Design and Implementation of Area-optimized AES Based on FPGA

AI-WEN LUO, QING-MING YI, MIN SHI
College of Information Science and Technology
Jinan University
Guangzhou, China
e-mail: communication_2009@126.com

***Abstract*—A new FPGA-based implementation scheme of the AES-128 (Advanced Encryption Standard, with 128-bit key) encryption algorithm is proposed in this paper. For maintaining the speed of encryption, the pipelining technology is applied and the mode of data transmission is modified in this design so that the chip size can be reduced. The 128-bit plaintext and the 128-bit initial key, as well as the 128-bit output of ciphertext, are all divided into four 32-bit consecutive units respectively controlled by the clock. The synthesis verification based on HJTC0.18um CMOS process shows that this new program can significantly decrease quantity of chip pins and effectively optimize the area of chip.**

Keywords-Area optimization; Pipelining; Verilog; FPGA

I. INTRODUCTION

With the rapid development and wide application of computer and communication networks, the information security has aroused high attention. Information security is not only applied to the political, military and diplomatic fields, but also applied to the common fields of people's daily lives. With the continuous development of cryptographic techniques, the long-serving DES algorithm with 56-bit key length has been broken because of the defect of short keys. The "Rijndael encryption algorithm" invented by Belgian cryptographers Joan Daemen and Vincent Rijmen's had been chosen as the standard AES (Advanced Encryption Standard) algorithm whose packet length is 128 bits and the key length is 128 bits, 192 bits, or 256 bits. Since 2006, the Rijndael algorithm of advanced encryption standard has become one of the most popular algorithms in symmetric key encryption. AES can resist various currently known attacks.

Hardware security solution based on highly optimized programmable FPGA provides the parallel processing capabilities and can achieve the required encryption performance benchmarks. The current area-optimized algorithms of AES are mainly based on the realization of S-box mode and the minimizing of the internal registers which could save the area of IP core significantly.

One new AES algorithm with 128-bit keys (AES-128) was described in this paper, which was realized in Verilog Hardware Description Language. The 128-bit plaintext and 128-bit key, as well as the 128-bit output data were all divided into four 32-bit consecutive units respectively. The pipelining technology was utilized in the intermediate nine round transformations so that the new algorithm achieved a balance between encryption speed and chip area, which met the requirements of practical application.

978-1-61284-109-0/11/$26.00 ©2011 IEEE

Firstly, functional simulation and timing analysis of this algorithm had been achieved in the ModelSim SE PLUS 6.0 and the Quartus Ⅱ 7.2 platform. Then we completed the synthesis simulation of this design based on HJTC0.18um CMOS process in ASIC design and verification platform provided by Synopsys Company. The data of each column (32 bits) in the state matrix was used to be an operand of encryption, when the operation of ShiftRows and SubBytes were incorporated. And each round of the intermediate nine Round Transformations of encryption was processed by pipelining technology.

The results show that this new algorithm with pipelining technology and special mode of data transmission can significantly decrease the quantity of chip pins and reduce the chip area.

II. THE FPGA IMPLEMENTATION OF AREA-OPTIMIZED AES-128

A. Brief Description of Rijndael Algorithm

Rijndael algorithm consists of encryption, decryption and key schedule algorithm.

The main operations of the encryption algorithm among the three parts of Rijndael algorithm include: bytes substitution (SubBytes), the row shift (ShiftRows), column mixing (MixColumns), and the round key adding (AddRoundKey). It is shown as Fig. 1.

