ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

VLSI Implementation of AES 128 Algorithm for Secure Data Transmission

Ankur Changela

Assistant Professor, Dept. of E.C., IITE, Indus University, Ahmedabad, India

ABSTRACT: Cryptography is the practice and study of techniques for secure communication in the presence of third parties called adversaries. More generally, cryptography is about constructing and analysing protocols that prevent third parties or the public from reading private messages. Major part of cryptography is to design and implement Encryption, a means to convert meaningful messages into total meaningless message. If the message is not encrypted, anyone would be able to read and monitor anyone's messages. Various algorithm has been proposed for encryption and description in the past. In this paper, AES-128 with some modification is presented to increase the security. AES-128 is first implemented on Matlab, and then simulated on HDL platform. Simulation results show the accuracy of the algorithm.

KEYWORDS: Encryption, Decryption, Round Key.

I. INTRODUCTION

Cryptography is derived from two words: Cryptos means hidden (secret) and graphein means writing. Earliercryptography was synonymous with encryption. Means hidden (secret) and graphein means writing. Earliercryptography was synonymous with encryption. Cryptography largely consists of designing and developing ways to convert readable text to complete meaningless (Encryption) and then using a reverse procedure to convert the meaningless text to readable text (Decryption). Nowadays a variety of computational and mathematical methods are used to develop algorithms.[1] A cipher is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a "key". The key is a secret (ideally known only to the communicants), usually a short string of characters, which is needed to decrypt the cipher text. Formally, a "cryptosystem" is the ordered list of elements of finite possible plaintexts, finite possible cipher-texts, finite possible keys, and the encryption and decryption algorithms which correspond to each key. Keys are important both formally and in actual practice, as ciphers without variable keys can be trivially broken with only the knowledge of the cipher used and are therefore useless for most purposes.[2-4]

Encryption is a way to protect the information/data from unauthorized access to prevent the classified information to be read by anyone without authority. In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it.[5] Encryption does not of itself prevent interference, but denies the intelligible content to an intruder. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm, which utilizes a key to encrypt the plain text, in the end generating cipher-text that can only be read if decrypted.[9-10] For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users. [6] Encryption algorithms are broadly classifies into two:

1. Symmetric key algorithms:
2. Asymmetric key algorithms:

Data manipulation in symmetric systems is faster than asymmetric systems as they generally use shorter key lengths. Use of asymmetric systems enhances the security of communication.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

II. PROPOSED AES-128 ALGORITHM

An AES 128 consists of 10 rounds. Each round further consists of a no. of predefined steps[6]. A round of encryption consists of several processing steps that include:

- Substitute Bytes
- Shift Rows
- Mix Columns
- Add round Key

Before the start of any round the original key is added to the input plain text and then in the next 9 rounds the 4 steps as above are repeated. In the 10th round the step of mixing columns is skipped and the loop is exited. The output at the end of Encryption is known as Cipher Text. The reverse happens in each round of decryption. It consist of the following steps:

- Inverse Shift Rows
- Inverse Substitution Bytes
- Inverse Add Round Key
- Inverse Mix Column

Before the start of any round, the last key is originally added. The last round doesn't involve the mix column stage. The output at the end of the Decryption process is known as Plain Text. General Block diagram of AES-128 is shown in figure 1 below.

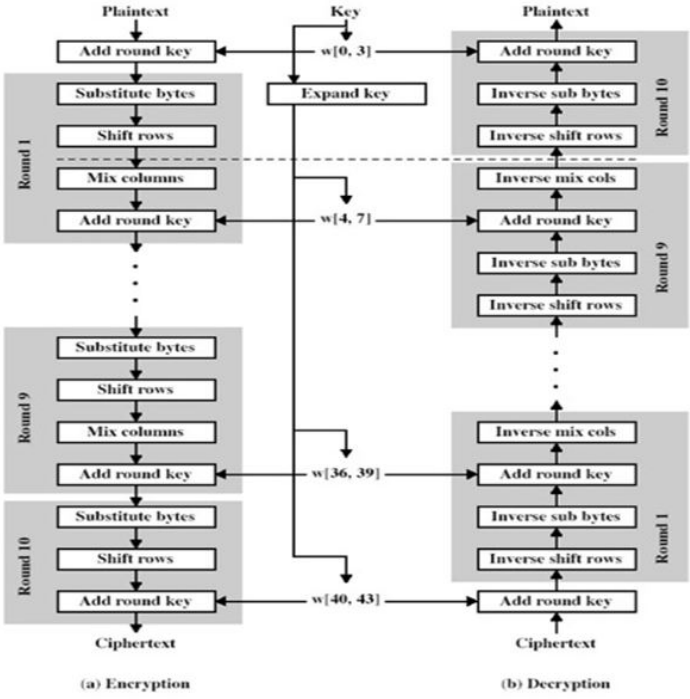

Figure 1: AES block diagram

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

A. *Steps in Encryption:*

The first step in the encryption process is substitution byte. This consists of a lookup table of byte values known as an S-box.[7] The S-box consists of all possible values of an 8 bit sequence (2^8 = 16x16 = 256). Each byte of message, is replacedwith a new byte in the S-box using row column mapping as shown in figure 2 below.

