Journal of Electronics and Information Science(2017) 2: 93-97
Clausius Scientific Press, Canada

Hardware Implementation of AES Encryption Algorithm Based on FPGA

Huanqing Xu[1, a], Yuming Zhang[2, b] and Jun Yang[3, c]

[1,2,3]School of Information Science and Engineering, Yunnan university, Kunming, China
[a]907889351@qq.com, [b]zhangyuming0722@qq.com, [c]junyang@ynu.edu.cn

Keywords: FPGA; AES encryption; Nios II; asymmetric encryption

Abstract: With the development of society, the information industry has attracted more and more attention by the state. Since the emergence of prism doors, it has made countries pay great attention to the direction of information security. The question about how to protect information security has become an increasingly concerned issue. This paper introduces a widely used algorithm based on FPGA of symmetric encryption algorithm AES, because its key has three kinds of length 128bit, 192bit, 256bit, which can guarantee its difficulty in the crack, so it is relatively safe, this design can include encryption path and decryption path, you can also shield the decryption path and only include the encrypted path to reduce the use of resources in order to apply to resources insufficiently when the data is encrypted. Besides, this article can also be used for the key and data transmission using 32bit bus, multi-clock transmission. Through the Jtag Uart module to achieve the computer and embedded system communication, you can use it in the IDE integrated environment to achieve the program window to debug and monitor.

1. Introduction

AES encryption and decryption has key length, anti-differential ability, which it is easy to achieve , Low cost, fast , which is to replace the traditional DES, 3DES encryption standards .On the other hand, Traditional software encryption method is to deal with slow, poor real-time , which is compared to the hardware encryption, it is easier to crack. The nios II which is processing system's peripheral configuration has great flexibility, which is depending on the user's specific needs and doing a variety of Peripherals upgrades, etc., rapid expansion, tailoring the corresponding IP CORE, thus it is ready to be a complete, powerful sopc system, not only can achieve flexible configuration, custom command, Remote hardware upgrades, etc., and the development cycle is Short, which it is not need to change the hardware layout design. Here the use of nios II processor is to achieve AES-based encryption and decryption system, AES algorithm is relative to some addition and multiplication, these addition and multiplication are defined in a specific field, which is characterized by the efficient use of hardware to achieve. The test data is used to encrypt and decrypt the system, the editor uses the random number generator to generate the key and the processed data is to increase the accuracy of the system test. The random number generator is designed by the editor according to the cellular automata theory Using FPGA independent design.

2. AES Algorithm Overview

AES algorithm includes encryption and decryption algorithm which is key expansion algorithm, because the AES algorithm is not completely symmetric, so encryption and decryption path has its own hardware. The encryption process consists of byte substitution transformations, row shift transformations, column mixed transformations, and round keys, and the resulting Nr rounds of the algorithm, where the last round does not to do column hybrid transformations. The encryption process is similar to the decryption process, except that the various transformations are used. The subkey where is used in the encryption and decryption algorithm is the same, and each round requires an extension key to participate, but the order of use is just the opposite. Since the length of the external input encryption key and decryption key is limited, the AES algorithm requires a key extension algorithm to generate the encryption and decryption keys which are required for each round. AES operations are byte-based, all the variables are used by the appropriate byte, the middle variable with the following 4 * 4 byte matrix expression in Figure 1.

$$S_{r,c} = \begin{bmatrix} s_{0,0}, s_{0,1}, s_{0,2}, s_{0,3} \\ s_{1,0}, s_{1,1}, s_{1,2}, s_{1,3} \\ s_{2,0}, s_{2,1}, s_{2,2}, s_{2,3} \\ s_{3,0}, s_{3,1}, s_{3,2}, s_{3,3} \end{bmatrix}$$

Fig.1 4 * 4 byte matrix expression

AES algorithm encryption process text description is as follows:

(a) to define a 128-bit intermediate state variable State, which is stored as a byte matrix;

(b) to give a plaintext to be decrypted, assign it to State, call the round key XOR function, the function is the key which is generated by the state and key extension function, and the output value covers the State variable.

(c) Nr-1 iteration of the State variable in the order of the round key exclusive OR, byte substitution, row shift, and column mixing. State stores the encryption result after each cycle;

(d) the last round of the cycle will be mixed with the cycle of operation for the cycle of the key XOR function, the results stored in the State;

(e) Define State as a cipher text and output it.

