Implementing Low Power and Efficient Image Encryption System Based On 128-Bit AES Algorithm On FPGA

[1]Shubhendu Kumar Mandloi, [2]Mukti Awad
[1]M-Tech student
[1]Department of Electronics & Communication, Acropolis,
Indore, India

***Abstract* - In this paper, the proposed 128bits AES Scheme provides high secure and less area and presents optimized mix column architecture to get less area and delay than the presented mix column. In select a highly secured symmetric key encryption algorithm Advanced Encryption Standard, in order to increase the speed and throughput using pipeline technique in four stages, control unit based on logic gates, in optimal design of multiplier blocks in mixcolumn phase and simultaneous production keys and rounds. In the existing architecture, we are used complex mix-column in masked AES architecture. It consists of 8 multipliers and 11 adders. It consumes more area and power. In the proposed architecture, a novel mix column is introduced to reduce the area and power than the existing method. The proposed mix-column consists of only 12 adders and 4 X time circuit. Instead of normal multiple in the Mix Columns, we are using X time unit to reduce the circuit complexity. The 128 bits AES Scheme is used for ATM Net banking, Satellite Communication, Military applications.**

***Index Terms* – Encryption standard (AES), pipelining, signal gating, image encryption, decryption, etc.**

I. Introduction

Today, the data transmission on the wireless networks suffers from severe changes. Therefore, encryption can present solutions to these problems through minimizing the risk of access to information and databases. Cryptography is a necessary tool that can be used protection of privacy, intellectual property, public security and social, business and financial information as well as the practice of e-commerce transactions and anonymous payments.

Advanced Encryption Standard is a cryptography algorithm proved to have the best quality among 15 candidates by National Institute of Standards and Technology. The advanced Encryption Standard has high security with relatively little memory and CPU resource requirements. It is easier to apply cryptographic on computer based communication scheme than on conventional systems like telephone, etc. It is not feasible to dedicate a general computer for each of such systems.

Advanced Encryption Standard is a symmetric block cipher. The same key is used to encrypt and decrypt the data. The plain text and the cipher text are the same size. AES is an algorithm for performing encryption which is a series of well-defined steps that can be followed as a procedure. For original information is known as plaintext, and the encrypted form as cipher text. The cipher text message contains all the information of the plaintext message, although is not in a format readable by a human or computer without the proper mechanism to decrypt it. The encrypting procedure is varied depending on the key which changes the detailed operation of the algorithm. Without the key, the cipher cannot be used to encrypt or decrypt.

II. CRYPTOGRAPHY

A system for encoding and decoding a secret message is called as cryptography. More generally, people think of cryptography as the art of mangling information into apparent unintelligibility in a manner allowing a secret method of untangling. The basic service provided by cryptography is the ability to send information among participants in a way that prevents others from reading it. This kind of cryptography can provide other services.

- Integrity checking – the reassuring the recipient of a message that the message has not been altered since it was generated through a legitimate source.
- Authentication - Verifying someone’s (or something’s) identity.

A message in its original form is known as plaintext or clear text scheme. The mangled information is known as cipher text. Process for producing cipher text from plaintext is known as encryption scheme. The reverse of encryption is called decryption. There are two types of encryption technique.

Symmetric cryptosystems

It is also known as secret key cryptography. Sender & receiver use same keys for encryption & decryption namely PUBLIC or PRIVATE respectively. The Secret key cryptography involve the use of a single key process. Given a message called plaintext and the key, encryption produces unintelligible data information which is about the similar length as the plaintext. Decryption is the reverse of encryption method; it is the conversion of cipher to plain text and uses the same key as encryption. The symmetric algorithms can be divided into stream ciphers and block ciphers. The stream ciphers encrypt a single bit of plaintext at a time, whereas block ciphers take a number of bits and encrypt them as a single unit. Here the same are used for the encryption and decryption processes so it is called as the secret key cryptography.

