RESEARCH PAPER | Engineering | Volume : 3 | Issue : 7 | July 2013 | ISSN - 2249-555X

Implementation of Aes For Image Cryptography Process on Sopc with Area Optimization

KEYWORDS: FPGA, PIPELINING

Mr. Amol Ananda Gore
Electronics & Telecommunication Department, Sinhgad Institute of Technology, Lonavala, India.

Prof. V. V. Deotare
Electronics & Telecommunication Department, Sinhgad Institute of Technology, Lonavala, India.

ABSTRACT *A FPGA implementation of Advanced Encryption Standard for Image Encryption/ Decryption is proposed in this paper. Pipelining is used to maintain speed of operation. The plaintext, initial key and output cipher text are 128 bit length, divided into four 32 bit units controlled by clock. This FPGA chip implementation is embedded in Xilinx Spartan3E with microblaze processor for Image encryption/decryption applications.*

1. INTRODUCTION

The AES ([1]) is a symmetric block cipher that was announced an U.S. Federal Information Processing Standard 197 (FIPS 197) by National Institute of Standards and Technology on November 26, 2001 after a 5-year standardization process. As for now (year 2008) it is still considered to be sufficiently secure and has been officially approved for protecting secret and top secret information by the US Government ([2]).

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 VHDL. 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.

Functional Simulation and Timing Analysis of this algorithm has done in Modelsim SE and Xilinx ISE 9.2i. This core with microblaze processor is embedded in same FPGA chip using RS232 interface with PC to obtain a prototyped image encryption/decryption system.

1. AES ALGORITHM

A. AES Encryption process

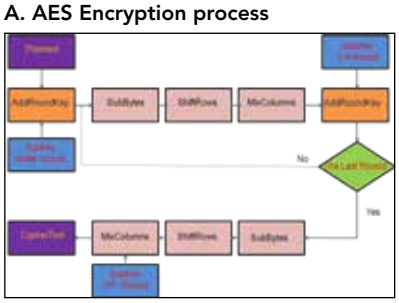


Fig. 1
AES Encryption process includes the following steps described.

Key Expansion:
In this step the master key is expanded into a total of 11 subkeys of 16 words of 8 bits. The first key is called the initial key and the remaining keys are called round keys.

AddRound key:
It is the simple step which performs an Exclusive-OR operation between state matrix and subkey during the each round.

Byte Substitution:
It is the nonlinear transformation that operates on each byte independently using the Sbox table.

ShiftRow:
In this step there is no shifting on first row. While second row is shifted cyclically by one byte, third row is shifted by 2 and fourth row is shifted by 3 bytes.

MixColumns:
This transformation works on the input matrix column by column that is of 32 bits each

B. Improved Algorithm Design

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 selects Nb bits of round key from the expanded key module. Round Transformation involves four modules by ByteSubstitution, ByteRotation, MixColumn and AddRoundKey.

The inputs of plaintext and initial key, intermediate inputs and outputs of round transformation, as well as the output of cipher text 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.

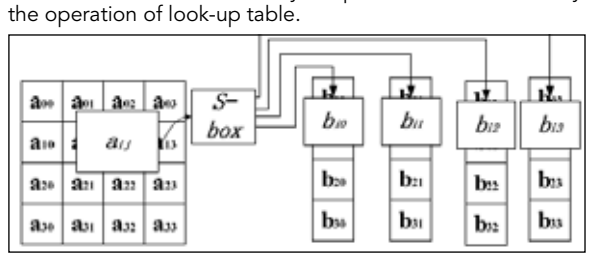


Fig.2

Volume : 3 | Issue : 7 | July 2013 | ISSN - 2249-555X

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 cipher text 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. New Algorithm Process

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.

