International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.18
August-2014,
Pages:3873-3877

www.semargroup.org,
www.ijsetr.com

Efficient Hardware Implementation for Advanced Encryption Standard with 256 Bit Key Length

P. PENCHALA REDDY[1], K. JAGADEESH KUMAR[2], DR. V.THRIMURTHULU[3]

[1]PG Scholar, Dept of ECE, CREC, Tirupati, AP, India, E-mail: penchalreddy1020@gmail.com.
[2]Asst Prof, Dept of ECE, CREC, Tirupati, AP, India, E-mail: jagadish.kasula@gmail.com.
[3]Professor, Dept of ECE, CREC, Tirupati, AP, India, E-mail: vtmurthy.v@gmail.com.

Abstract: Increasing need of high security in communication led to the development of several cryptographic algorithms hence sending data securely over a transmission link is critically important in many applications. Hardware implementation of cryptographic algorithms are physically secure than software implementations since outside attackers cannot modify them. This paper presents an efficient hardware architecture design & implementation of Advanced Encryption Standard (AES) – Rijndael cryptosystem using Spartan3E XC3S1200E FPGA.

Keywords: Rijndael; Cryptography; Hardware Implementation; Security; FPGA; Verilog; Encryption; Decryption.

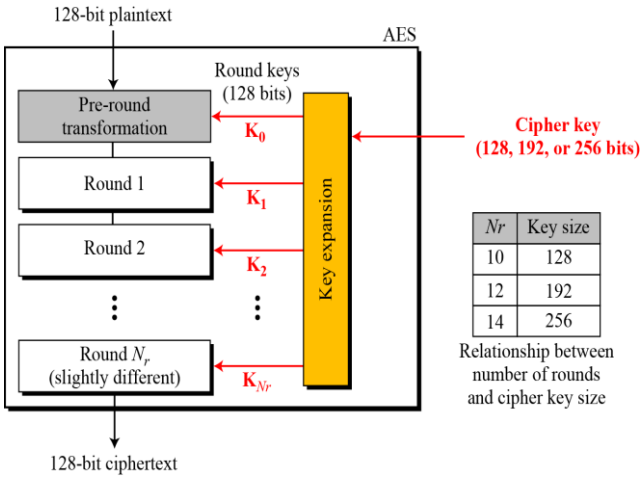
I. INTRODUCTION

Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. It uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key. The Advanced Encryption Standard, in the following referenced[2] as AES, is the winner of the contest, held in 1997 by the US Government, after the Data Encryption Standard was found too weak because of its small key size and the technological advancements in processor power. Fifteen candidates were accepted in 1998 and based on public comments the pool was reduced to five finalists in 1999. In October 2000, one of these five algorithms was selected as the forthcoming standard: a slightly modified version of the Rijndael. There are many architecture proposals for AES Rijndael algorithm [4,5], but many of them are poor in terms of area and speed. This paper proposes a different approach to increase speed by utilizing lesser resources available in FPGA. This paper is structured as follows: Section II describes the existing AES algorithm and Section III describes the proposed method. The result and conclusion are described in Section IV and V respectively.

II. EXISTING AES ALGORITHM

AES use Rijndael algorithm [6] by Joan Daeman and Vicent Rijimen for both encryption and decryption. The AES cryptography algorithm is capable of encrypting and decrypting 128 bit data using cipher keys of 128, 196 or 256 bits (AES128, AES196 and AES256) [5]. The AES is a computer security standard from NIST intended for protecting electronic data. Federal Information Processing Standards (FIPS) Publication 197 gives the specification of AES. Rijndael encryption consist of four operations

1. Substitution
2. Shift Row
3. Mix Column
4. Key Addition



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Fig.1. Algorithm for AES Encryption.