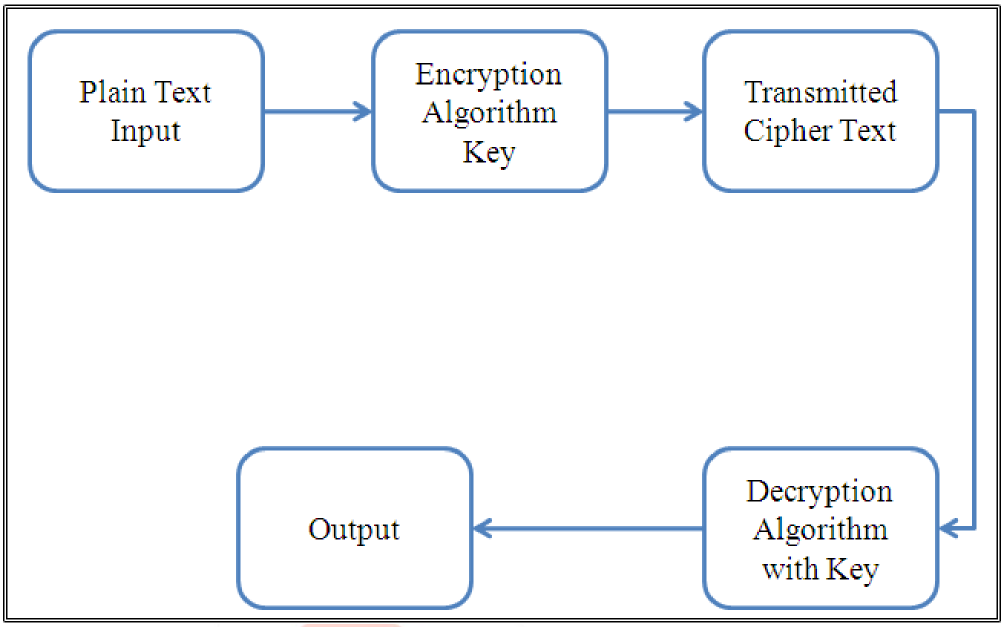


Fig. 1 Block diagram of symmetric cryptosystem

III. ADVANCED ENCRYPTION ALGORITHM

The cipher consists of N rounds, where the number of rounds depends on the key length: it is 10 rounds for a 16-byte key; and 12 rounds for 24-byte key; and 14 rounds for a 32-byte key. The first N – 1 rounds consist of four distinct transformation functions: Sub Bytes, Shift Rows, Mix Columns, and Add Round Key, which are described subsequently. For final round contains only 3 transformation, and there is a initial single transformation before the first round, which can be considered Round 0. Each transformation takes one or more 4 x 4 matrices as input and produces a 4 x 4 matrix as output. Shows in figure .2 that the output of each round is a 4 x 4 matrix, with the output of the final round being the cipher text. Also, the key expansion function generates N + 1 round keys, each of which is a distinct 4 x 4 matrix. Every round key serve as one of the inputs to the Add Round Key transformation in each round.

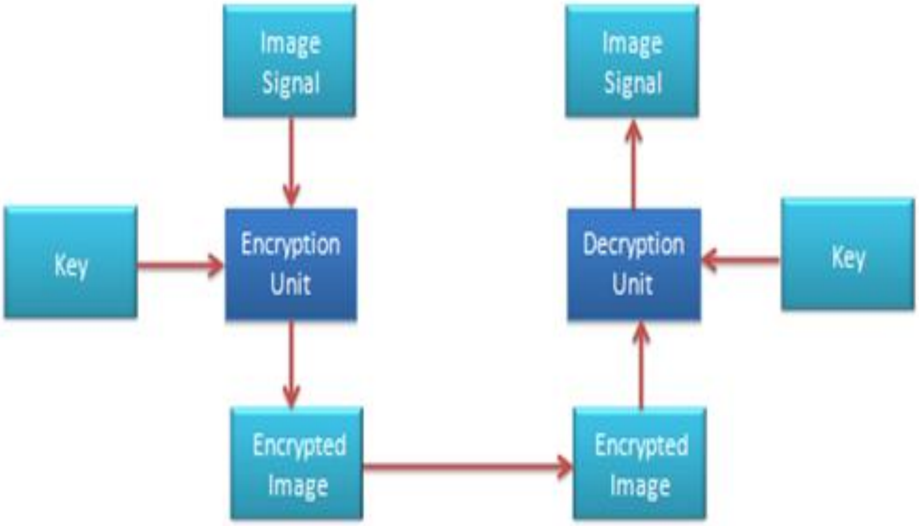


Fig. 2 Block diagram of entire system

IV. FUNDAMENTALS OF AES

The AES has four steps through which it accomplishes data security while transferring any data from sender to receiver. The steps are illustrated as follows

