Volume 4, Issue 2, April 2017 ISSN: 2348-8565 (Online)

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Performance Evaluation of the AES Algorithm

Pragya Mishra
M. Tech. Research Scholar
Lakshmi Narain College of Technology
Jabalpur (M.P.), [INDIA]
Email: pragya.mishra1231@gmail.com

Prajyant Pathak
Head of the Department
Department of Electronics & Communication Engg.
Lakshmi Narain College of Technology
Jabalpur (M.P.), [INDIA]
Email: prajyant.pathak@yahoo.com

ABSTRACT

The confidentiality of the information stored in computer systems and sent through networks is a matter of primary concern. Generally, confidentiality is obtained by encrypting/decrypting the information with a symmetric algorithm. Currently, the most used and standardized algorithm is the Advanced Encryption Standard (AES), but the encryption and decryption usually causes undesired delays in the access to information. As users must necessarily use either AES or another encryption/decryption algorithm to guarantee confidentiality, the implications on performance must always be evaluated. It is very interesting to evaluate the influence of the configuration parameters of AES on performance, in order to select an appropriate configuration. This work provides a performance evaluation methodology to estimate how the configuration of any encryption/decryption algorithm affects the performance. The methodology has been applied to the AES algorithm in five different execution platforms, obtaining useful results for any user of the AES algorithm.

Keywords:— *performance evaluation; AES algorithm; encryption performance; AES configuration.*

I. INTRODUCTION

Confidentiality of information is an issue of primary importance To guarantee confidentiality, the generator (transmitter) must encrypt the information and the user(receiver) must decrypt it. This process of encryption and decryption is carried out with symmetric cryptographic algorithms, like DES (Data Encryption Standard), 3DES (Triple DES), Blowfish, Two fish, RC4, RC6, CAST, Advanced Encryption Standard (AES), etc. Currently, AES is one of the most commonly used algorithms. In the year 2000 this algorithm won the competition promoted by NIST (National Institute of Standards and Technology of the US Department of Commerce) to design a cryptographically strong encryption/decryption algorithm. In 2001, AES was adopted as a FIPS standard [1] (Federal Information Processing Standards) for use in the US Federal Administration. Until now, no security flaws have been found in the AES algorithm, and it will take many years for computers to reach the computational power required to realize a brute force attack in a reasonable period of time. For these reasons, this evaluation work will focus exclusively on the AES algorithm. Fig.1 shows the overall encryption process with AES. The input to the algorithm is a single 128-bit block of plain text and the output is also a single 128-bit block of cipher text. The encryption process consists of multiple rounds, and the number of standard (intermediate) rounds depends on the key length: 9 for a 128-bit key, 11 for a 192-bit key, and 13 for a 256-bit key. The initial round only contains 1 transformation, but the intermediate rounds contain 4 transformations, and the final round contains 3 transformations. The Add_Round_Key transformation is a simple bitwise XOR of the current information state with the initial key or an expansion (transformation) of the initial key. The Sub_Bytes transformation substitutes each byte of the state for a new byte using a pre-calculated table called a Substitution Box (S-Box). Its objective is to introduce confusion in the information state.T e Shift_Rows transformation carries out a circular displacement of the bytes within each row of the information state. Its objective is to introduce diffusion in the state. The Mix_Columns transformation mixes the bytes within each column of the state to generate new values for the bytes of the column. Its objective is to introduce additional diffusion in the information state. The AES algorithm always encrypts 16 bytes of plain text and generates 16 bytes of cipher text. If the information to encrypt is larger than 16 bytes, it must be divided in blocks of 16 bytes that are encrypted sequentially or in parallel. Furthermore, if the size of the information to encrypt is not multiple of 16, the last block must be padded.

II. METHODOLOGY

Composite Field Arithmetic S-Box:

The Sub Bytes transformation is a non-linear operation in AES wherein each byte of a state is mapped to a different value. The Sub Bytes transformation is done through S-box. There are two techniques to perform substitutions, (i) using ROM table, and (ii) using composite field arithmetic. The Sub Bytes transformation, done through S-box mapping is computationally inefficient when implemented using a ROM. But, it is not efficient for applications requiring very high throughput as ROM accessing involves one complete clock cycle for mapping one 8-bits state element and consequently 16 clock cycles are required to transform the 128 bits data (16 bytes).

To increase the throughput, parallel ROMs are required resulting in large size of chip area. Therefore, a more feasible solution is to implement an S - box is by using composite field arithmetic which uses only logic elements in the implementation. Substitution is the most complex steps in terms of cost and implementation. Therefore, its hardware optimization for VLSI implementation is very important to reduce the area and power of the AES architecture. The ROM based approach requires high amount of memory and also it causes low latency because of ROM access time. Therefore, composite field arithmetic is more suitable for S-box (substitution) implementation.

The Speed improvement along with an area reduction has been the most challenging research in VLSI implementation. We propose high speed VLSI architecture for S-box. The FPGA implementation of the architecture is done along with comparison with some existing transformation techniques. The proposed architecture has delayed improvement and low power consumption.