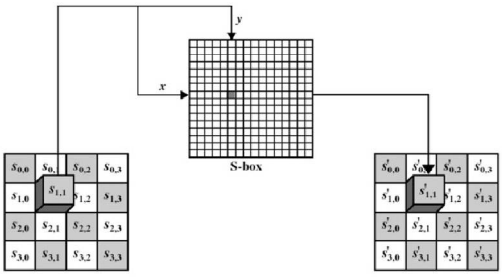

Figure 2:Byte Substitution

The next stage is the shift row transformation. General rule for row shift transformation is that in i^{th} row, bytes will be shifted (i-1) times in circular mode. Each State is treated as an array of four byte columns, i.e. the first column actually represents bytes 1, 2, 3 and 4. A one byte shift is therefore a linear distance of four bytes. The transformation also ensures that the four bytes of one column are spread out to four different columns. The next stage is mix column transformation. In this step, each column is operated on individually. Each byte of a column is mapped into a new value that is a function of all four bytes in the column. The last stage is "Add Round Key". In this stage a round key is added to the generated output after the mix column stage. A new key is generated for every round using the key for the previous round using key expansion algorithm.[10] Thus each round key is originally generated from the main key. In all 11 128 bits round key are needed. Round key is added to each bit using bitwise XOR expansion as shown in figure 3 below.

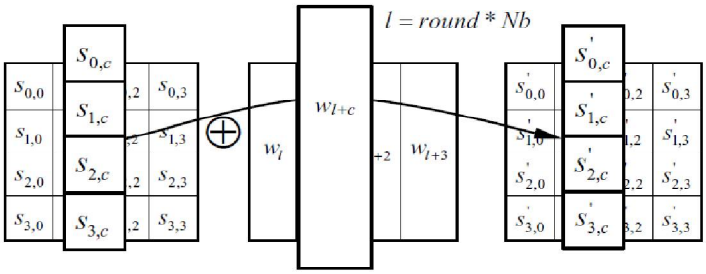

Figure 3: Add round key

B. *Steps in Decryption:*

The first step in decryption is inverse shift row transformation. The inverse shift row transformation works just like the shift row transformation but exactly in the opposite manner. The first row isn't shifted and is left unaltered The second row is shifted by 1 bytes to the right in a circular manner The third row is shifted by 2 bytes to the right in a circular manner The fourth row is shifted by 3 bytes to the right in a circular manner.

Next step is Inverse Substitution Bytes. The inverse substitution transformation is done using an inverse S-box. It works in similar manner as the substitution byte transformation. In this case the original value is obtained by using the substituted value. The leftmost byte represents the row and the rightmost the column.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

The add round key is its own inverse function as XOR is its own inverse. The round keys have to be selected in reverse manner. Hence initially the last key i.e. the 11th key is used. Round key is added to each bit using bitwise XOR expansion as shown in figure 4 below. Last step is Inverse Mix Column Transformation to get an original data.

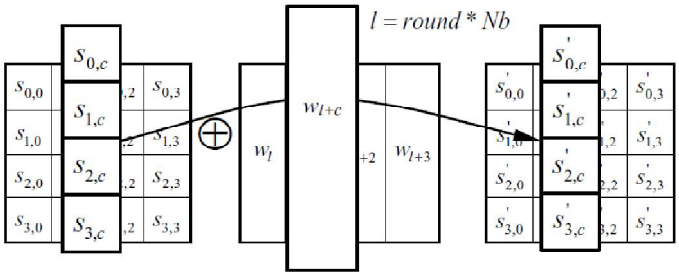

Figure 41: Add round key

III. SIMULATION RESULTS

The entire AES system is first implemented on Matlab and the various NIST test vectors are tested. An image called 'Lena image' is chosen as test vector and image is encrypted using proposed Algorithm. Image is read from the file 'lena_1.bmp'. A key is then used to encrypt the image. The encrypted image is stored as 'e_lena.bmp'. As shown in figure 5, the encrypted image is unrecognizable.

