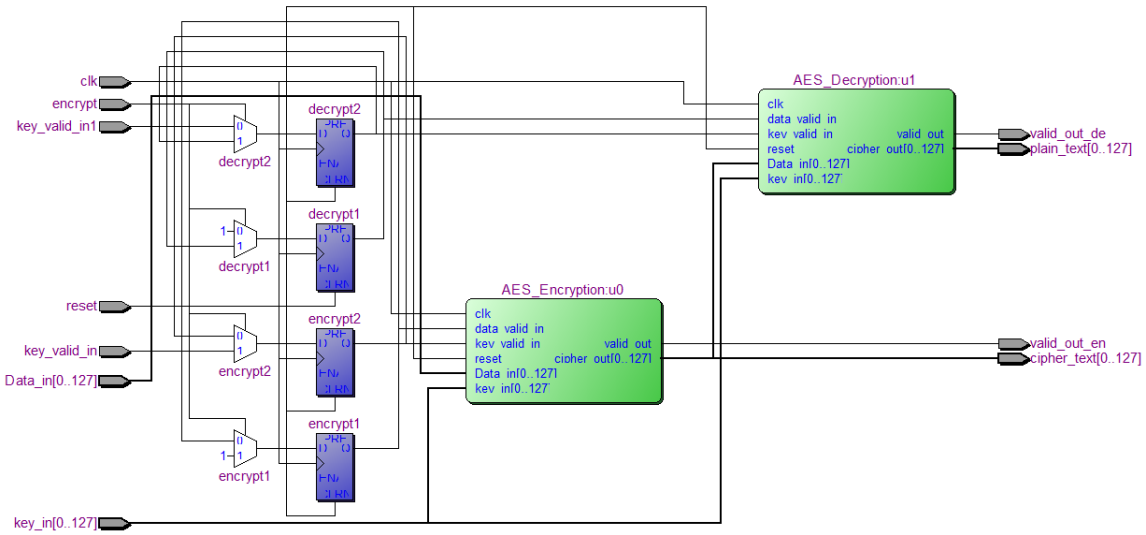

Figure 4: Structure of AES design.

Encryption takes 40 clock cycles to finish all data, whereas decryption takes 80 clock cycles to complete all data, as shown in Figure 5.

