Designing and Implementing FPGA for AES

Mihir Narayan Mohanty

Abstract— The execution of DES and triple DES is not possible on hardware platform because they consume huge memory space. We can use field programmable gate arrays in order to do the hardware implementation because of its low charge, advertising space and reconfiguration nature. This paper aims at reducing the delay by using pipeline for speeding up the process. The proposed pipeline structure has a characteristic of having round keys which during iterations of encryption are utilized and an encryption method is used for generating them in parallel. The overall delay related to a delay of coding of plaintext block is reduced. The simulation is done in VHDL by Xilinx and implementation is done on FPGA Spartan 3E.

KeyWords: AES, piplelining, Cryptography, Cipher text, FPGA, Rijndael (Encryption, Decryption).

I. INTRODUCTION

It is necessary to secure information from being accessed by an unauthorized party. Cryptography is a Greek word, meaning "secret writing" this makes records ease and protects the records from attacks. Humans used traditional cryptography communicating secretly. It comprise of encryption and decryption, where encryption is the primary method. A specified algorithm is used to convert the plain text into a secured/cipher text (encrypted message). The reserve processes can be used to decode the cipher text. AES encryption is based on a personal and a public key algorithm, in which a pair of keys are involved where one is meant for encryption and other one for decryption. Public key algorithm forms base for advanced encryption[1].

II. ADVANCED ENCRYPTION STANDARD (AES)/RIJNDAEL

Rjindael [2] is the much larger encryption algorithm and AES is its subset. Rjindael became one of the algorithms to compete with NSIT for becoming widespread algorithm. In 2010, NSIT awarded Rjindael as the algorithm for high security reasons, efficiency, total performance, power consumption and ease.

In AES encryption and decryption is done by a single secret key. In case of asymmetric ciphers, two different keys, the public and private keys are used. The public key which has encrypted the plain text can be decrypted only by a private key. AES is a block cipher. The one by one bit encryption of plaintext bits is done in stream ciphers and there is a variation in transformations throughout an encryption technique.

The functioning of AES algorithm [3] is on the 128, 192 or 256 bits for the process of encryption. The blocks of 128 bits are AES algorithm's I/O. The series of 128, 192 or 256 bits is a input of cipher key input [4]. The number of columns in a cipher key are 4, 6 or 8 which is the length. AES algorithm has three versions of classification.

III. AES ALGORITHM

A two dimensional array of 4x4 bytes is used for performing all operations. It is also known as the state, and any individual byte here is referred to as Sr.c, in which 'r' denotes the row whereas 'c' represent the column. The state is populated with plaintext at the start of an encryption process. A cipher key performs substitutions and permutations. The operations of AES cipher are done either on a state or on entire row/column. The key expansion routine is used to derive round keys from cipher keys for performing an initial round key addition in state. For every round of transformations performed in state, a key expansion routine is used for the generation of a sequence of round keys [5]. Each state array performs different sets of transformations and each one of them depending on a cipher key. Except the last round of AES cipher, all rounds has given transformations as MixColumns, SubBytes (s-box), Shiftrows, AddRoundKey. *BYTE SUBSTITUTION:*A non-linear byte form of substitution table is used for the operation of transformation on every byte of a state. This substitution table comprise of columns/rows.

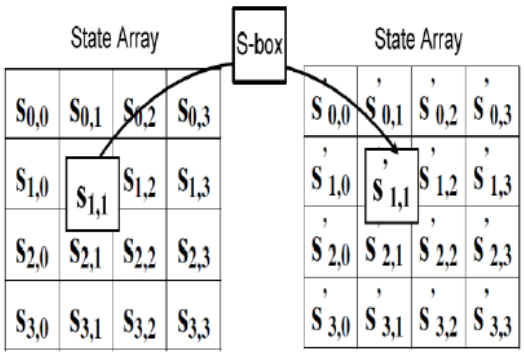


Fig. 1(a)

A. SHIFT ROWS

All the rows except the first one are circularly shifted and the first row with shift row transformation. Circular left shift of one byte is done in the second row. Third row is left shifted by 2 byte round whereas fourth one is rounded shifted by 3 bytes. A right circular shifting will be done for the decryption technique.

Revised Manuscript Received on September 10, 2019.

Mihir Narayan Mohanty, Dept. of Electronics & Communication Engineering, Siksha O Anusandhan Deemed to be University, Odisha, India (E-mail: mihirmohanty@soa.ac.in)

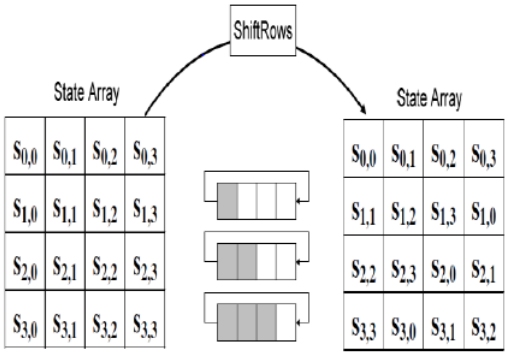


Fig. 1(b)

B. MIX COLUMNS TRANSFORMATION

This transformation depends upon the Galios field multiplication [6]. Substitution of every column's byte with different value is done for permitting in a column, featuring of all 4 bytes.Every column is treated as a polynomial and transformation is operated on the state column. They are referred as polynomials over GF(28).