A. Add round key

State is represented as follows (16 bytes):

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Add round key(state,key):

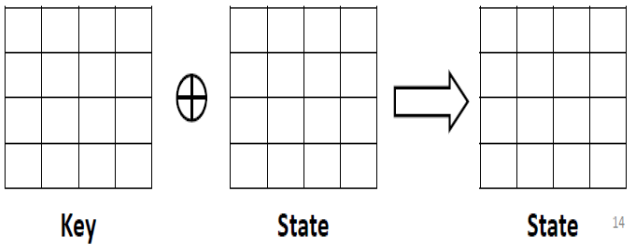


Fig2: Add round key.

B. Sub bytes transformation

Bytes are transformed using a non linear s-box

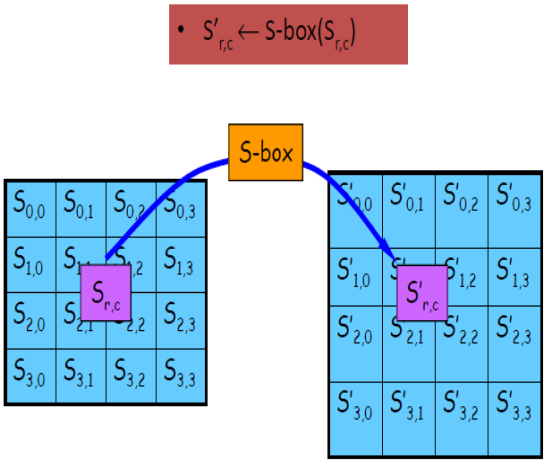


Fig3. Sub bytes transformation.

Byte substitution using a non-linear (but invertible) S-Box (independently on each byte). S-box is represented as a 16x16 array, rows and columns indexed by hexadecimal bits. 8 bytes replaced as follows: 8 bytes define a hexadecimal number rc, then Sr,c = binary(S-box(r, c))

C. Shift rows

Circular Left Shift of a number of bytes equal to the row number

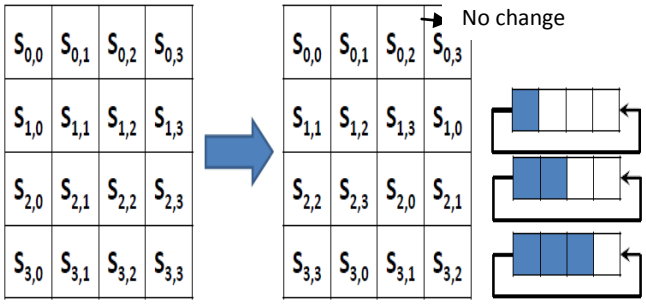


Fig4. Shift rows.

D. Mix column transformation:

Bytes in columns are combined linearly. Interpret each column as a vector of length 4. Each column of State is replaced by another column obtained by multiplying that column with a matrix in a particular field (Galois Field).

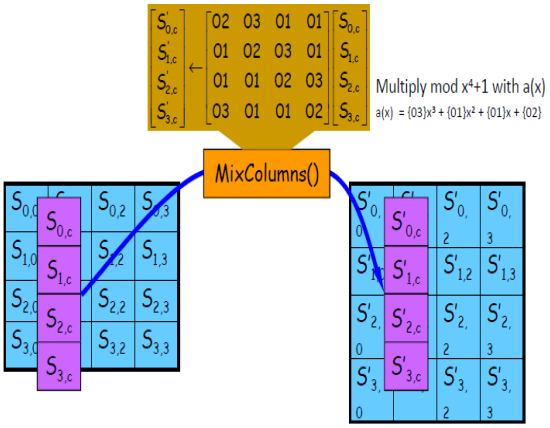


Fig5. Mix column transformation

The Rijandael decryption consists of four inverse operations of encryption which are compliment functions of encryption. They are

1. Inverse Substitution
2. Inverse Shift Row
3. Inverse Mix Column
4. Key addition

III .PROPOSED METHOD

The proposed architecture is designed to get maximum speed and lesser area. The proposed architecture has four parts