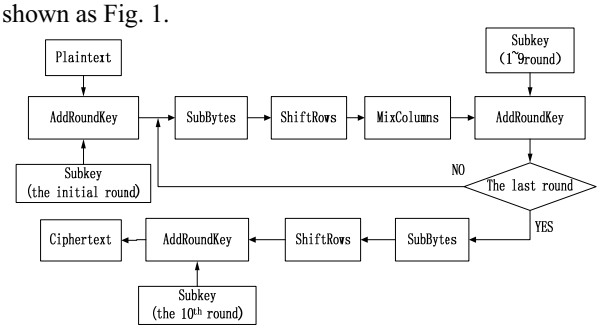

Figure 1. The structure of Rijndael encryption algorithm

Encryption algorithm processes Nr+1 rounds of transformation of the plaintext for the ciphertext. The value of Nr in AES algorithm whose packet length is 128 bits should be 10, 12, or 14 respectively, corresponding to the key length of 128,192,256 bits.

In this paper, only the (AES-128) encryption scheme with 128-bit keys is considered.

B. The Design of Improved AES-128 Encryption Algorithm

1) Two main processes of AES encryption algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation.

Key schedule consists of two modules: key expansion and round key selection. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selectes Nb bits of round key from the expanded key module.

Round Transformation involves four modules by ByteSubstitution, ByteRotation, MixColumn and AddRoundKey.

2) Key points for the design:

In the AES-128, the data in the main process mentioned above is mapped to a 4×4 two-dimensional matrix. The matrix is also called state matrix, which is shown as Fig.2.

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Figure 2. The state matrix

In the four transformation modules of round transformation, the ByteRotation, MixColumn and AddRoundKey are all linear transformations except the ByteSub.

Take analysis of the AES algorithm principle and we can find:

- ByteSubstitution operation simply replaces the element of 128-bit input plaintext with the inverse element corresponding to the Galois field GF (2^8), whose smallest unit of operation is 8 bits/ group.
- ByteRotation operation takes cyclic shift of the 128-bit state matrix, in which one row (32 bits) is taken as the smallest operand.
- MixClumns operation takes multiplication and addition operations of the results of ByteRotation with the corresponding irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ in GF(2^8), whose minimum operating unit is 32 bits.
- Addroundkey operation takes a simple XOR operation with 8-bit units.

The inputs of plaintext and initial key, intermediate inputs and outputs of round transformation, as well as the output of ciphertext in the AES algorithm are all stored in the state matrixes, which are processed in one byte or one word. Thus, in order to take operations at least bits, the original 128-bit data should be segmented. We design some external controllers in the new algorithm, so that the data transmission and processing can be implemented on each column of the state matrix (32bit). That means the data should be packed and put into further operations.

Take the independent and reversible bytes substitution operation of S-box as example. Firstly, the state matrix is divided into four columns. And then byte replacement is achieved by the operation of look-up table shown as Fig. 3.

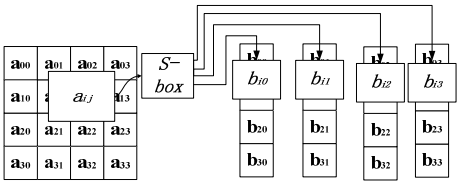

Figure 3. Bytes segmentation and replacement processing

Therefore, the original 128-bit input of plaintext and key will be replaced with four consecutive 32-bit input sequences respectively. In order to decrease the output ports, four continuous 32-bit ciphertext sequences have taken place of the original 128-bit output by adding a clock controller. The 128-bit data in the round transformation is also split into four groups of 32-bit data before the operation of pipelining.

C. The Process of New algorithms

From the above analysis, we can find that the process of AES encryption can be mainly divided into two parts: key schedule and round transformation. The improved structure is also divided into these two major processes. The initial key will be sent to the two modules: Keyexpansion and Keyselection, while the plaintext is to be sent to the round transformation after the roundkey is selected. But the operand of data transmission is turned into a 32-bit unit. The process of new algorithm is shown as Fig. 4.