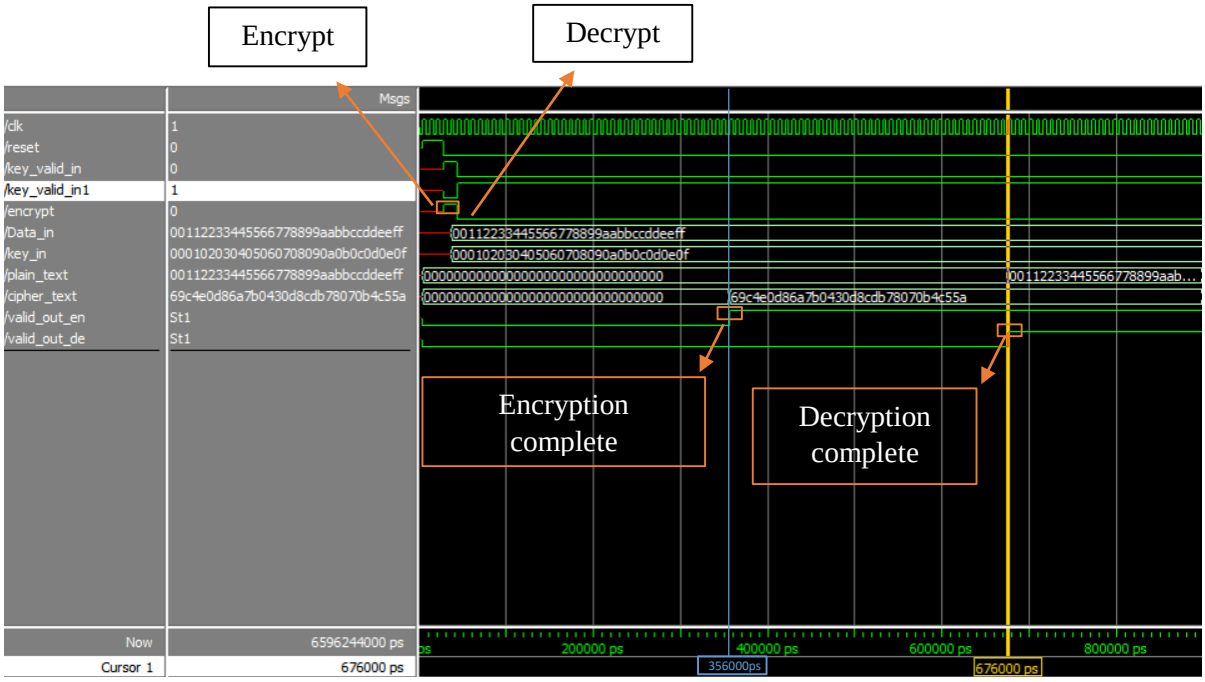

Figure 5: Simulation result for AES-128 encryption and decryption.

This system operates at a frequency of 125 MHz and has an 8ns period. AES throughput for AES is 0.2 $Gbps$. Throughput is defined as the AES performance of the FPGA device over a specific period. It refers to the amount of data sent successfully in a certain amount of time (measured in bits per second (bps)). The higher the frequency, the higher the throughput, but the bigger the number of components used. The throughput of the AES is calculated using Equation 1 [2].

$$Throughput = \frac{128\ bits}{\text{Cycles per Encrypted or Decrypted Block} * \text{Time period}} \qquad Eq.1$$

$$\begin{aligned} Throughput\ (encryption) &= \frac{128\ bits}{\text{Cycles per Encrypted Block} * \text{Time period}} \\ &= \frac{128\ bits}{40 * 8\text{ns}} \\ &= 0.4\ Gbps \end{aligned}$$

$$\begin{aligned} Throughput\ (decryption) &= \frac{128\ bits}{\text{Cycles per Decrypted Block} * \text{Time period}} \\ &= \frac{128\ bits}{80 * 8\text{ns}} \\ &= 0.2\ Gbps \end{aligned}$$

Table 2 shows the key, plaintext, and ciphertext used in simulation to check the functionality of AES. All the comparison results are shown in Table 3 with frequency, throughput, and element utilization for both encryption and decryption modes.

Table 2: Key, plaintext, and ciphertext are used in simulation to check the functionality of AES

key	Plaintext	Ciphertext
000102030405060708090a0b0c0d0e0f	00112233445566778899aabbccddeeff	69c4e0d86a7b0430d8cdb78070b4c55a
2b7e151628aed2a6abf7158809cf4f3c	3925841d02dc09fbdc118597196a0b32	7dfdff39cc79c14315baf5ef727cc0cf
000102030405060708090a0b0c0d0e0f	00112233445566778899aabbccddeeff	69c4e0d86a7b0430d8cdb78070b4c55a
2b7e151628aed2a6abf7158809cf4f3c	f34481ec3cc627bacd5dc3fb08f273e6	e42023437f94d94d2a085dfcd40c2cd0

All the comparison results are shown in Table 3 with frequency, throughput, and element utilization for both encryption and decryption modes. According to Table 3, the proposed work operates at a frequency of 125 MHz and has a throughput of 0.2 $Gbps$. Author [3] has 0.29 Gbps throughput with 2814 LUTs in one mode, author [4] has 0.197 Gbps throughput with 3047 slices in one mode, and author [5] has 60 Gbps throughput with 9756 slices in one mode. The AES hardware architecture for author [3] is parallelism but the frequency used is higher than the proposed work resulting in high throughput. The author [4] achieves almost the same frequency and throughput as the proposed work, but its AES algorithm is controlled by MicroBlaze. The author [5] proposed a pipeline structure and special multiplexer-based architecture for S-box implementation, and the high frequency is used in the

architecture resulting in very high throughput. So, it is the reason the result from the author [5] is higher than the proposed work. In [6], encryption and decryption throughputs are 1.28 Gbps and 1.07 Gbps with 10773 LUTs and 15240 LUTs, respectively. Hardware design approaches employed by the author [6] rely only on pre-computed look-up tables (LUTs) and high frequency is used resulting high throughput than the proposed work. In [7], encryption and decryption throughputs are 2.33 Gbps and 2.33 Gbps with 5037LEs and 5049 LEs, respectively. To achieve high throughput, THE author [7] presented an optimized and combined architecture using InvSubBytes and InvMixColumn.