Sub Bytes Transformation

Sub Bytes transformation is a non-linear by substitution that operates independently on each byte of the State using a substitution table (S-box). In this S-box which is constructed by composing two transformations as:

- Take the multiplicative inverse in the finite field GF (2^8) and the element {00} is mapped to itself.

- The following affine transformation the bit, bi is the bit of the byte, and ci is the ith bit of a byte c with the value {63} or {01100011}. A prime on indicates that the variable is to be updated with the value on the right.

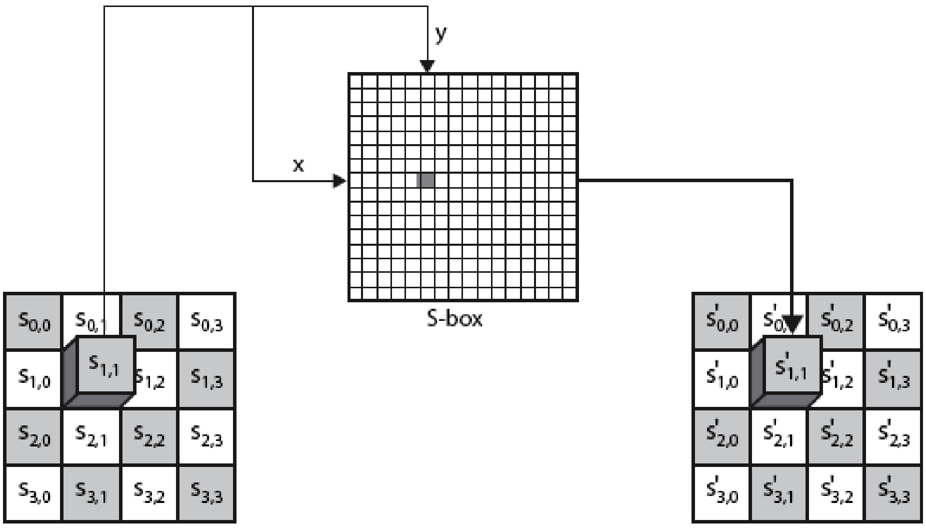
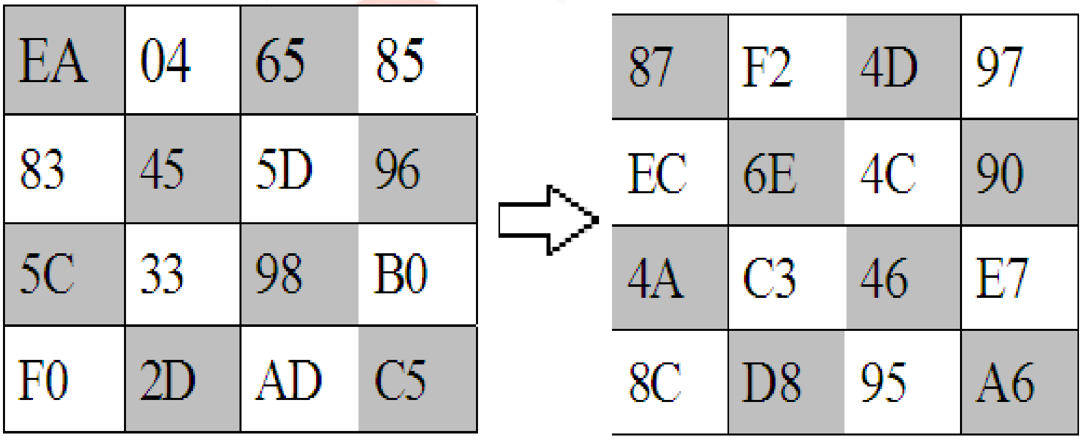

y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig. 3 Byte Substitution operation on each byte of state independently



EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

Fig. 4 Example of the sub bytes transformation

The Substitute bytes stage uses an S-box to perform a byte-by-byte substitution of the block. Here is a single 8-bit wide S-box used on every byte. In this S-box is a permutation of all 256 8-bit values, in constructed using a transformation which treats the values as polynomials in GF(2^8).

