

PAPER • OPEN ACCESS

AES Algorithm Optimization and FPGA Implementation

To cite this article: Yufeng Liu *et al* 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **267** 042070

View the [article online](#) for updates and enhancements.

AES Algorithm Optimization and FPGA Implementation

Yufeng Liu¹, Xiangyang Xu² and Hao Su³

¹School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, Hebei Province, 050000, China

²School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, Hebei Province, 050000, China

³School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, Hebei Province, 050000, China

414695556@qq.com

Abstract. AES is an existing encryption algorithm with strong anti-attack capability, fast encryption speed and good portability. Implementing AES algorithms on FPGA can process data faster. In order to improve the running speed of the overall system, the re-design uses a full-flow technology to implement the algorithm. And the algorithm is optimized, finally using quartus for synthesis.

1. Introduction

In recent years, with the rapid development of network technology, people are increasingly demanding information security. The software implemented AES algorithm has been unable to keep up with the times because the encryption and decryption speed is not satisfactory limited by the serial system of the CPU. At the same time, software-based solutions are also weak in physical security and vulnerable to security attacks. In addition, the doubt that there may be a "back door" in the AES algorithm prompted us to explore and study it, and try to optimize the algorithm. In this paper, an optimized AES algorithm is proposed. The randomness test shows that the optimization scheme improves the security of the algorithm. According to the AES algorithm structure, the white-top design method is adopted, and a clock frequency and data throughput can be designed. Among them, the key problem in the algorithm is how to optimize each transform to improve security: the key issue in implementation is how to make an efficient encryption and decryption system based on the FPGA-based solution. Minimizing the resource consumption of chips while guaranteeing the speed.

2. Introduction to AES algorithm

Now is an era of rapid technological development. Every day, countless people use the Internet to send and receive e-mails, information, business data transmission, etc. As the openness of the network brings convenience to people, it also brings some threats of information leakage and tampering. The need for users to securely store information, securely transmit and securely process is becoming more and more urgent. Encryption algorithm is a kind of effective method for security protection of encryption technology. DES encryption algorithm protects people's information security for nearly 20 years. Due to its weak key property, it was broken in 1999, and the United States began to collect from the world. Urgent encryption algorithm, Rijndael stands out among the 15 candidate algorithms and becomes a new encryption algorithm.



Since the introduction of AES, it has been widely used in database encryption, video encryption, IC card and hard disk encryption of stored data. Because AES is easy to promote, software and hardware can be implemented. It is easy to upgrade in software, but the throughput is low. Security is also a hidden danger. Compared with software, hardware implementation is safe and difficult to be broken. In general, hardware implementations are available in both FPGA and ASIC modes. FPGA has programmability, configurability, ASCII lack of flexibility, and long development cycle. Therefore, FPGA is chosen as the hardware platform to implement AES algorithm.

The AES algorithm is a symmetric encryption algorithm, which mainly consists of three parts: encryption, decryption, and key expansion. The AES qualified plaintext packet length can only be 128 bits, and the key length can be any one of 128, 192 or 256 bits. The number of selected rounds N_r related to the round key is 10, 12, 14 rounds respectively. The three key lengths are integrated, and the required key length can be selected for encryption and decryption.

3. AES algorithm optimization

3.1. S-box optimization

Existing S-box research focuses on its design criteria and construction methods. The S-box design quasi-side mainly includes: nonlinearity, differential uniformity, algebraic number and number of items. Different properties are used to resist different attacks such as differential uniformity for resisting differential analysis, algebraic number and item number distribution for resisting interpolation attacks, etc. Usually there is a constraint relationship between the design criteria, and the design needs to be compromised. Based on the above design criteria, there are many S-box construction methods in the actual design: random selection and testing, almost complete nonlinear permutation, mathematical functions and compounding of mathematical functions in different groups.