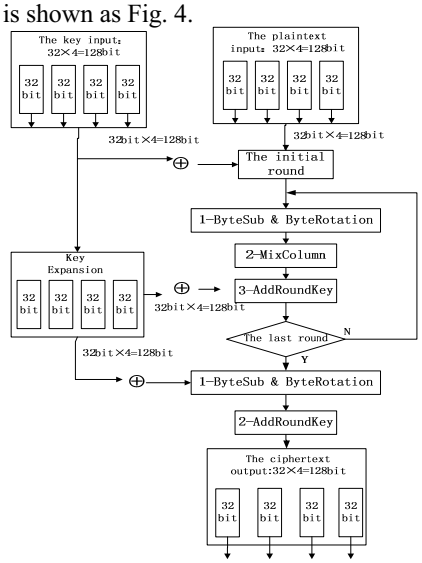

Figure 4. The new improved structure of AES algorithm

The functions of various parts of the structure shown above are described as follow:

- The initial round of encryption:

The four packets of consecutive 32-bit plaintext (128 bits) have been put into the corresponding registers. Meanwhile, another four packets of consecutive 32-bit initial key (128 bits) have been put into other registers by the control of the enable clock signal. Furthermore, this module should combine the plaintext and initial key by using the XOR operators.

- Round Transformation in the intermediate steps:

A round transformation mainly realizes the function of SubBytes and MixColumns with 32-bit columns. Four packets

of round transformation are processed independently. Then the results of MixColumns and the 32-bit keys sourced from Keyexpansion are combined by using XOR operators. Here, the round transformation is a module with 64 input ports (32-bit plaintext+32-bit key) and 32 output ports.

The function of SubByte is realized by Look-Up Table (LUT). It means that the operation is completed by the Find and Replace after all replacement units are stored in a memory (256×8bit = 1024 bit).

The implementation of MixColumn is mainly based on the mathematical analysis in the Galois field GF(2^8). Only the multiplication module and the 32-bit XOR module of each processing unit(one column) are needed to design, because the elements of the multiplication and addition in Galois field are commutative and associative. Then the function of MixColumn can be achieved.

Fig.5 is a block diagram for the introduction of pipelining technology used in the round transformation.

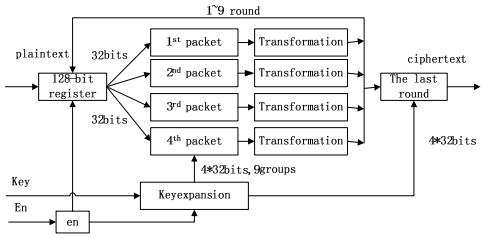

Figure 5. The round processing with pipeline technology

In the process of pipelining, the 128-bit data is divided into four consecutive 32-bit packets that take round transformation independently.

The operation of the above four groups of data can be realized in pipelining technology. In brief, it can be described as follow: store the unprocessed data in the 128-bit register, and control the clock for re-starting the 128-bit register to read the new data when the four groups' operations have been overcome. Thus the 128-bit round-operating unit has been transformed into four 32-bit round-operating elements. The internal pipelining processing should be implemented during the whole nine intermediate Round Transformations of the four packets before achieving the 128-bit ciphertext.

- The process of the last round

The final round is a 128-bit processor. After nine rounds of operations included Shiftrows, SubByte and Mixclumns, the 128-bit intermediate encrypted data will be used in XOR operation with the final expanded key(4*32bit), which is provided by the key expansion module. The output of final round in the processor is the desired 128-bit ciphertext.

Similarly, the ciphertext is divided into four packets of 32-bit data by an external enable signal.

- Key expansion and Key extraction

This module is implemented basically the same with the traditional way as another part of the AES encryption algorithm. The only difference lies on the mode of data transmission. The initial key and expanded keys are divided into four 32-bit data before being extracted.

All of the above modules can be decomposed into basic operations of seeking and XOR if the AES algorithm is implemented on FPGA. So the basic processing unit (look-up-table) of FPGA can be used. The operation of AddRoundKey is taken first in each round. When the plaintext and initial key are input, the encryption module starts running, and the expanded keys are stored into the registers at the same time. This implementation method is independent on a specific FPGA.