Shift Row Transformation

In the Shift Rows transformation technique, the bytes in the last three rows of the state are cyclically shifted over different numbers of bytes offsets. The first row, r = 0, is not shifted. the shift value (r, Nb) depends on the row, r, as follows (Nb = 4): shift(1,4); shift(2,4);shift(3,4); shift(5.4) This has the effect of moving bytes to lower” positions in the row, while the “lowest” bytes wrap around into the top” of the row.

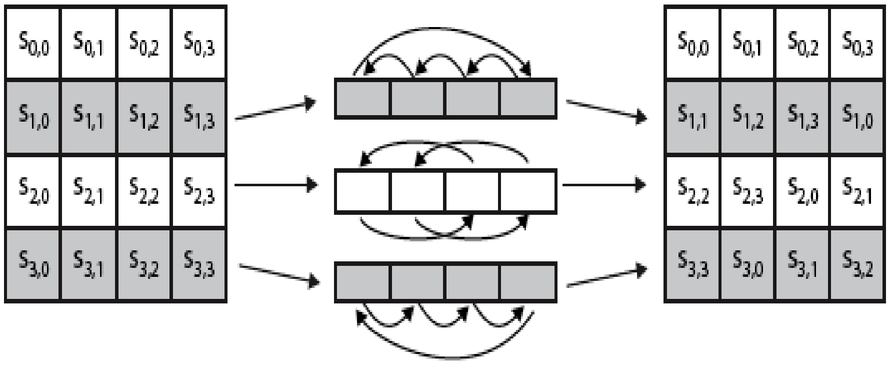
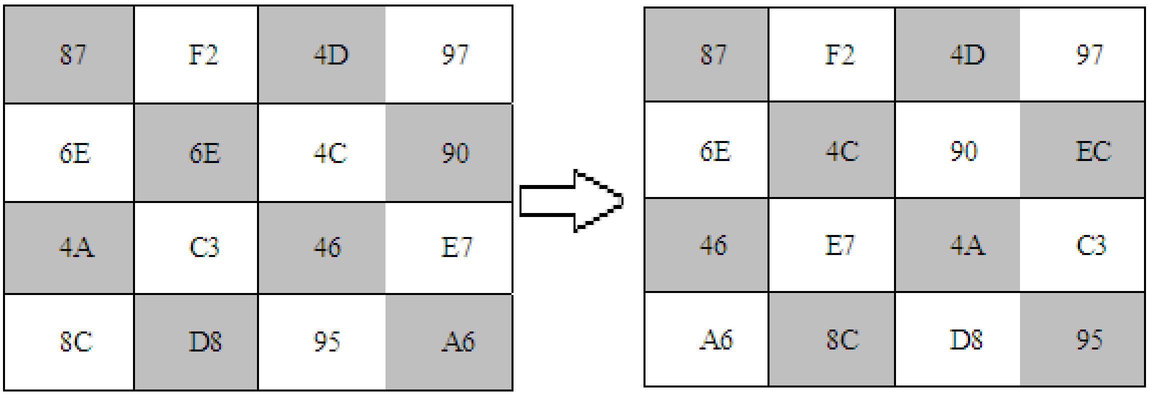

$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$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