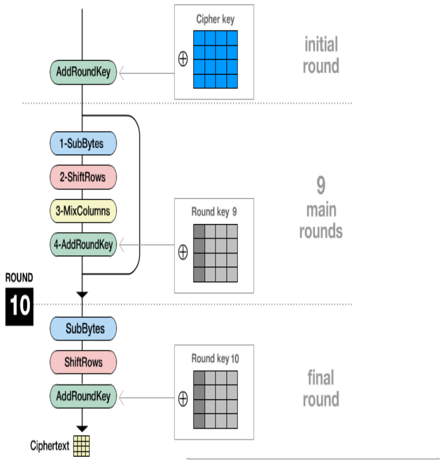


Figure 1: Encryption Process

Inner Working of Rounds

The algorithm initiates from Add round key process followed by total of 9 iterations of four processes and the 10th iteration of 3 processes. This is applicable for both encryption and decryption including a special case that each process of an iteration the decryption algorithm is the reverse of its corresponding process from encryption algorithm. 4 processes used are as given

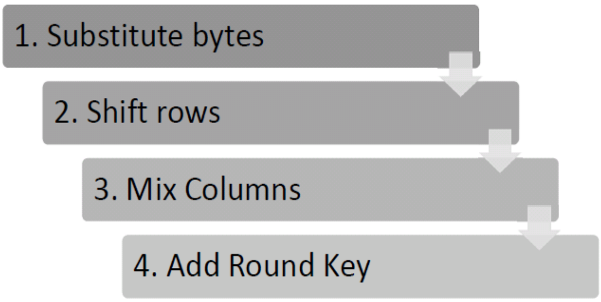

1. Substitute bytes
2. Shift rows
3. Mix Columns
4. Add Round Key

Figure 2 : Four Stages of Encryption

The 10th iteration just doesn't use the Mix Columns transformation. The decryption algorithm initiates from an Add round key process trailed by 9 iterations of decryption process which comprises of the subsequent processes shown in figure

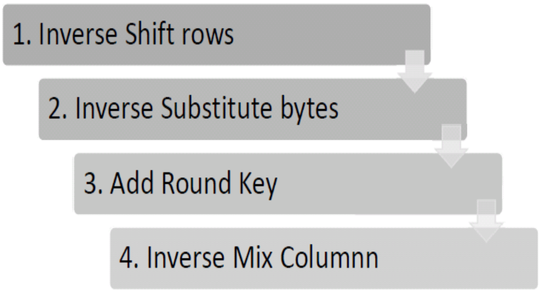

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Add Round Key
4. Inverse Mix Columnn

Figure 3 : Four Stages of Decryption

Byte Substitution

Byte Substitution is basically a lookup table utilizing a 16×16 double dimension of byte values known as **s-box**. This dimension comprises of every conceivable combos of 8 bit sequence (2^8=16 × 16 = 256). Nonetheless, the s-box isn't only an arbitrary stage of these qualities and there is an overall-characterized technique for making the s-box matrix. The architects of Rijndael demonstrated how this was carried out dissimilar to the s-box DES for which not at all justification was given. We won't be excessively interested here how the s-boxes are created and can basically take them as lookup tables. Again the dimension that gets worked upon all around the encryption is called state-matrix.

III. Synthesis & Simulation

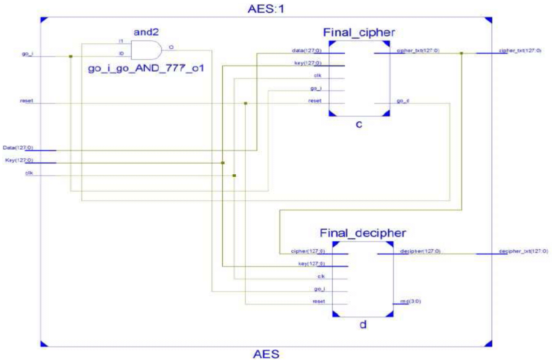
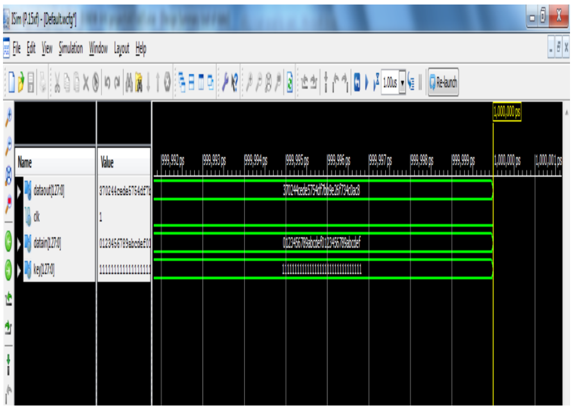
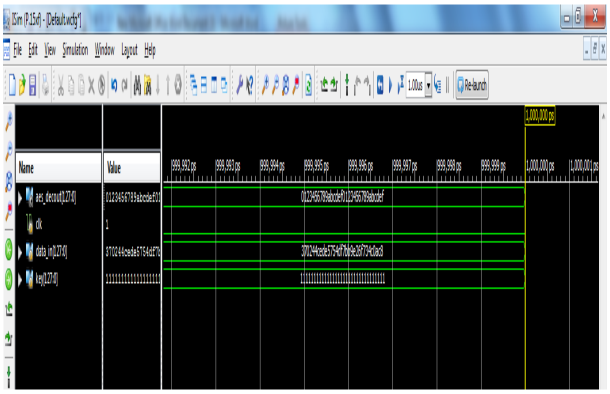