3. FPGA-based AES Encryption and Decryption System Design

S-box transform, also known as byte transform, which is the only nonlinear operation that plays a decisive role in the security level of the algorithm. In order to make the lookup table operation faster, logical resource utilization is higher, we will replace the table stored in the FPGA internal chip RAM unit, the encryption process according to the length of the input data to find RAM in the replacement table and to complete each an quick operation of the bytes. The S-box in this design has a reconfigurable function, It improves its security level and application range, reduces power consumption, and it is better resistant to differential attacks. we construct the following S-box module using the multiplicative inverse and affine transformation of the finite field. the S-box is also suitable for the S-box transform module (Inv Sub-byte). The design of the key scheduling module is shown in Figure 2.

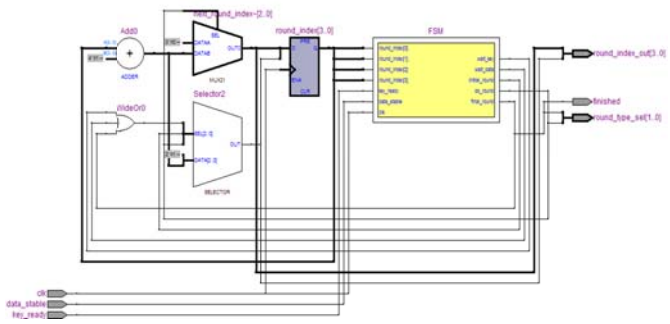

Fig. 2 Key scheduling module logic circuit diagram

4. Simulation Test and Performance Analysis

In the previous chapters, the background, theoretical basis and design method of the chaos neural network security processor model which is based on FPGA are expounded in detail. In this chapter, the security processor model is simulated and tested, which is of the processor system demonstration. The hardware part of the platform is developed by using the DE2 board. The software part is tested using the Nios II development environment. The security processor is compiled, integrated and simulated in the Quartus II and Modelsim test software to verify that the system meets the requirements in terms of timing and function. Pin configuration is based on EP2C35F672C6 chip. Below we will be on the AES encryption and decryption sub-module, which is based on chaotic neural network S-box module and safe processor model system simulation test and system performance analysis.

(a) S box conversion module

From Figure 3 S-box transformation simulation diagram can be seen, 00 corresponds to the conversion of 63,42 into 2C, refer to the S-box conversion table shows that the S-box module function is correct, to complete the byte replacement operation.

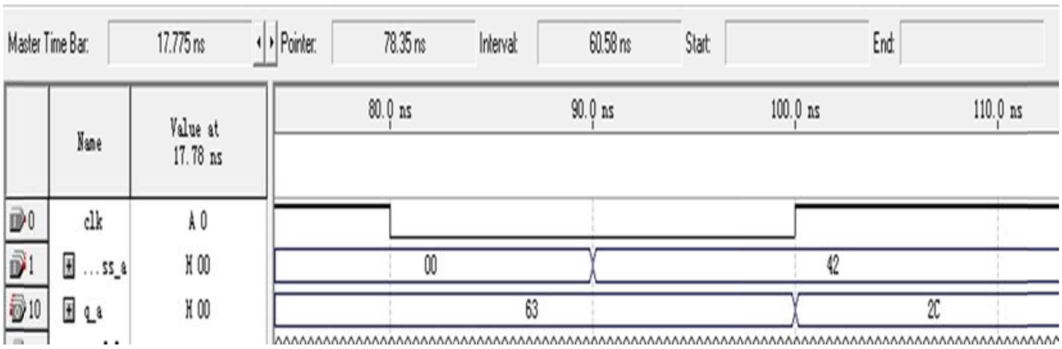

Fig. 3 S box transformation simulation chart

(b) line shift module

Here the line shift module to 4*4 byte matrix as input, in order to display convenience, the matrix transpose, so the two rows as a whole unit, a total of four units, you can see the first unit has not changed , The second unit begins to move forward in turn. The conversion of the encryption process is as follows, and the decryption process is reversed. The row shift module simulation in the specific encryption mode is shown in Figure 4.