1. Mode selection logic
2. Key expansion block
3. Cipher block
4. Decipher block.

Fig5 shows Multi mode AES block diagram which has Key expansion, Mode selection logic, and Encryption and Decryption blocks. Key expansion block functionality is to expand the key for the given key data length depending on the key select length. Start_key_exp is used to start Key expansion for the given key length once its expansion done the key data is used for encryption and decryption block for to encrypt data or decrypt the data respectively. Encryption block is to encrypt the data of block size 128 bit depending on the mode selection logic. Mode selection logic is to select one of the modes out of three they are ECB (Electronic codebook), CBC (Cipher Block Chain) and CTR (Counter mode).

In cryptography, modes of operation enable the repeated and secure use of a block cipher under a single key. A block cipher by itself allows encryption only of a single data block of the cipher's block length. When targeting a variable-length message, the data must first be partitioned into separate cipher blocks. Typically, the last block must also be extended to match the cipher's block length using a suitable padding scheme. A mode of operation describes the process of encrypting each of these blocks, and generally uses randomization based on an additional input value, often called an initialization vector, to allow doing so safely.

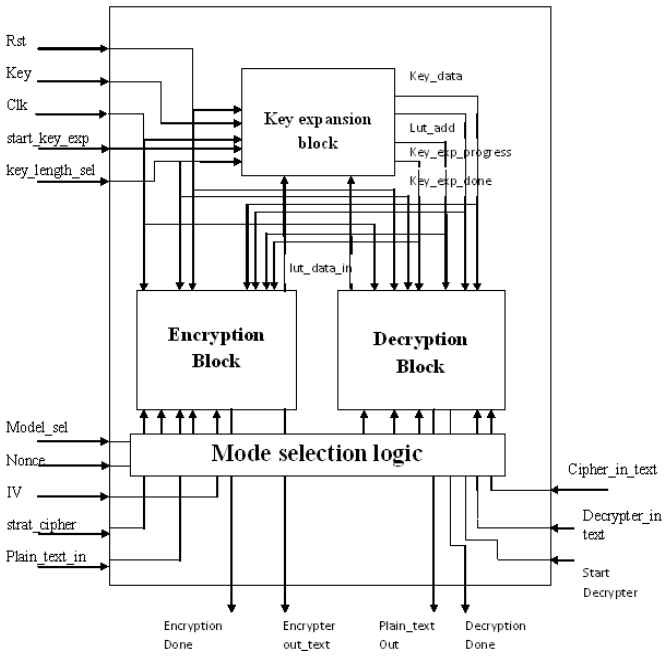


Fig6. Block Diagram of Multimode AES.

Modes of operation have primarily been defined for encryption and authentication. Historically, encryption modes have been studied extensively in regard to their error propagation properties under various scenarios of data modification. Later development regarded integrity protection as an entirely separate cryptographic goal from encryption. Some modern modes of operation combine encryption and authentication in an efficient way, and are known as authenticated encryption modes. Mode selection logic is used select which mode is of operation is to perform. Here we are used ECB, CBC and CTR mode encryption. Each encryption is has its own applications. Mode 00, a mode 01 and mode 1x selection activates ECB, CBC and CTR respectively for encryption and decryption.

A. Mode selection logic

1. Encryption mode selection description

As shown if figure 6 encryption mode selection logic which has three modes of operation they are (ECB, CBC and CTR) in that for mode 00(Electronic code book mode) during encryption process plaintext is directly applied to encryption block and the output is directly connected to the output. The width of the plain text is 128 bit.

Mode 01 is (Cipher block chaining mode). In this mode initialization vector (IV) is XORed with plain text as input to the encryption block and the output cipher text is again given back as IV to the next iteration of encryption.

Mode 1x is (Counter mode) which as nonce of 120bit is concatenated with the counter of 8 bit as input to the encryption block and the output cipher text is XORed with the plain text, from the next iteration count value is incremented and concatenated with the same nonce value to input of encryption block. Encryption output is XORed with plain text.