III. FUNCTIONAL SIMULATION AND SYNTHESIS VERIFICATION

In this paper, the new structure of AES-128 encryption algorithm introduced above is implemented with Verilog hardware description language, while minimizing the input / output ports to save redundant area of the chip.

The V file named aes_control in the project of the design contains the input and output ports, interface converters and controllers. Other function modules are described in independent V files respectively. We used ModelSim SE PLUS 6.0 for the waveform simulation platform and verified the results. And then took a further validation in Quartus Ⅱ 7.2 before synthesis verification. Then we completed the synthesis simulation of the design in ASIC design and verification platform of Synopsys using HJTC0.18um CMOS technology.

A. The Simulation in the Modelsim SE PLUS 6.0 Platform

Firstly, all project files of the design were compiled in Modelsim SE PLUS 6.0 simulation platform. If the files were all compiled successfully, the simulated waveforms could be obtained when loading the test file *test_bench_top*. Fig. 6 shows the simulation waveform of the new algorithm.

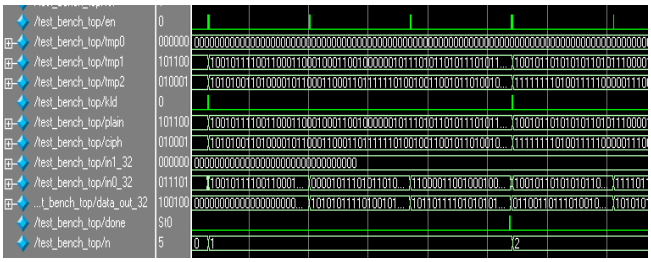

Figure 6. The 32-bit plaintext, 32-bit initial key and 32-bit cyphertext

The initial 128-bit input tmp0 sequences are extracted to four 32-bit words as the plaintext (128bit) shown as Fig.6; meanwhile, the 128-bit input sequences tmp1 are extracted to four 32-bit words as initial key (128bit); the sequences of tmp2(128bit) are the correct ciphertext data, which is used for validating the correctness of the new encryption scheme.

Shown as the waveforms in Fig.6, we found that the input in0 of four continuous state words and 128 bits plaintext tmp0 express the same by the control signal of en; four consecutive state-words of input in1 are consistent with 128 bits key. After a complete process of AES encryption, the output stream data_out_32 exports four continuous 32-bit sequences, which are consistent with the 128bits ciphertext tmp2.

In conclusion, the logic function of improved algorithm is correct and it satisfies the requirement of AES encryption algorithm.

B. The Simulation in the Quartus Ⅱ7.2 Platform

The logic function of new AES has been verified in the ModelSim SE PLUS 6.0 platform. In order to take the pre-analysis about the physical parameters of the chip before synthesis verification, a successful simulation has been done in the platform of Quartus Ⅱ 7.2.

Simulated in the device *EP2C70F896C8* that belongs to the device family of Cyclone II, we obtain some basic different information between the unimproved algorithm and improved algorithm when contrasting two reports in the platform. The results are shown in Table I.

TABLE I. THE COMPARISON OF PARAMETERS

	Total logic elements	Total registers	Total pins
unimproved	1511	674	389
improved	1931	1320	102

Above table shows that the logic elements of the new improved structure increase and the total registers is more than twice of the original quantity. The reason lies on the segmentation of data in the Round Transformation. The pipelining process of four 32-bit packets data needs more registers than before. A certain clock delay will be produced in the encryption process, because of the processing mode of packets. So the pipelining technology is used in the round transformation, ensuring that the encryption speed meets the actual demand.

In summary, the chip pins are greatly reduced at the expense of certain cost of clock delay, so that the area of chip can be optimized. Since power consumption has a direct link with the area, the power consumption is also decreased, which will be manifested later in the synthesis verification.

