ISSN 2320-9984

Volume 5 Issue 2 | June 2017

ISSN: 2320-9984 (Online)

International Journal of Modern Engineering & Management Research

Website: www.ijmemr.org

An Efficient FPGA Implementation of the Advanced Encryption Standard (AES) Algorithm Using S-Box

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 goal of every encryption algorithm is to make it as difficult as possible to decrypt the generated cipher text without using the key. If a really good encryption algorithm is used, there is no technique significantly better than methodically trying every possible key. For such an algorithm, the longer the key, the more difficult it is to decrypt a piece of cipher text without possessing the key. There are mainly two types of encryption algorithms, a private key (also called symmetric key having a single key for encryption and decryption) and public key (separate key for encryption and decryption). In terms of computational complexity, private key algorithm is less complex than a public key algorithm. The simple architecture of private key algorithm attracts the VLSI implementation through the basic digital components like basic gates and flip-flops.*

Keywords:**—***S-Box, FPGA, encryption, decryption, AES, SubBytes, ShiftRows, MixColumns.*

1. INTRODUCTION

AES algorithm is an iterative algorithm, which requires many computation cycles. A software platform cannot provide the high speed encryption of data, specially used for realtime applications. Audio/video content encryption is required in real-time in business deals via video conferencing. Therefore, dedicated hardware implementation is inevitable in such applications. Hardware implementation can be done through different architectures trading throughput with area and power consumption. At any time, designing best architecture for a particular design with low area and low latency is a challenge. Hardware implementations of the AES algorithm vary according to the application. While some applications require very high throughputs as in e-commerce servers, others require a medium throughput range as in designs for cell phones. Some others require very low area and low power implementations to be used application as RFID cards. The AES is an iterative algorithm and uses four operations in different rounds, namely SubBytes, ShiftRows, MixColumns and Key Additions transformations. SubBytes transformation is done through S-box. S-box is the vital component in the AES architecture that decides the speed/throughput of the AES [1]. 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 its hardware optimization for VLSI implementation is very important to reduce the area and power of the AES architecture.

AES algorithm is an iterative algorithm, which requires many computation cycles. A software platform cannot provide the high speed encryption of data, specially used for real-time applications. Audio/video content encryption is required in real-time in business deals via video conferencing. Therefore, dedicated hardware implementation is inevitable in such applications. Hardware implementation can be done through different architectures trading throughput with area and power consumption. The design optimization can be done by replacing conventional modules in AES architecture with a module which best suits for the area and latency reduction details in [3],. Further, there are mainly two different design styles found and its implementation in different devices namely, iterative and concurrent (pipeline) for implementation in Xilinx FPGA [9]. It has observed that concurrent implementation requires less time but the area is large with high power consumption. The transformations used in different rounds are same so, algorithm can be used repeatedly and area and power can save with the improvement in speed [1]. The iterative implementation could be efficient as per the requirement published in [1].