In the AES algorithm, the S-box design follows the criteria: nonlinearity; algebraic complexity. The designer suggests to choose the inverse multiplication operation on $GF(2^8)$ to construct the S-box. However, the algebraic expression of the S-box generated directly by inverse transformation on $GF(2^8)$ is too simple, and the attacker may use this to interpolate attack. To compensate for this defect, the designer performed a transformation on the ring $Z_2[x]/(x^8+1)$ on the result of the inverse multiplication, which makes the generated S-box algebra expression very complicated. In addition, the designer has limited the transformation so that the S-box has no fixed points and no fixed points, although there is currently no attack method for this feature. The specific construction process of S-box in AES algorithm is as follows:

- Find the multiplicative inverse of the element on $GF(2^8)$:

$$X^{-1} = \begin{cases} x^{254}, & x \neq 0 \\ x, & x = 0 \end{cases}$$

- Affine transformation

$$b(x) = (u(x)a(x) + v(x)) \bmod (x^8 + 1)$$

It is denoted as L_u and L_v $u(x)$ and $v(x)$ are called affine transformation pairs, and the matrix expression is:

$$\begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Definition 1: If the positive integer n : $S^n(x)=x$, the S-box is said to be periodic, and the number of cycles that the elements in $GF(2^8)$ return to themselves through the continuous transformation of the S-box.

Definition 2: If the positive integer n : $L_{u,v}^n=E$ (E is a unit matrix), then the affine transformation $L_{u,v}$ is periodic and the affine transformation period can only be 1, 2, 4, 8 or 16. The AES algorithm S-box affine transformation period is 4, and the affine transformation period maximum is not reached, so the affine transformation period should also be considered as a design criterion.

It has been pointed out before that the number of S-box algebra expressions of the AES algorithm is 254, and the number of items of the algebraic formula is 9. The overly simple form raises doubts about its security. It has been proved that the matrix operation on $GF(2^8)$ is the main reason for the low complexity of S-box algebraic equations. The finite field generator polynomial $m(x)$, the affine matrix called (x) and the affine constant $v(x)$ are changed. Does not increase the number and complexity of S-box algebra. The above shortcomings are related to the order of multiplication inverse and affine transformation. They adopt a new affine transformation pair {6b.5d} and change the S-box construction order. However, this construction scheme performs 2 affine transformations. The circuit area and path delay increase, this paper finally chooses the S-box construction scheme: S-box nonlinear transformation is $y=(ux)^{-1}+v$, inverse S-box nonlinear transformation is $y=u^{-1}(x+v)^{-1}$, affine transformation pair $(u,v)=(\{34\}, \{ba\})$. The optimization scheme uses only one affine transformation, which is smaller than the original scheme circuit and has a lower path delay. Table 1 compares the S-box and S-box properties of the original AES algorithm. It can be seen that the optimized S-box has better properties.

Table 1. Compares the S-box and S-box properties of the original AES algorithm.

	Balance	Orthogonality	S-box strict avalanche criterion distance	Inverse S-box strict avalanche criterion distance	Affine transformation period	Iteration cycle	S-box algebraic term	Inverse S-box algebraic term
Original	Yes	Yes	432	536	4	88	9	255
This article	yes	yes	376	304	16	256	253	254

3.2. Mixcolumns optimization

In the operation of Mixcolumn and IncMixcolumn, the main operations are defined by the following two formulas:

$$out_x = [2 \ 3 \ 1 \ 1] \bullet [a \ b \ c \ d]^T \quad (1)$$

$$out_y [E \ B \ D \ 9] \bullet [a \ b \ c \ d]^T \quad (2)$$

In the design, you can rent two formulas from the following:

$$out_x = 2(a+b)+b+(c+d) \quad (3)$$

$$\text{And } out_y = 4(2(a+b)+2(c+d)+(a+c))+2(a+b)+b+(c+d) \quad (4)$$

The operations and results of equations (1) and (2) are listed in Table 2. The result of processing from step 1 to step 5 yields out_x , followed by out_x and W_8 to get out_y . Therefore, during the execution process, the hardware resources used in the operation and the results obtained by the operation can be used in step 9 and step 10. As shown in Figure 1, this new structure (byte-column hybrid module) requires only 8 adders and 4 multipliers. This design greatly reduces the complexity of the hardware and significantly saves resources compared to the original solution.