In CTR mode both encryption and decryption is done with the same encryption block. During decryption process start decipher and CTR mode is selected then the same encryption block used by giving data to the same encryption block as shown in figure 6.

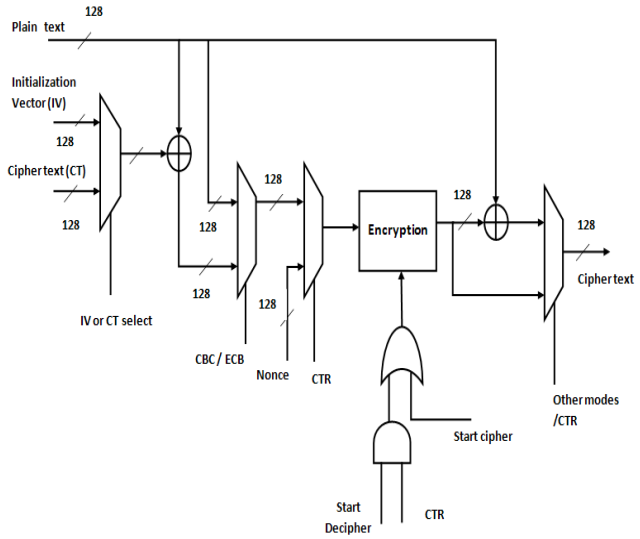


Fig7. Encryption mode selections

2. Decryption mode selection description

In decryption mode selection logic for mode 00(ECB) cipher text of 128 bit data is applied to the decryption block and the output of that decryption block is directly connected to the output pin in ECB mode as shown in figure 7

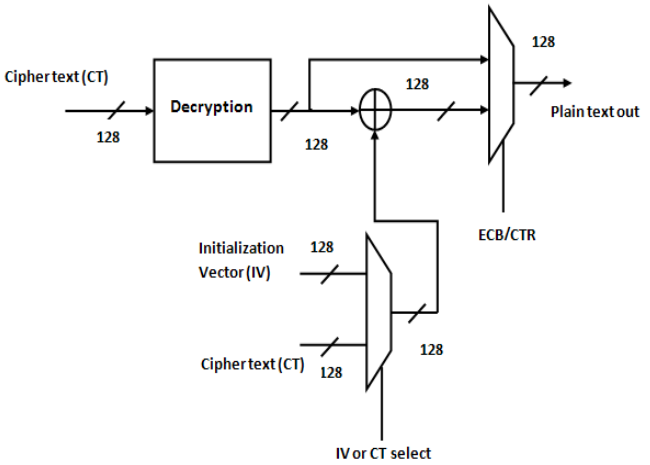


Fig8. Decipher mode selection logic

For mode 01 (CBC) cipher text is given to decryption block the output from the block is XORed with IV to get the plain text out. From the next iteration previous cipher in text is given as IV for getting the plain text out. The AES algorithm takes the Cipher Key, ***K***, and performs a Key Expansion routine to generate a key schedule. The Key Expansion generates a total of ***Nb*** (***Nr*** + 1) words: the algorithm requires an initial set of ***Nb*** words, and each of the ***Nr*** rounds requires ***Nb*** words of key data. The resulting key schedule consists of a linear array of 4-byte words, denoted [*wi*], with *i* in the range 0< = *i* < ***Nb***(***Nr*** + 1).