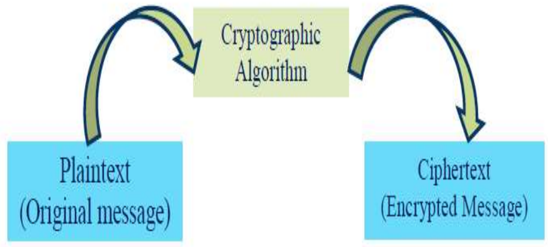


Figure 1 : Basic Step of Encryption in Cryptography

2. METHODOLOGY

There are four transformations in the AES algorithm among all the transformation, SubBytes is complex and non-linear. There are two techniques found to implement S-BOX, one using RAM and another using composite field arithmetic architecture. The implementation of the composite field S-BOX is accomplished using combinational logic circuits rather than using pre-stored S-BOX values. S-BOX substitution starts by finding the multiplicative inverse of the number in GF (28), and then applying the affine transformation. Implementing a circuit to find the multiplicative inverse in the finite field GF (28) is very complex and costly, therefore, [9] has suggested using the finite field GF (24) to find the multiplicative inverse of elements in the finite field GF (28). First detailed implementation of the composite field S-BOX was published in [6]. The S-Box is at the major of any AES implementation and is measured a full complexity design consuming the main portion of the power and energy inexpensive of the AES hardware. The substitute way is to design the S-Box circuit using combinational logic directly from its arithmetic operations. This method has a fine delay - path from S-Box processing. The AES algorithm can be implemented on a varied range of platforms under different constraints [2]. In transportable applications figuring resources are usually restricted and dedicated hardware implementation of the safety purpose is essential. AES Implementation using FPGA (Field Programmable Gate Array) is not appropriate for such applications generally due to size and power limitations. A full-custom chip is more suitable for compact small foot-print design in such a case. The Galois Field arithmetic for S-Box, it is very clearly evident that the implementation of S-Box/InvS-Box needs a large number of XOR operations [5]. The novel XOR has been designed using minimum number of transistors and it has high noise margin and low power consumption as compared to existing XOR designs. The new approach to minimize the silicon - area of S-Box design demonstrated by using a new 2-input XOR gate for low-power composite field arithmetic to reduce the power dissipation and delays for the complete circuit [15].

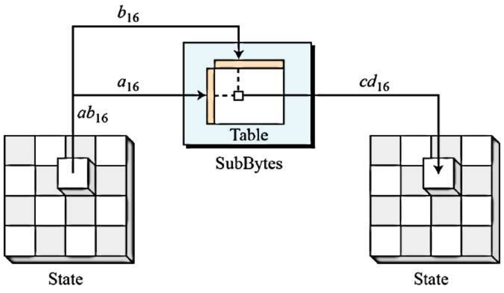
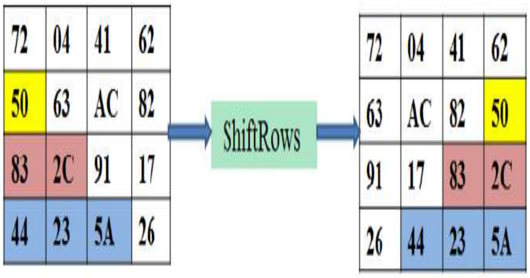


Figure 2 : SubBytes Transformation

Figure indicates that how the transformation can be done. There are two hexadecimal digits a and b in one state element, the left digit (a) defines the row and the right digit (b) defines the column of the substitution table. The junction of these two digits is the new bytes. Inverse SubBytes transformation is inverse of SubBytes transformation. It can find in the similar way only table which is used for mapping the byte is different. The SubBytes transformation is done through S-box. There are two techniques to perform substitutions, (i) using S-BOX table, and using composite field arithmetic. There are separate tables for SubBytes and its inverse; Table II is used for SubBytes transformation and Table used for its inverse and It can be found using S-box architecture in composite field arithmetic.



72	04	41	62
50	63	AC	82
83	2C	91	17
44	23	5A	26

72	04	41	62
63	AC	82	50
91	17	83	2C
26	44	23	5A

Figure 3: Shift Rows Transformation for AES Encryption

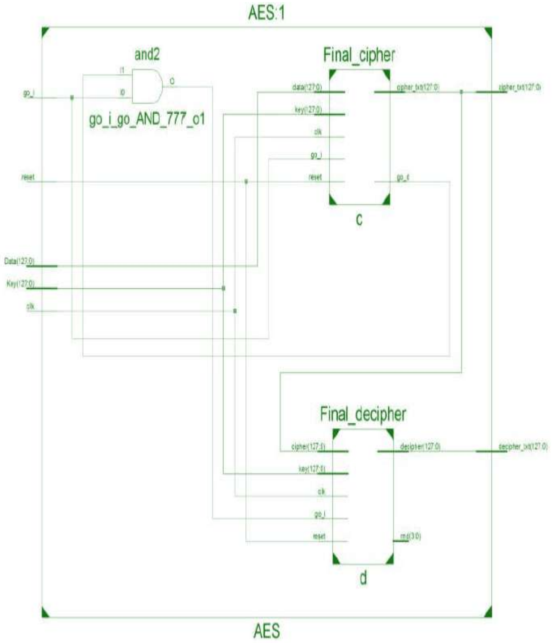


Figure 4: Internal Schematic of Decipher Block.