Table 2. Recombination operation.

step	operating	step	operating
1	$W_1=a+b$	6	$W_6=W_2+W_4+W_5$
2	$W_2=a+c$	7	$W_7=W_6*2$
3	$W_3=c+d$	8	$W_8=W_7*2$
4	$W_4=W_1*2$	9	$Out_x=W_3+W_4+b$
5	$W_5=W_3*2$	10	$Out_y=Out_x+W_8$

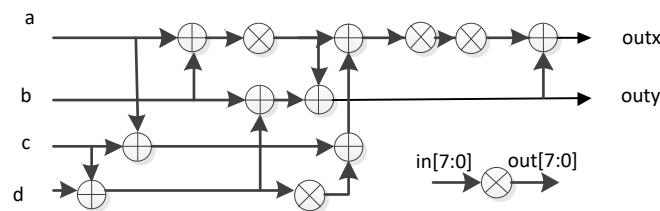


Figure 1. Column Mixing Module

The calculation formula in the multiplier module in Figure 1 is:

$$out[7:0] = \{in[6:4], in[3:0]\} \wedge \{in[7], in[7], 0, in[7], in[7]\}$$

4. Simulation and final implementation results

4.1. Overall design

This article uses the full-flow technology, which is to add a register shortening path demonstration in the combination logic unit to meet higher clock frequency requirements, improve system throughput, and this design is to use 128, 192, 256 bits. The key lengths are combined to achieve three free choices, and encryption and decryption can be performed simultaneously. Compared with the design of the AES algorithm before optimization, this design has been improved and optimized in many aspects, which greatly improved the work efficiency. Figure 2 is the circuit interface diagram of the overall design, Table 3 is the circuit port declaration, keylen represents the key length selection 00 represents 128bit, 01 represents 192bit, and 10 represents 256bit.

Table 3. Recombination operation.

Pin signal	direction	Bit width	description
Clk	In	1	System clock
Key_valid	In	1	Key is valid
Reset_n	In	1	Reset
Key[255:0]	In	256	Key length
Keylen[1:0]	In	2	Key length selection
Enc_block[127:0]	In	128	Encryption
Dec_block[127:0]	In	128	Decrypt
Enc_new_block[127:0]	Out	128	Ciphertext
Dec_new_block[127:0]	Out	128	Plaintext

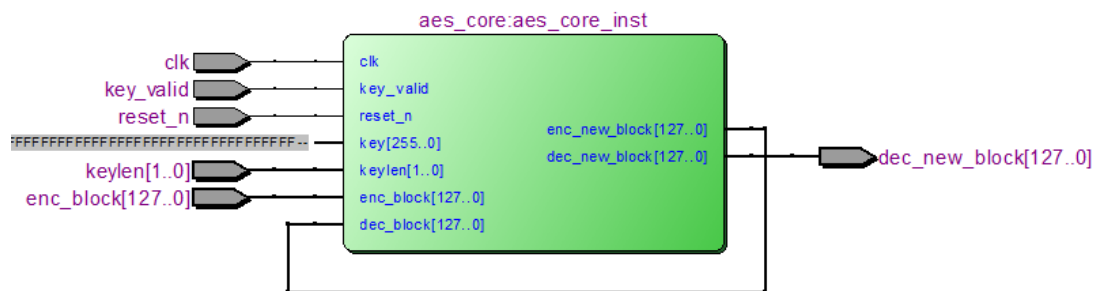


Figure 2. The circuit interface diagram of the overall design

4.2. Simulation and performance comparison

This design choice is the simulation using Modelsim, the full flow structure can be seen from the overall simulation, as shown in Figure 3.

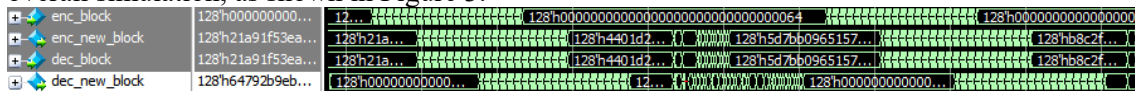


Figure 3. Overall simulation.