B. Key Expansion block description

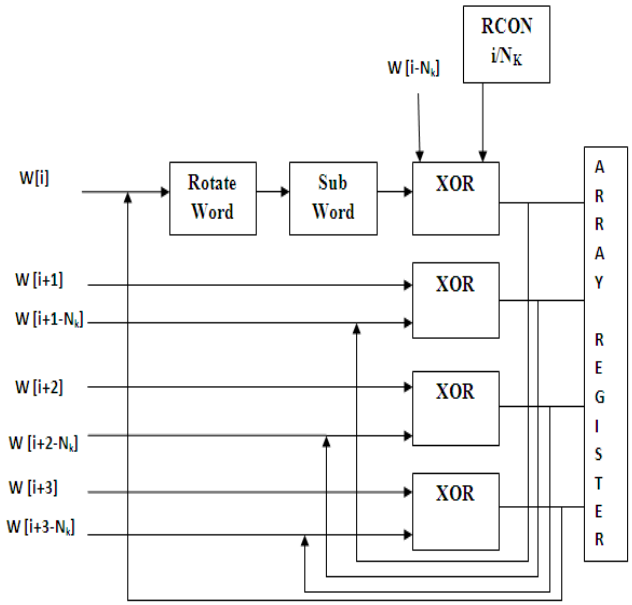


Fig9. Key expansion block diagram.

In each round of key expansion it has to pass through rotate, sub word, or with RCON and XOR with W[i-Nk].Each round final value is stored in array register to start the next iteration of the key expansion.

C. Cipher Block description

Cipher block contain add round, shift rows, sub bytes and mix columns rounds. In encryption it has total Nr number of rounds. During the first round input data is XORed with key data of size 128, then the result data is applied through an iterative process of shift rows, SubBytes, mix columns and XOR with key data till number of rounds equal to Nr-1. During the last round mix column step is skipped.

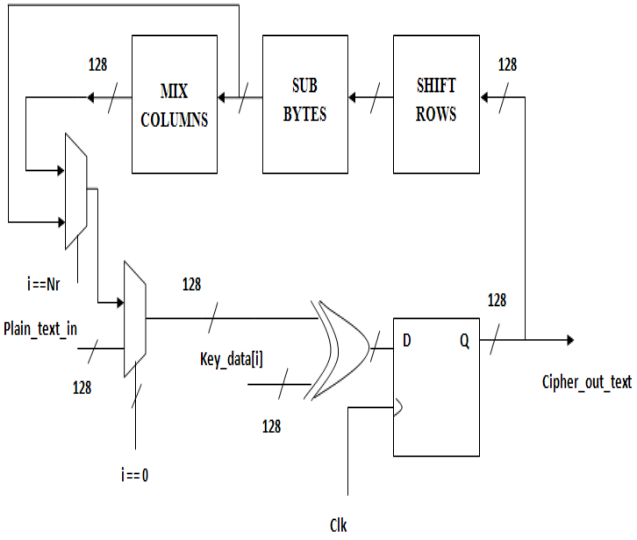


Fig10. Cipher block

D. Decipher block description

The individual transformations used in the Decipher are Inv Shift Rows (), Inv Sub Bytes (), Inv Mix Columns (), and Add Round Key ().

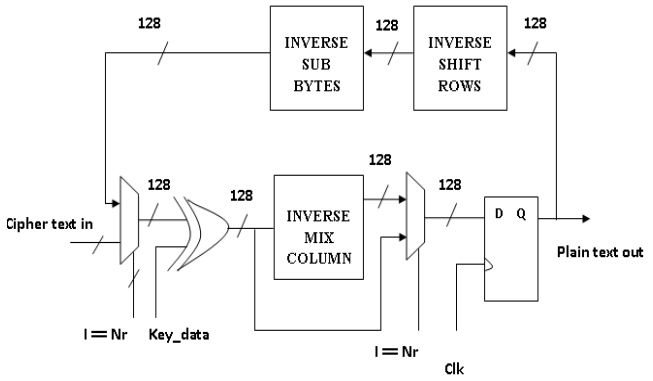


Fig.10 Decipher block.