The proposed design is operated slower than other previous work due to the technical specification of the FPGA Cyclone IV EP4CE115F29I7 such as the max frequency is 200MHz. The maximum frequency for the FPGA limits the use of high frequency to achieve high throughput, so all except the author [3] get higher throughput than the proposed work. FPGAs are built using customizable logic elements (LEs), also known as configurable logic blocks (CLBs). A logic element is made up of LUTs, which are programmable logic gates. So, LEs are often bigger than LUTs. As a result, high frequencies in the system provide high throughput, and high throughput causes the FPGA to employ more elements. The time to complete the encryption mode is 356000ps, whereas the time to complete the decryption mode is 676000ps. The device can encrypt or decrypt the data at a faster speed.

Table 3: Comparison result of AES with 128-bit data, 128-bit key length

Author	Device	Mode	Frequency (MHz)	Throughput (Gbps)	Element Utilization
Proposed work	Cyclone IV E	enc / dec	125	0.2	12044 LEs
[3]	ARTIX-7 xc7a100t-3-csg324	enc / dec	273.289	0.29	2814 LUTs for one mode
[4]	Artix-7 XC7A100T-1CSG324C.	enc / dec	100	0.197	3047 slices for one mode
[5]	Virtex5 xc5vfx70t-3ffl136	enc / dec	460	60	9756 slices for one mode
[6]	Virtex-7 XC7VX690T	enc	208.073	1.28	10773 LUTs
		dec	237.023	1.07	15240 LUTs
[7]	EPlS80F1508C5	enc	200	2.33	5037LEs
		dec	200	2.33	5049 LEs

4. Conclusion

Encrypt and decrypt information using the AES algorithm to ensure that the access system is safe and that people trust it. AES is implemented using Field Programmable Gate Array (FPGA) EP4CE115F29I7 with Cyclone IV E device. The AES architecture operates at 125 MHz with a throughput of 0.2 Gbps, with a total of 12044 LEs. Although it has a slower throughput than previous work on AES performance, it can encrypt or decrypt in a shorter time. In future work, to strengthen system security, the AES key length can be increased to 192 or 256 bits. Although 128 bits are good enough for security, it would take 1 billion years to crack AES with a key length of 128 bits. A 128-bit key length is sufficient for security, as it would take 1 billion years to crack that key length despite using a computer. Furthermore, the system's throughput might be increased to decrease the processing time for encrypting and decrypting, allowing user experts to check information in less time.

Acknowledgement

The authors would also like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] J. Yenuguvanilanka and O. Elkeelany, "Performance evaluation of hardware models of Advanced Encryption Standard (AES) algorithm," in IEEE SoutheastCon 2008, pp. 222–225, 2008.

[2] H. Hamzah, N. Ahmad, and S. H. Ruslan, "The 128-bit AES design by using FPGA," Journal of Physics: Conference Series, vol. 1529, no. 2, pp. 1-7, 2020.

[3] N. Jain, D. S. Ajnar, and P. K. Jain, "Optimization of Advanced Encryption Standard Algorithm (AES) on Field Programmable Gate Array (FPGA)," in 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 1086–1090, 2019.

[4] T. Abdelmoghni, O. Z. Mohamed, B. Billel, M. Mohamed, And L. Sidahmed, "Implementation of Aes Coprocessor For Wireless Sensor Networks," in 2018 International Conference on Applied Smart Systems (ICASS), pp. 1–5, 2018.

[5] H. Kouzehzar, M. N. Moghadam, and P. Torkzadeh, "A High Data Rate Pipelined Architecture of AES Encryption/Decryption in Storage Area Networks," in Electrical Engineering (ICEE), Iranian Conference on, pp. 23–28, 2018.

[6] N. S. S. Srinivas and Md. Akramuddin, "FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption," in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1769–1776, 2016.

[7] X. Zhang, H. Li, S. Yang, and S. Han, "On a High-Performance and Balanced Method of Hardware Implementation for AES," in 2013 IEEE Seventh International Conference on Software Security and Reliability Companion, pp. 16–20, 2013.