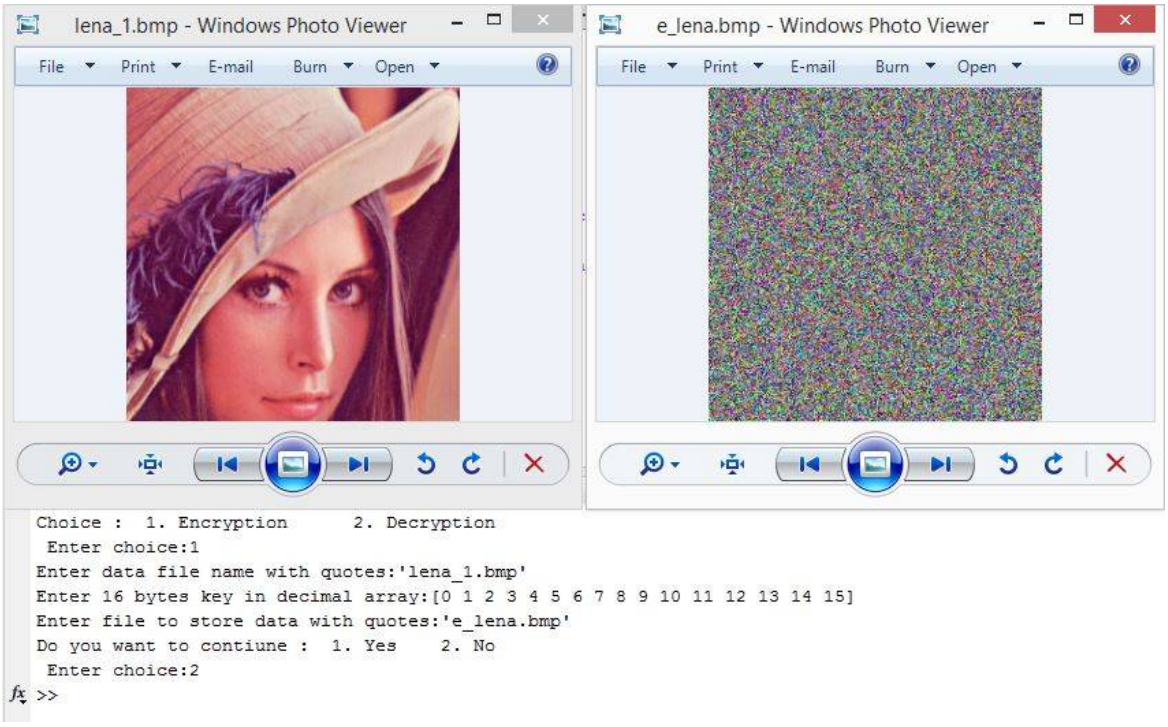

Figure 5: Encryption of Lena image.

The same encrypted image 'e_lena.bmp' is to be decrypted. Image is read from the file 'e_lena.bmp'. The same key is then used to decrypt the image. The decrypted image is stored as 'd_lena.bmp'. As shown in figure 6, the decrypted image is same as the original Lena image.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