Figure 4 is the verification of the encryption simulation, the encryption is selected by the 256-bit key.

Key: 256'h603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4

Plaintext: 128'h00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C8

Ciphertext: 128'h34 68 97 56 8B 1D 3C 03 E2 54 EB 9E 6E 25 A8 F7

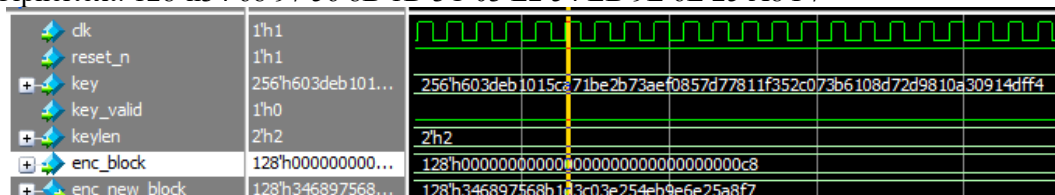


Figure 4. The verification of the encryption simulation.

Figure 5 is the decryption simulation, which is the inverse of the encryption simulation.

Key: 256'h603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4

Ciphertext: 128'h34 68 97 56 8B 1D 3C 03 E2 54 EB 9E 6E 25 A8 F7

Plaintext: 128'h00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C8

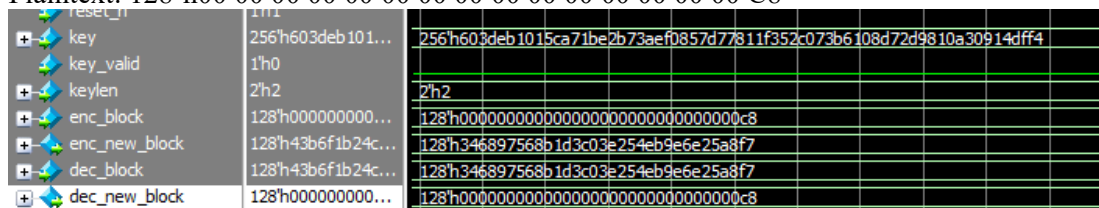


Figure 5. The decryption simulation.

5. Conclusion

In this paper, FPGA is used to realize the encryption and decryption process of the optimized AES algorithm. This design optimizes the S-box and column mixing in the AES algorithm. In the implementation process, the whole-flow method is used to operate, which greatly improves the operation, reducing the use area, is also optimized in the design process, so that encryption and decryption can be carried out at the same time, and the length of the key can be selected at will, greatly improving work efficiency and security.

References

- [1] El Maraghy, M., Hesham, S., Abd El Ghany, M.A. (2013) Real-time efficient FPGA implementation of aes algorithm. In: IEEE International SOC Conference. Erlangen. pp. 203-208.
- [2] Zhou, Y.B., Li, Y.Z. (2014) The Design and Implementation of a Symmetric Encryption Algorithm Based on DES. In: 2014 5th IEEE International Conference on Software Engineering and Service Science. China. 2014: 517-520.
- [3] Jamal, S. (2017) Implementation of Advanced Encryption Standard (AES) 192 Bit on FPGA. J.JOURNAL OF INFORMATIONC OMMUNICATION TECHNOLOGIES AND ROBOTICS APPLICATIONS (JICTRA). 2(02), 2226-3683.
- [4] Zhu, Y.W., Zhang, H.Q., Bao, Y.B. (2013) Study of the AES Realization Method on the Reconfigurable Hardware. In: 2013 International Conference on Computer Sciences and Applications. China. 2013: 72-76.
- [5] Parikh, P., Narkhede, S. (2016) High performance implementation of mixing of column and mixing of column for AES on FPGA. Computation of Power, Energy Information and Commuincation (ICCPEIC), 2016 International Conference on. IEEE, Kaohsiung. Taiwan, 2016: 174-179.