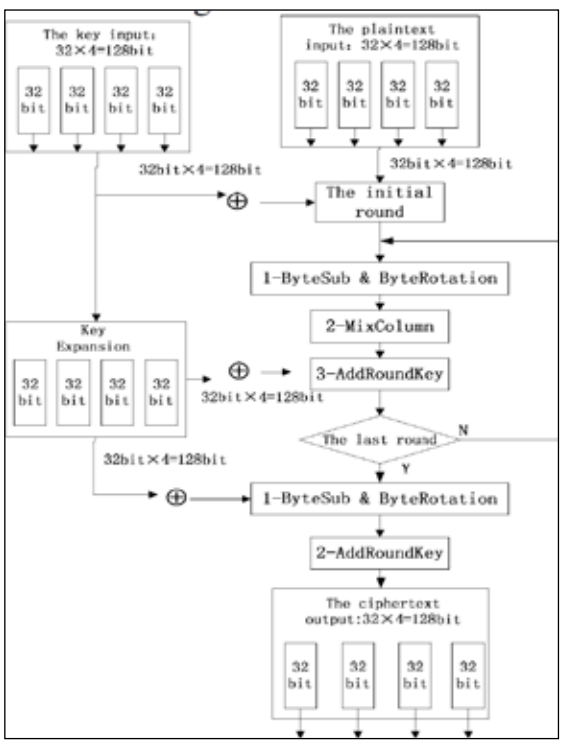


Fig. 3

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

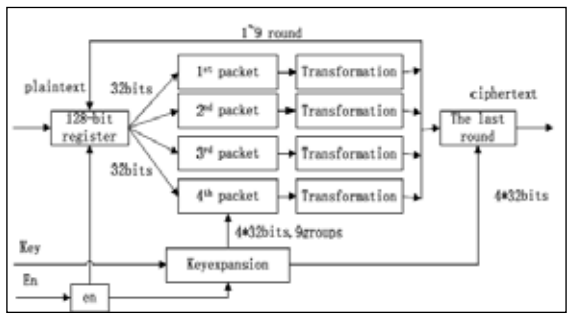
The initial round of encryption:

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(28). 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.



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.

2. FUNCTIONAL SIMULATION AND VERIFICATION

The new improved structure of AES-128 encryption algorithm is implemented with VHDL. We used ModelSim SE for the waveform and verified the results.

Slices	Throughput (Gbps)	Throughput/ Area (Mb/Sec/ Slic)	Support	Ref
626	3.4	5.43	Enc	[6]
1470	2.8	1.9	Enc/Dec	[7]
751	4.0	5.33	Enc	Serial Implementation
573	5.25	9.16	Enc/Dec	Pipelined Implementation

Table I. Comparison of FPGA implementations of AES

3. CONCLUSION

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

Volume : 3 | Issue : 7 | July 2013 | ISSN - 2249-555X

directly related to the chip area. Therefore the encryption device implemented in this method can meet some practical Applications like image encryption.

REFERENCE | [1] AI-WEN LUO, QING-MING YI, MIN SHI "Design and Implementation of Area-optimized AES Based on FPGA" | Published by 2011 International Conference on Business Management and Electronic Information. | | [2] Kuo-Huang Chang1, Yi-Cheng Chen2, Chung-Cheng Hsieh1, Chi-Wu Huang2 and Chi-Jeng Chang1 "Embedded a low area 32-bit AES for image encryption/decryption application" | Published by IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009. | | [3] Ahmed, S.; Samsudin, K.; Ramli, A.R.; Rokhani, F.Z. "Effective implementation of AES-XTS on FPGA" Published in TENCON 2011 - 2011 IEEE Region 10 Conference | | [4] Kaur, Swinder; Vig, Renu, "Efficient Implementation of AES Algorithm in FPGA Device" | Published in International Conference onConference on Computational Intelligence and Multimedia Applications 2007. | [5] Shanxin Qu; Guochu Shou; Yihong Hu; Zhigang Guo; Zongjue Qian "High Throughput, Pipelined Implementation of AES on FPGA" Published in International Symposium on Information Engineering and Electronic Commerce, 2009. IEEC '09. | [6] Helion Technologies Ltd, "Fast AES XTS/CBC Core for Xilinx FPGA" – (XEX-based Tweaked Codebook with Ciphertext Stealing), IP Core, | http://www.heliontech.com/aes_xex.htm. | | [7] Hatzidimitriou, E.; Kakarountas, A.P.; , "Implementation of a P1619 crypto-core for Shared Storage Media," MELECON 2010 - 2010 15th | IEEE Mediterranean Electrotechnical Conference , vol., no., pp.597-601, | | | | | | [8] M. Dworkin, Recommendation for Block Cipher Modes of Operation: | The XTS-AES Mode for Confidentiality on Storage Devices, NIST Special Publication 800-38E, US Nat'l Inst. of Standards and Technology, 2010; http://csrc.nist.gov/publications/nistpubs/800 38E/nist-sp-800-38E.pdf. | | [9] Martin, L.; "XTS: A Mode of AES for Encrypting Hard Disks," Security & Privacy, IEEE , vol.8, no.3, pp.68-69, May-June 2010 |