Fig. 5 Shift row transformation operation



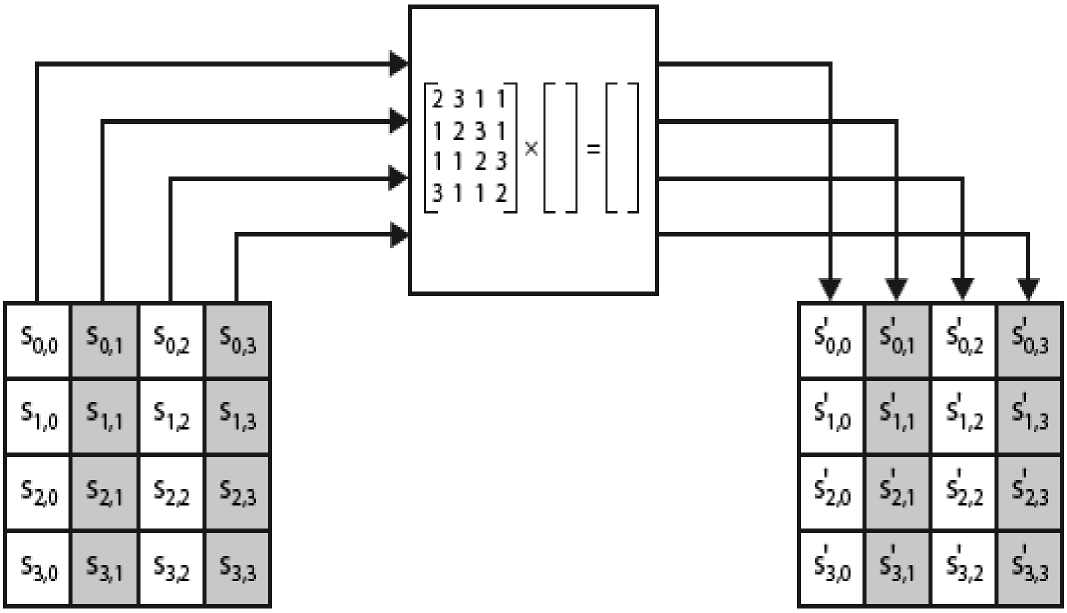
87	F2	4D	97
6E	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Fig. 6 Example of the shift rows permutation.

Mix Column Transformation

The Mix Columns transformation operates on the state column-by-column, in treating each column as a four term. The columns are considered as polynomials over GF (28) and multiplied modulo x4 + 1 with a fixed polynomial a(x), given by a(x)={03}x3 +{01}x2 + {01}x + {02}

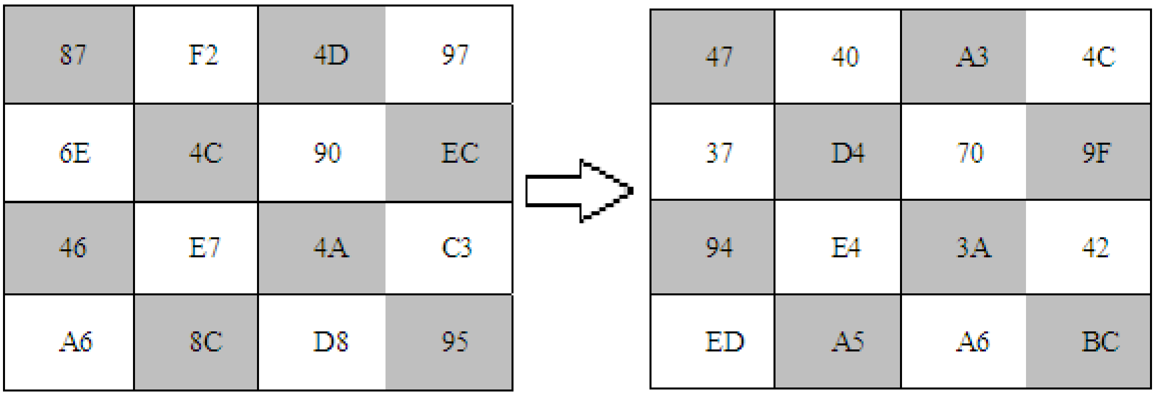


$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

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

$s'_{0,0}$	$s'_{0,1}$	$s'_{0,2}$	$s'_{0,3}$
$s'_{1,0}$	$s'_{1,1}$	$s'_{1,2}$	$s'_{1,3}$
$s'_{2,0}$	$s'_{2,1}$	$s'_{2,2}$	$s'_{2,3}$
$s'_{3,0}$	$s'_{3,1}$	$s'_{3,2}$	$s'_{3,3}$

Fig. 7 Mix Columns transformation operation



87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Fig. 8 Example of the mix column transformation