C. Synthesis Verification of New Algorithm

The synthesis verification of new algorithm is based on HJTC0.18um CMOS in the ASIC design and verification platform provided by Synopsys Company.

In the synthesis simulation we can find that the cell area, dynamic power, and data require time of the encryption device have significantly changed. It is shown in Table II.

TABLE II. COMPARISON IN ENCRYPTION CHIP PARAMETERS

	Cell area	Total Dynamic Power	Data require time
unimproved	200504.48 *10-12m	22.0214mW	6.00ns
improved	52131.166 *10-12m	14.0253mW	10.74ns

The pipelining technology and 32-bit packet segmentation greatly reduces the area of the chip.

Dynamic power consumption accounts for the majority of the circuit power consumption, and the dynamic power is relatively reduced compared with the unimproved algorithms, and the encrypted rate decreases. However, this clock delay is acceptable and still meets the application requirement.

IV. CONCLUSION

A FPGA implementation of area-optimized AES algorithm which meets the actual application is proposed in this paper. After being coded with Verilog Hardware Description Language, the waveform simulation of the new algorithm was taken in the ModelSim SE PLUS 6.0 and Quartus Ⅱ 7.2 platform. Ultimately, a synthesis simulation of the new algorithm has been done.

The result shows that the design with the pipelining technology and special data transmission mode can optimize the chip area effectively. Meanwhile, this design reduces power consumption to some extent, for the power consumption is directly related to the chip area. Therefore the encryption device implemented in this method can meet some practical applications.

As the S-box is implemented by look-up-table in this design, the chip area and power can still be optimized. So the future work should focus on the implementation mode of S-box. Mathematics in Galois field (2^8) can accomplish the bytes substitution of the AES algorithm, which could be another idea of further research.

ACKNOWLEDGMENT

This paper is supported by the Fundamental Research Funds for the Central Universities (21610512), Guangdong Province Enterprise Commissioner action Plan Project (2009B090600127) and 2010 Guangzhou Science and Technology Support Project (Video format transcoding algorithm research and AVS transcoder design).

REFERENCES

[1] J.Yang, J.Ding, N.Li and Y.X.Guo,"FPGA-based design and implementation of reduced AES algorithm" IEEE Inter.Conf. Chal Envir Sci Com Engin(CESCE).,Vol.02, Issue.5-6, pp.67-70, Jun 2010.

[2] A.M.Deshpande, M.S.Deshpande and D.N.Kayatanavar,"FPGA Implementation of AES Encryption and Decryption"IEEE Inter.Conf.Cont,Auto,Com,and Ener., vol.01,issue04, pp.1-6,Jun.2009.

[3] Hiremath.S. and Suma.M.S.,"Advanced Encryption Standard Implemented on FPGA" IEEE Inter.Conf. Comp Elec Engin.(IECEE),vol.02,issue.28,pp.656-660,Dec.2009.

[4] Abdel-hafeez.S.,Sawalmeh.A. and Bataineh.S.,"High Performance AES Design using Pipelining Structure over GF(2^8)" IEEE Inter Conf.Signal Proc and Com.,vol.24-27, pp.716-719,Nov. 2007.

[5] Rizk.M.R.M. and Morsy, M., "Optimized Area and Optimized Speed Hardware Implementations of AES on FPGA", IEEE Inter Conf. Desig Tes Wor.,vol.1,issue.16,pp.207-217, Dec. 2007.

[6] Liberatori.M.,Otero.F.,Bonadero.J.C. and Castineira.J. "AES-128 Cipher. High Speed, Low Cost FPGA Implementation", IEEE Conf. Southern Programmable Logic(SPL),vol.04,issue.07,pp.195-198,Jun. 2007.

[7] Abdelhalim.M.B., Aslan.H.K. and Farouk.H. "A design for an FPGA-based implementation of Rijndael cipher",ITICT. Ena Techn N Kn Soc.(ETNKS), vol.5,issue.6,pp.897-912,Dec.2005.