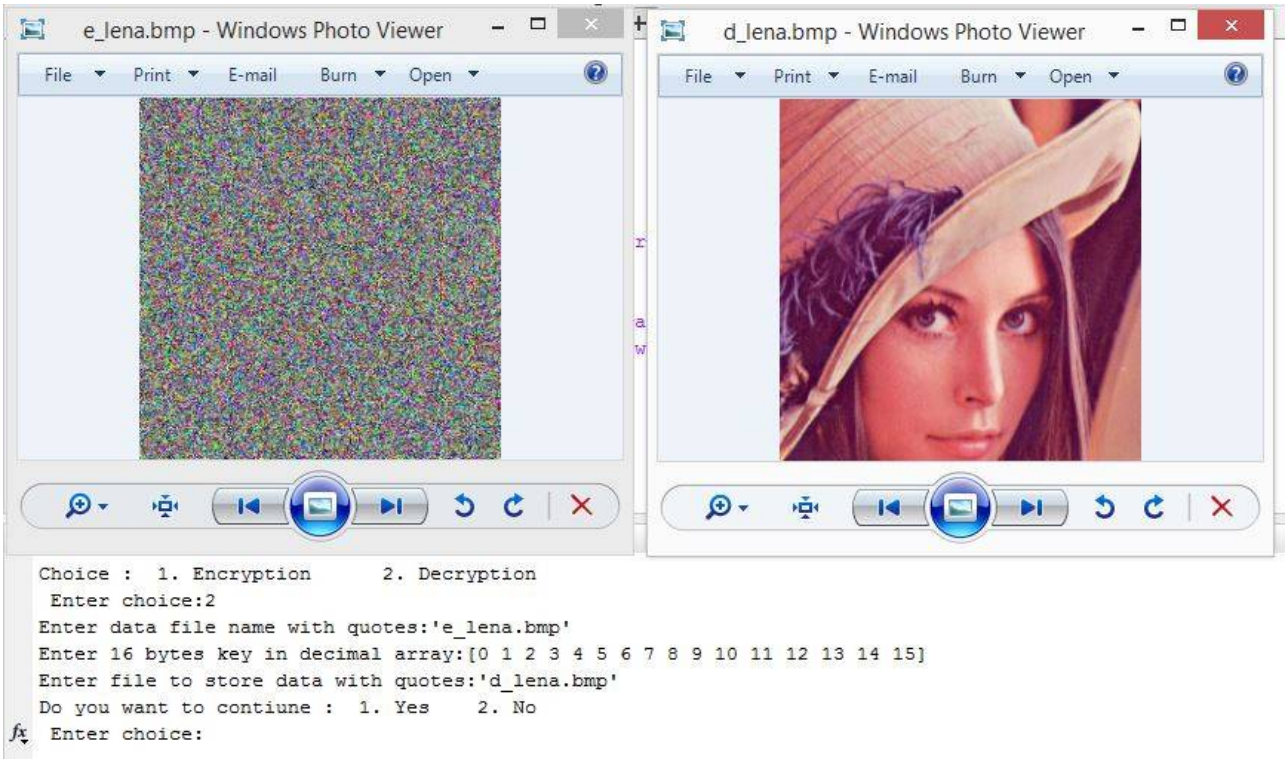

Figure 6: Decryption of Lena Image

After successful implementation on Matlab, HDL coding in Verilog is done. Various modules are created which perform the desired tasks. The figure7 shows encryption of one entire BRAM data, i.e. it encrypts entire 256 bytes of data. The sof signal signifies the start of frame of 256 bytes. The eof signal signifies the end of 256 bytes and data after this is to be discarded. A sofout signal is sent with the first encrypted output byte and an eofout signal with the last encrypted byte. Each byte is accompanied with a clockout pulse. Data is sent out 16 bytes at a time, i.e 128 bits at a time. The 'data for encryption' bus signifies the data which is sent for decryption. Upon reset the entire cycle is repeated.

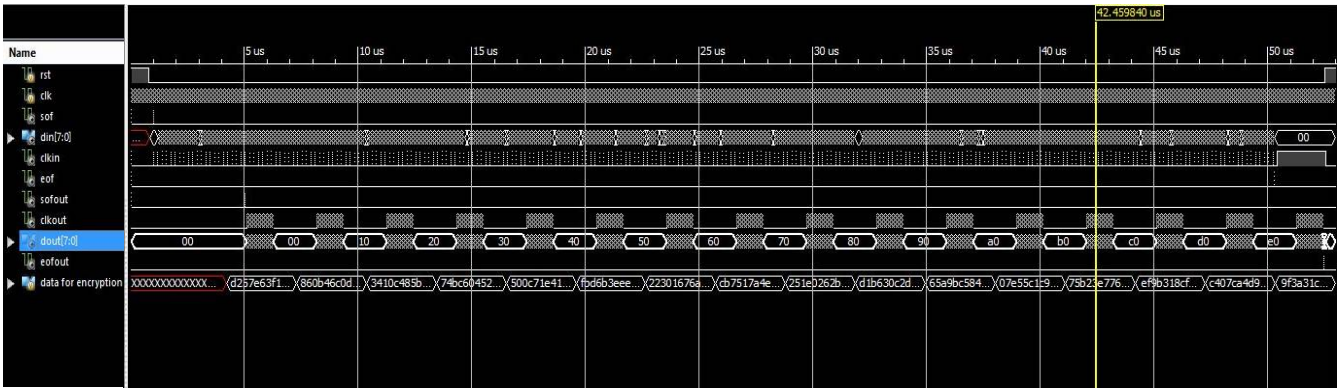

Figure 7: Encryption of full 256 bytes

The figure 8 shows decryption of one entire BRAM data, i.e. it decrypts entire 256 bytes of data. The sof signal signifies the start of frame of 256 bytes. The eof signal signifies the end of 256 bytes and data after this is to be discarded. A sofout signal is sent with the first decrypted output byte and an eofout signal with the last decrypted byte.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Each byte is accompanied with a clockout pulse. Data is sent out 16 bytes at a time, i.e 128 bits at a time. The 'data for decryption' bus signifies the data which is sent for decryption. Upon reset the entire cycle is repeated.