$(\{02\} \bullet \{87\}) \oplus (\{03\} \bullet \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$
$\{87\} \oplus (\{02\} \bullet \{6E\}) \oplus \{03\} \bullet \{46\} \oplus \{A6\} = \{37\}$
$\{87\} \oplus \{6E\} \oplus \{02\} \bullet \{46\} \oplus (\{03\} \bullet \{A6\}) = \{94\}$
$(\{03\} \bullet \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \bullet \{A6\}) = \{ED\}$

In practise, you implement Mix Columns through expressing the transformation on each column as 4 equations to compute the new bytes for that column. This computation only involves shifts value, XORs and conditional XORs. The decryption computation requires the use of the inverse of the matrix, which has larger coefficients, and is thus potentially a little harder & slower to implement. The designers & the AES standard provide an alternate characterization of Mix Columns, which treats each column of State to be a four-term polynomial with coefficients in GF(2^8).

Add Round Key

In the Add Round Key transformation, a Round Key is added to the State through a simple bitwise XOR operation. Each Round Key consists of Nb words from the key schedule those Nb words are each added into the columns of the State, such that In the Cipher, the initial Round Key addition occurs when round = 0, prior to the first application of the round function. The application of the Add Round Key transformation to the Nr rounds of the Cipher occurs.

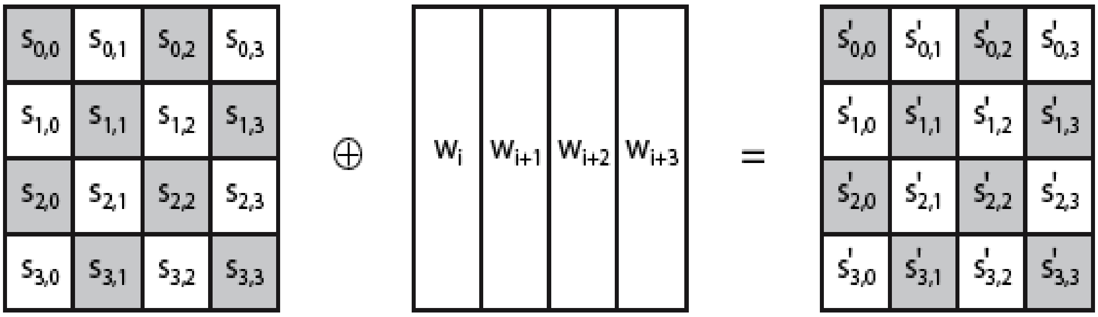
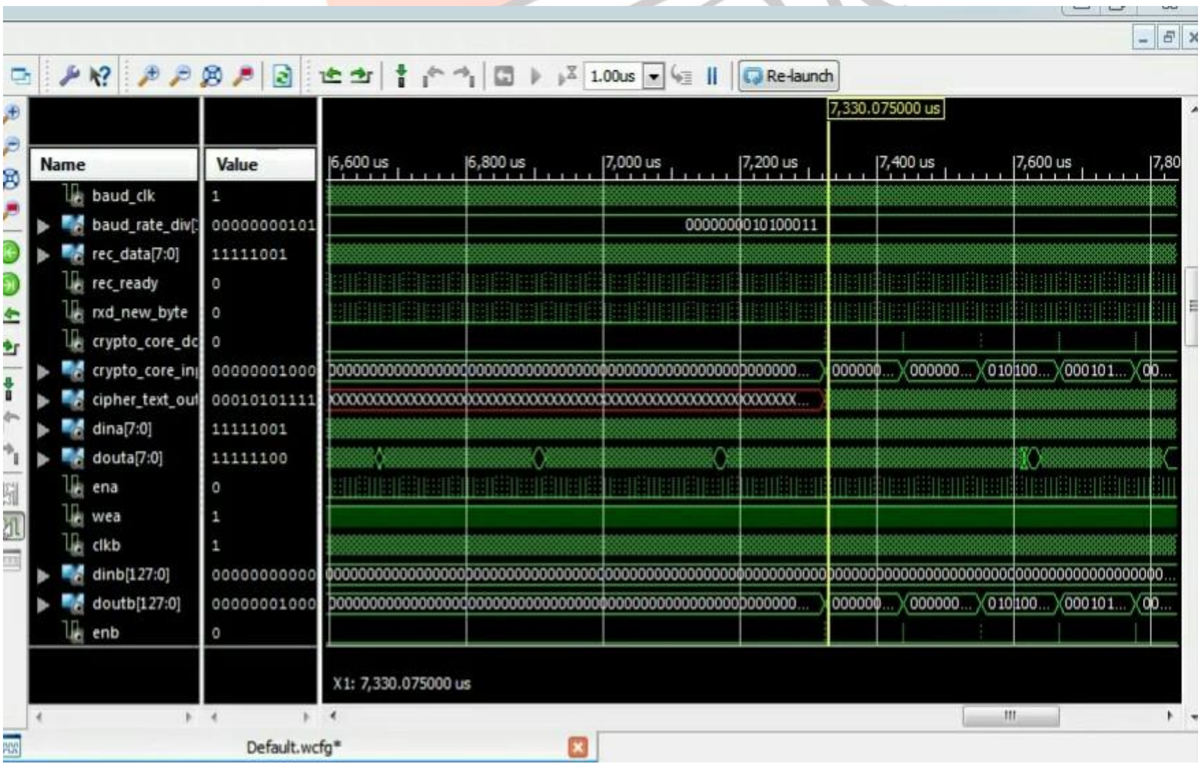