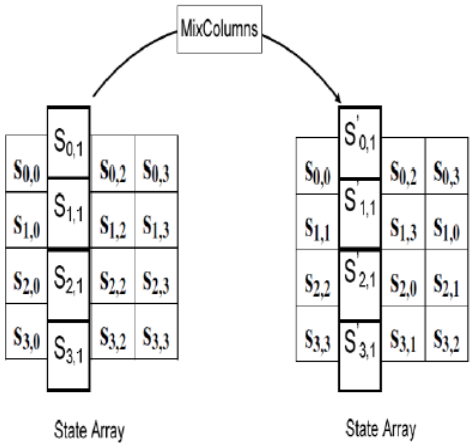


Fig. 1(c)

C. ADDROUNDKEY

It accounts for key enlargement by XORing all four 32 bit words of extended key with 128 bit state units. The most effective operation for AddRoundKey entails key for the purpose of security. A 44 word linear array are produced as an output by taking a four-word (16-byte) as input by the AES key expansion.

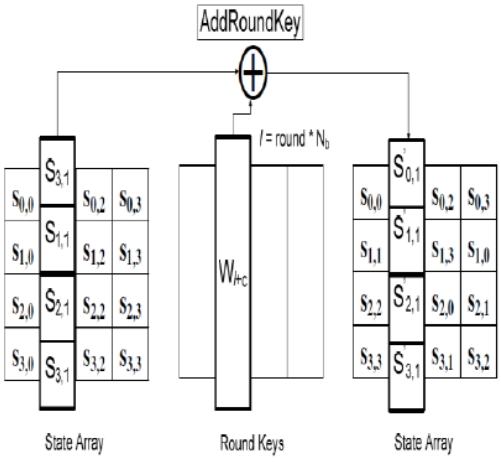
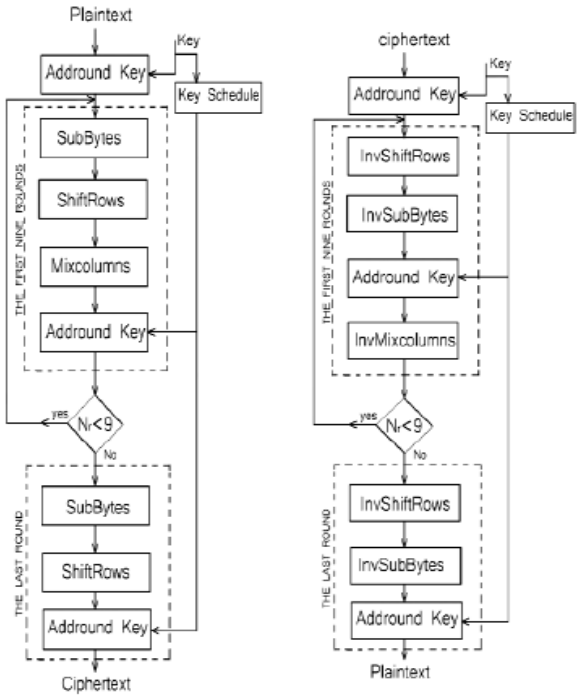


Fig. 1(d)

A block diagram of RIJNDAEL for encryption/decryption is given below:



(A) Encryption **(B) Decryption**

Fig. 2(a) Rijndael encryption/decryption

IV. AES KEY EXPANSION

Each encryption round requires 4 words of round keys by the AES algorithmA linear array of 44 words is produced as an output by taking a four-word key as an input by AES key expansion algorithm. The copying of key in 4 primary words is done. 4 words at a time are filled for the rest of the key. The newly added word w[i] depends upon the previous word w[i-1] and four word positions returned w[i-4]. XOR gate is used in three out of four instances [7]. Figure 2(b) illustrates the way every round uses four of these words. Each subkey is 128 bits long and each word is incorporated with 32 bytes.