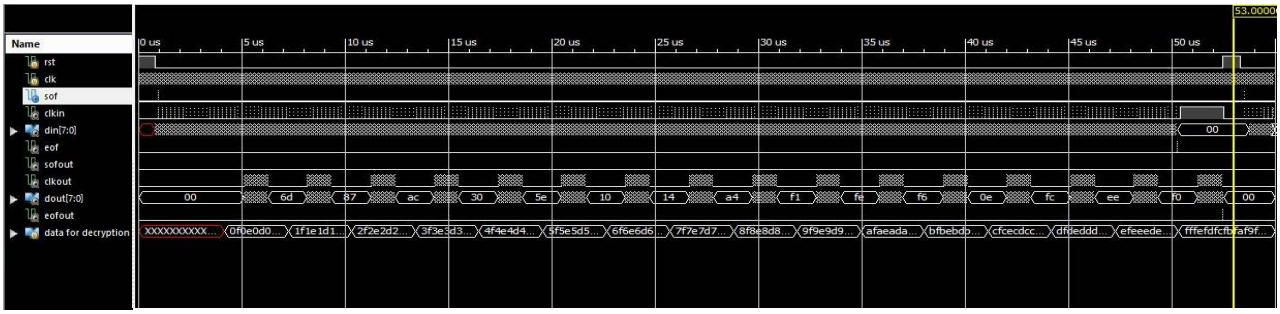

Figure 2: Decryption of full 256 bytes

IV. CONCLUSION AND FUTURE WORK

The aim of the project is to implementthe real time AES encryption and decryption using Verilog HDL so that it can be used for FPGA or ASIC implementation. Before HDL implementation, the complete design is simulated and verified on MATLAB. Various modules are created and then the entire design is optimized to utilize minimal resources. Data which is to be encrypted is stored in a BRAM. The encrypted data is again stored in another BRAM which is used as data for decryption module. The result of both encryption and decryption is verified and AES-128 is successfully implemented on HDL.

REFERENCES

[1]. Manjesh K N and R K Karunavathi ," Secured High throughput implementation of AES Algorithm ", International Journal of Advanced Research in Computer Science and Software Engineering, vol.5 pp. 1193- 1198, May 2013 .
[2]. Hoang Trang and Nguyen Van Loi, "An efficient FPGA implementation of the Advanced Encryption Standard (AES) algorithm", IEEE Transactions, vol. 20 pp. 978-1-4673-0309-5, 2012.
[3]. AJ Elbirt, W Yip, B Chetwynd and C Paar, "An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists", IEEE Transactions on VLSI, 2010.
[4]. G.P. Saggese, A. Mazzeo, N. Mazzocca and A.G.M. Strollo. An FPGA-Based Performance Analysis of the Unrolling, Tiling, and Pipelining of the AES Algorithm, *in FPL 2003*, LNCS 2778, pp. 292- 302, 2003.
[5]. KimmoJarvinen, MattiTommiska and JormaSkytta, "Comparative Survey of High Performance Cryptographic Algorithm Implementations on FPGAs", IEEE Proceedings - Information Security, vol. 152, no. I, Oct. 2005, pp. 3-12.
[6]. Tim Good and MohannedBenaissa "Very Small FPGA Application-Specific Instruction Processor for AES," IEEE transactions on circuits and systems, vol. 53, issue. 7, pp. 1477-1486, July 2006.
[7]. Xinmiao Zhang and Keshab K. Parhi "High-Speed VLSI Architectures for the AES Algorithm," IEEE transactions on very large scale integration (vlsi) systems, vol. 12, ISSUE. 09, pp. 957-967, September 2004.
[8]. Adam J. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," IEEE transactions on very large scale integration (vlsi) systems, vol. 9, issue. 4, pp. 545-557, August 2001.
[9]. P. S. Abhijith, M. Srivastava, A. Mishra, M. Goswami and B. R. Singh, "High Performance Hardware Implementation of AES using Minimal Resources," IEEE International Conference on Intelligent Systems and Signal Processing, Gujarat, pp. 338-343, March 2013.
[10]. Trang Hoang and Van Loi Nguyen, "An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm," IEEE International Conference on Computing and Communication Technologies, Research, Innovation and Vision for the Future, Ho Chi Minh City, pp. 1-4, February-March 2012.