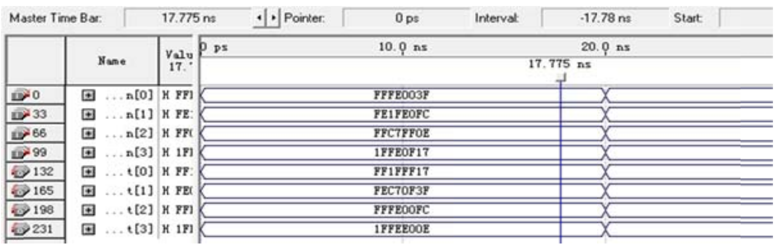

Fig.4 Line shift module simulation diagram (encryption mode)

(c) column mixing module

The simulation of the column mixing module in the encryption mode is shown in Figure 5. According to the principle of Chapter II, we can see that the function simulation of the module is correct.

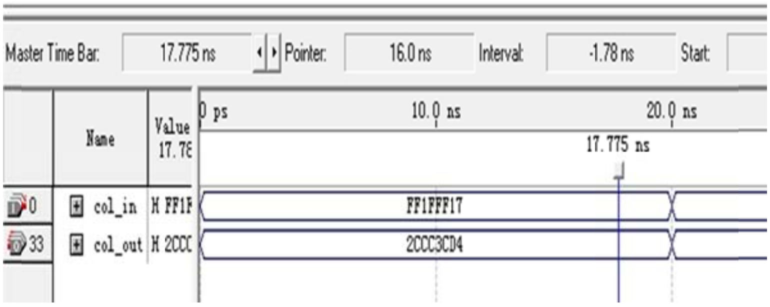

Fig.5 Column Mixing Module Simulation Chart (Encryption Mode)

5. Overall Comprehensive Performance Analysis

From the comprehensive results on the map, we can find the use of the system hardware resources, the total number of system chip logic unit 33216, the number of pins is 475, the number of storage is 483840. The system occupies 4988 (15%) logical units, 76 (16%) and 75776bits (16%) of memory, the system's hardware resource utilization is higher, the system's maximum stable frequency of 93.64MHz. Table 1 shows the comparison of the test data of this scheme with the related scheme. It can be seen that the scheme still has less hardware resource consumption and higher through put after the introduction of the chaotic neural network improved S-box, and the comprehensive performance ratio Very good, effectively improve the security processor model of resource utilization, it can be widely used in the field of security encryption.

Table 1 Demonstration data example

Program	Hardware consumption (per number)	Throughput rate (Mbps)	Performance ratio (Mbps/numeber)
This program	5600	4590	0.819
Traditional solution (not improved S box)	2800	1080	0.385
Literature [64] program	1569	279	0.177
Literature [65] program	7890	4980	0.631

6. Conclusion

This chapter is simulation test and performance analysis. Firstly, the simulation and system demonstration of the AES module of the encryption and decryption system, the improved S-box and the safety processor model of the chaotic neural network are presented respectively. Secondly, for the improved S-box, the use of nonlinear, avalanche, differential approximation probability and other performance indicators to assess. Finally, through the comparison with other programs to achieve the overall platform performance analysis. The analysis shows that the scheme has less hardware resource consumption and higher throughput, and the comprehensive performance ratio is very good, which effectively improves the resource utilization rate of the security processor model and can be widely used in the field of security encryption.

References

[1] Imaña J L. Low-delay AES polynomial basis multiplier[J]. Electronics Letters, 2016, 52(11):930-932.

[2] Jankowski K, Laurent P. Packed AES-GCM Algorithm Suitable for AES/PCLMULQDQ Instructions[J]. IEEE Transactions on Computers, 2011, 60(1):135-138.

[3] Rahimunnisa K, Karthigaikumar P, Kirubavathy J, et al. A 0.13-μm implementation of 5 Gb/s and 3-mW folded parallel architecture for AES algorithm[J]. International Journal of Electronics, 2014, 101(2):182-193.

[4] Priya S S S, Karthigaikumar P, Sivamangai N M, et al. High Throughput AES Algorithm Using Parallel Subbytes and MixColumn[J]. Wireless Personal Communications, 2016:1-17.

[5] Cho J, Soekamtoputra S, Choi K, et al. Power dissipation and area comparison of 512-bit and 1024-bit key AES[J]. Computers & Mathematics with Applications, 2013, 65(9):1378-1383.

[6] Alzahrani A, Demara R F. Fast Online Diagnosis and Recovery of Reconfigurable Logic Fabrics using Design Disjunction [J]. IEEE Transactions on Computers, 2016, 65(10):1-1.

[7] Jun Y, Jun D, Na L, et al. FPGA-Based Design and Implementation of Reduced AES Algorithm[J]. IEEE, 2010, 2:67-70.