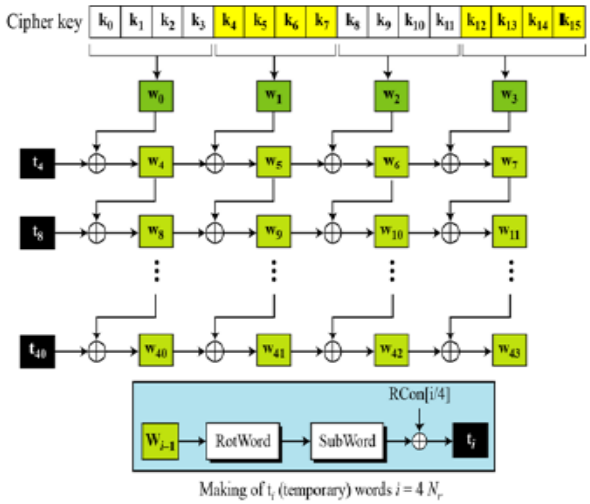


Fig. 2(b) Expansion of AES key

V. METHODOLOGY

The encryption rounds is used in the round key generation for proposed design of pipeline. The AES algorithm's encryption for round 'n' with round 'n+1' key generation. The speed is improved with cost in terms of area of pipelining techniques. A high throughput and high efficiency are achieved by using pipelining methodology. Each encryption iteration requires lower delay and it is the most essential benefit of pipelined design. The iteration is present at the starting of every cycle for round keys for every encryption. Every round of encryption is completed quickly when the delay in every encryption is lower. Thus, overall encryption delay is reduced and the design is controlled at higher clock frequencies. The message encryption rate is increased by the higher clock frequencies and this enhances suitability of design for the time critical encryption applications. Figure 2(d) illustrates a complete pipelined structure comprising of k registers. One round comprises of one pipeline stage and each round is having different transformations[8]. The second transformation is fed as an input by the output of first transformation. It is done through registers and continues till last round is completed.

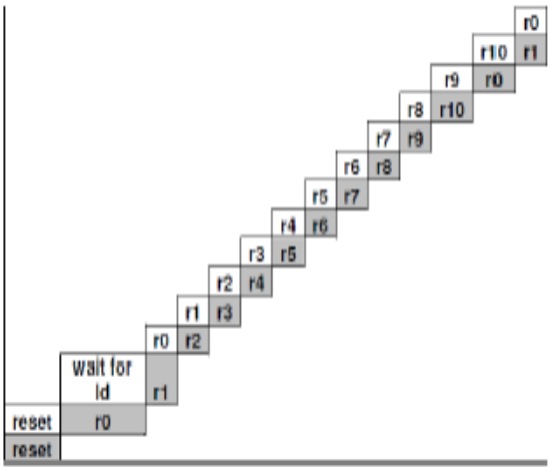


Fig. 2(c) Round key generation and cipher rounds

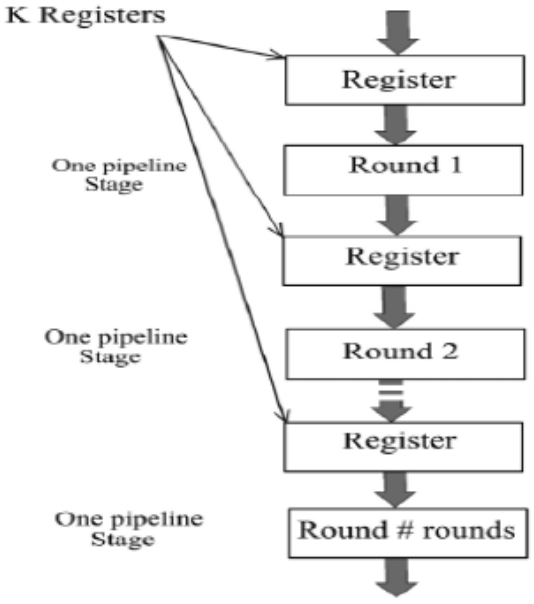


Fig. 2(d) Pipelined Structure

VI. RESULTS AND OUTCOME

The system Verilog hardware description language was used for the designing, modelling and verifying a hardware accelerator for AES 128 encryption algorithm. The delay associated with every round of encryption is reduced by the pipelined design of AES encryption algorithm. It leads to the increment in the throughput of message encryption and validates the hardware model for essential encryption applications. The final secrecy of encryption key is provided by the implementation of AES in the hardware with much faster speed in comparison with software implementation and higher throughput is achieved by inherent hardware concurrency.

VII. . REFERENCES

1. A. U. Rahman, S. U. Miah, and S. Azad, "Advanced encryption standard," in *Practical Cryptography: Algorithms and Implementations Using C++*, 2014.
2. J. Daemen and V. Rijmen, *The Design of Rijndael*. 2002.
3. P. Chodowiec and K. Gaj, "Very Compact FPGA Implementation of the AES Algorithm," 2010.
4. R. B. Lee and Y. Y. Chen, "Processor accelerator for AES," in *Proceedings of the 2010 IEEE 8th Symposium on Application Specific Processors, SASP'10*, 2010.
5. N. S. S. Srinivas and M. Akramuddin, "FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption," in *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, 2016.
6. A. M. Borkar, R. V. Kshirsagar, and M. V. Vyawahare, "FPGA implementation of AES algorithm," in *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, 2011.
7. W. Stallings, *Network Security Essentials : Applications and Standards Fourth Edition*. 2011.
8. L. T. M.Madheswaran, "a Single Chip Design and Implementation of Aes -128 / 192 / 256 Encryption Algorithms," *Int. J. Eng. Sci. Technol.*, 2010.