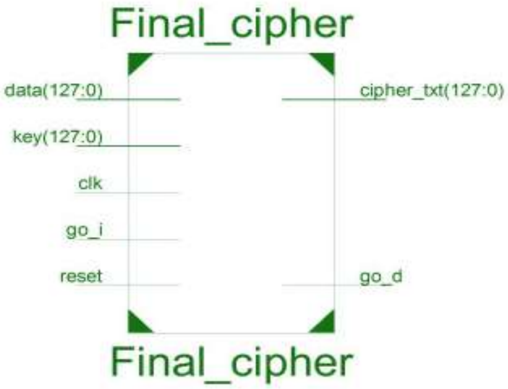


Figure 5: Blocks Schematic of Data Path of Cipher Block.

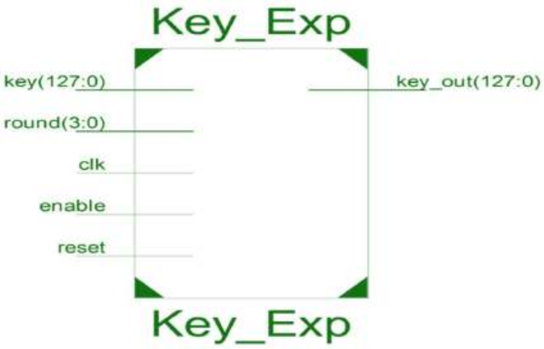
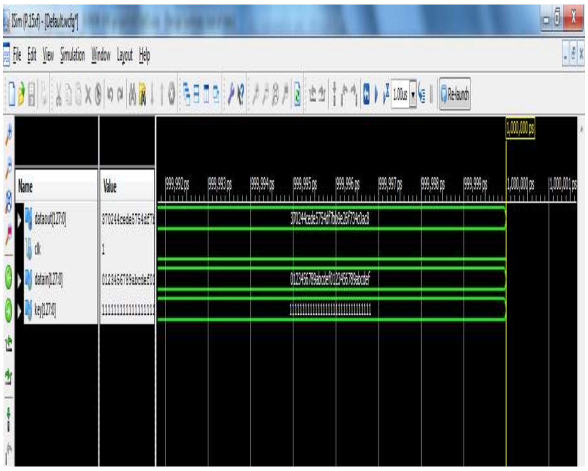


Figure 6 : Block Schematic of Key Expansion Unit

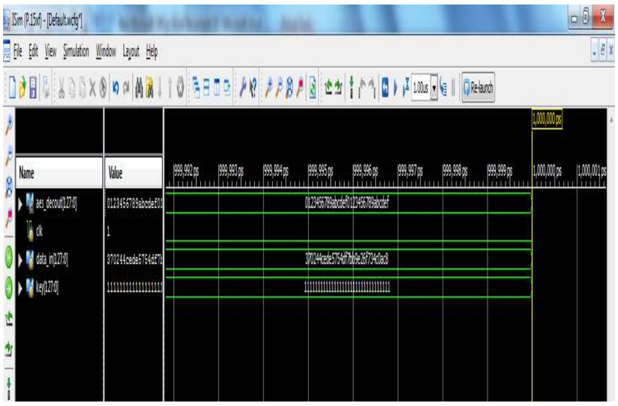
3. SIMULATION RESULTS

We have implemented the complete encryption and decryption modules of Advanced Encryption Standard that is all the four transformations that are used at encryption and at decryption side. The Simulation results of complete AES encryption and AES decryption is shown below in Fig. Here All the transformations have been simulated by using Xilinx ISE Design suite 14.1.



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

Figure 7 : AES Encryption



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

Figure 8 : AES Decryption

4. COMPARATIVE RESULT

Table 1 Comparative Result of Logic Utilization

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

5. 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.

REFERENCES:

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

[2] 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.

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

[4] Rafidah Ahmad and Widad Ismail, "Implementation of high performance Advanced Encryption Standard -128 for Wimax Application on FPGA, IEEE 2014".

[5] 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.

[6] B.A. Forouzan and D. Mukhopadhyay, "Cryptography and Network Security", 2nd Ed., Tata McGraw Hill, New Delhi, 2012.

[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.