Figure 4 : Internal Schematic of AES block



Name | Value
dataout[127:0] | 370244cede5754df7b | 370244cede5754df7bb9e26f734c0ac8
clk | 1
datain[127:0] | 0123456789abcdef01 | 0123456789abcdef0123456789abcdef
key[127:0] | 1111111111111111 | 11111111111111111111111111111111

Figure 5 : AES Encryption

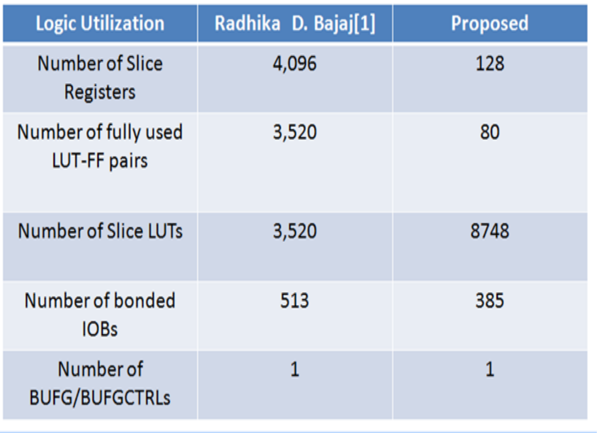


Name | Value
aes_decout[127:0] | 0123456789abcdef01 | 0123456789abcdef0123456789abcdef
clk | 1
data_in[127:0] | 370244cede5754df7b | 370244cede5754df7bb9e26f734c0ac8
key[127:0] | 1111111111111111 | 11111111111111111111111111111111

Figure 6: AES Decryption

IV. Result

Table 1: Comparative Result



Logic Utilization	Radhika D. Bajaj[1]	Proposed
Number of Slice Registers	4,096	128
Number of fully used LUT-FF pairs	3,520	80
Number of Slice LUTs	3,520	8748
Number of bonded IOBs	513	385
Number of BUFG/BUFGCTRLs	1	1

V. Conclusion

We have proposed optimized VLSI architecture of S-box for AES algorithm. The architecture of s-box in composite field has been modified in order to have high speed and low areas. This thesis was successfully completed with the implementation of AES algorithm on 128 bit message. The encrypted cipher text and the decrypted text are analyzed and proved to be correct. The encryption efficiency of the proposed AES algorithm was studied and met with satisfactory results.

REFERENCE:

[1] Radhika D.Bajaj1, Dr. U.M. Gokhale2, Design and Simulation of AES Algorithm for Cryptography, ISSN 2321 3361 © 2016 IJESC.

[2] A.P. Anusha Naidu, B. Prof (Mrs.) Poorvi K. Joshi, FPGA Implementation of Fully Pipelined Advanced Encryption Standard, IEEE ICCSP 2015 conference.

[3] Murtada. M. Abdelwahab and Abdelrasoul. J. Alzubaidi, VLSI implementation of Advance Encryption Algorithm using index technique, International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering, 2015.

[4] Daniel F. García, Performance Evaluation of Advanced Encryption Standard Algorithm, Second International Conference on Mathematics and Computers in Sciences and in Industry 2015.

[5] Rafidah ahmad and widad ismail, Implementation of high performance Advanced Encryption Standard -128 for wimax application on FPGA, IEEE 2014.

[6] Ritu Pahal and Vikas Kumar, Efficient Implementation of AES, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013.

[7] M. I. Soliman, G. Y. Abozaid, "FPGA implementation and performance evaluation of a high throughput crypto coprocessor," *Journal of Parallel and Distributed Computing*, Vol. 71 (8), pp.1075-1084, Aug. 2011

[8] V. K. Pachghare, Cryptography and information security, E. E. Ed., PHI Learning, New Delhi, 2009.

[9] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard Using Composite Fields," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 19 (1), pp. 85-91, Jan. 2011.

[10] X. Zhang, K. K. Parhi, "High-Speed VLSI Architectures for the AES

Algorithm," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12 (9), pp. 957-967, Sep. 2004.

[11] M. Jridi and A. AlFalou, "A VLSI implementation of a new simultaneous images compression and encryption method," 2010 IEEE International Conference on Imaging Systems and Techniques (IST), pp.75 -79, July 2010.

[12] Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, and Cheng-Wen Wu, "A High- Throughput Low-Cost AES Processor," IEEE Communications Magazine, Vol.41 (12), pp.86-91, Dec. 2003.

* * * * *