$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$\oplus$
W_i W_{i+1} W_{i+2} W_{i+3}
=
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig. 9 Illustrates the Add Round key stage, which like Byte Substitution ,operates on each byte of state independently.

V. RESULT

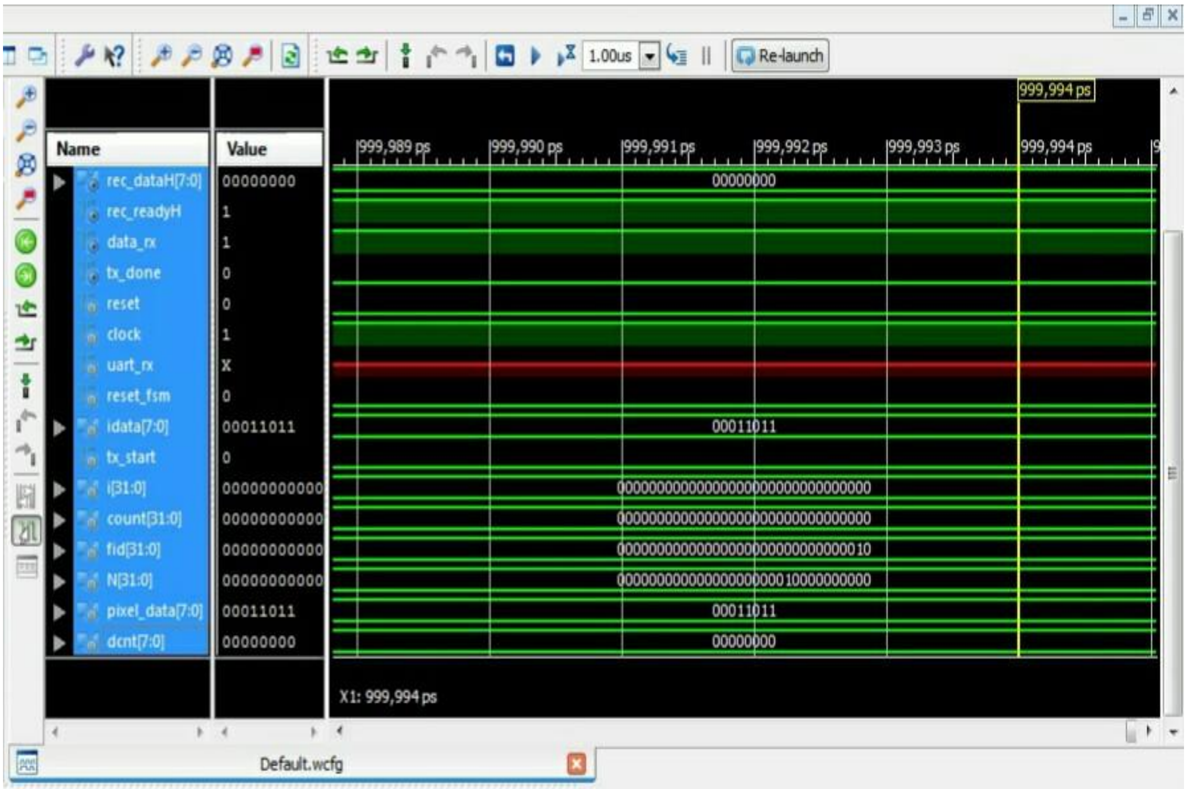
For all practical purposes, the simulation is carried out on the ISIM simulator and use of ModelSim occasionally. The test bench is written for the top most modules which ultimately sits on the FPGA and rest of the modules are tested either by forcing the values in simulation mode or through a local test bench written specifically for that module. In this report simulation results of various important modules are discussed.