During decryption process first round cipher in text XORed with key data when number of rounds equal to Nr, second round onwards till Nr –1 rounds Inv Shift Rows (), Inv Sub Bytes (), Add Round Key () and Inv Mix Columns () as shown in figure below. Flip flops are used to store 128 bit data in each round; stored data is used in each start of round. During the final round mix column step is skipped. During inverse cipher the number of rounds is same as cipher block rounds but it is not exactly reverse of it. In inverse cipher after sub bytes key data is XORed with sub byte output then it is given to inverse mix column.

IV. SIMULATION RESULTS

Advanced Encryption Standard (AES) is a symmetric key cipher technique used to secure and encrypt operating systems, hard drives, networking systems, files, e-mails, and other similar data. In cryptography, AES consist of three block ciphers taken from a larger collection published originally as Rijndael. Each cipher has a 128-bit block size with three different key sizes of 128, 192, and 256 bits. After expansion of Key length 128 or 192 or 256 bits stored in the memory.

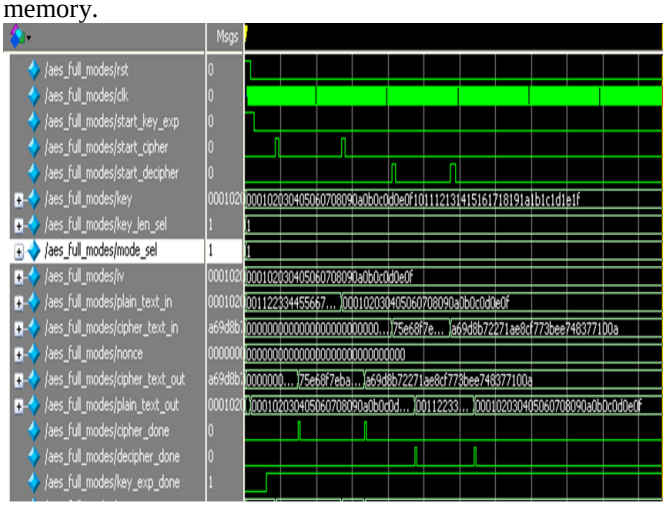


Fig11. Simulation Results.

The combination of a simple, portable and efficient AES cryptographic algorithm implemented in Verilog source code provides an excellent platform for high security applications. A synthesizable Verilog code is developed for the

implementation of both encryption and decryption process with different modes. Design has features such as low cost high speed and low area efficient. Each program simulation results are verified with ModelSim PE and are synthesized in Xilinx ISE 9.2i and are Implemented on Spartan3E XC3S1200E FPGA.

V. REFERENCES

[1] M. Goswami and S. Kannojiya, "High Performance FPGA Implementation of AES Algorithm with 128-Bit Keys," Proc. IEEE Int. Conf. Advances Computing Comm., vol. 1, Himarpur, India, 2011, pp. 281-286

[2] FIPS-197, NIST - National Institute of Standards and Technology," Announcing the Advanced Encryption Standard (AES)," http://csrc.nist.gov/publications/fips/ fips197/fips-97.pdf, 2001.

[3] Abhijith P.S. Mallika Srivastava and Aprna mishra, "High performance hardware implementation of AES using minimal resources," International conference on Intelligent systems and signal processing, 2013.

[4] U. Kretzschmar, A. Astarloa, J. Lazaro, U. Bidarte and J. Jimenez, "Robustness analysis of different AES implementations on SRAM based FPGAs," Int. Conf. on Reconfigurable Computing and FPGAs 2011, pp. 255-260.

[5] W. Stallings, "Cryptography and network security principles and practice," Pearson edition 2009, pp. 135-160.

[6] H. Trang and N.V. Loi, "An efficient FPGA implementation of the Advanced Encryption Standard algorithm," IEEE Int. Conf. on Computing and Communication Technologies, Research, Innovation and Vision for the Future (RIVF), 2012, pp. 1-4.

[7] An FPGA Implementation of 30Gbps Security Module for GPON Systems BY Truong Quang Vinh1, Ju-Hyun Park1, Young-Chul Kim1, Kwang-Ok Kim2 2008 IEEE.