Name	Value
baud_clk	1
baud_rate_div[	00000000101
rec_data[7:0]	11111001
rec_ready	0
rxd_new_byte	0
crypto_core_dc	0
crypto_core_in	00000001000
cipher_text_out	00010101111
dina[7:0]	11111001
douta[7:0]	11111100
ena	0
wea	1
clkb	1
dinb[127:0]	00000000000
doutb[127:0]	00000001000
enb	0

Fig. 10 Simulation result for encryption

The above figure shows the simulation result for encryption process in which it shows the received data with its value and the test bench waveforms.



Name	Value
rec_dataH[7:0]	00000000
rec_readyH	1
data_rx	1
tx_done	0
reset	0
clock	1
uart_rx	X
reset_fsm	0
idata[7:0]	00011011
tx_start	0
i[31:0]	00000000000
count[31:0]	00000000000
fid[31:0]	00000000000
N[31:0]	00000000000
pixel_data[7:0]	00011011
dcnt[7:0]	00000000

X1: 999,994 ps

Fig. 11 Simulation result for decryption

The above figure shows the simulation result for decryption process in which it shows the received data with its value and the test bench waveforms.

Conclusion

In this paper implementation of Advanced Encryption Standard algorithm is used for data encryption that can process with the data block of 128 bit and cipher key length of 128 bit. Implemented the encryption and decryption units on Xilinx Spartan-3A(XC3S700A-FG484-5C). Here the test case is to encrypt the image, it means the converting the plain text into the cipher text which can't find out or unintelligible for the hackers. While encrypting the data the secret key is used to secure the data. The AES algorithm provide a enhanced security and less open to attacks in which it supports the larger key so finally it provides high security compared to DES and Triple DES. While computing the existing AES, the Sub Bytes transformation consumes the more memory in AES so to overcome this affine transform is used in AES flow.

References

[1] G. H. Karimian, B. Rashidi, and A.farmani , "A High Speed and Low Power Image Encryption with 128-Bit AES Algorithm ",International Journal of Computer and Electrical Engineering, Vol. 4, No. 3, June 2012.

[2] Sumalatha Patil , "Design of High Speed 128 bit AES Algorithm for Data Encryption", International Journal of Current Engineering and Technology , 2013.

[3] Sammoud Ali, "Performances analysis of image encryption for medical applications", Journal of Information Sciences and Computing Technologies, Volume 1, Issue 1, January, 2015.

[4] N. Sloss, D. Symes, and C. Wright, "ARM System Developer's Guide, Designing and Optimizing System Software," Morgan Kaufmann, 2004.

[5] B. Gladman, "A specification for Rijndael, the AES Algorithm," Available at http://fp.gladman.plus.com, May 2002.

[6] XYSSL Crypto Library, GNU Lesser General Public License, 2003.

[7] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki, "A Modified AES Based Algorithm for Image Encryption," World Academy of Science, Engineering and Technology, 2007.

[8] Y. Cheng, C. C. C. Hsieh, C. W. Huang, and C. J. Chang, and K. H. Chang, "Embedded a Low Area 32-bit AES for Image Encryption/Decryption Application," Circuits and Systems, ISCAS 2009. IEEE International Symposium on, pp. 1922–1925, 2009.