IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

FPGA Implementation of 256 Bit Key AES Algorithm Optimization of Sub Bytes Block and Key Schedule using verilog HDL

Sumithra Poonaykar G.C[1], Shylaja.V[2]

Department of M. Tech, Bangalore Institute of Technology, Bangalore, India [1]

Department of M. Tech, Assistant Professor, Bangalore Institute of Technology, Bangalore, India [2]

Abstract: Hardware security gets involved in various operations and they are in the field of e-commerce, banking, communications, satellite, image processing and so on. Cryptography is nothing but it is the procedure of transforming the plain input text into the cipher output or vice versa. Three forms of Cryptography: Private Key Cryptography, Public Key Cryptography & Hash functions. Private Key is nothing but it makes use of a similar key for Encryption & also for Decryption process whereas Public Key is nothing but it uses the two different keys for encryption and also for decryption process. Since AES operates using similar key for Encryption as well as Decryption so this type performance are important hastily, easy to apply and it requires truly lower processing power. Encryption process is only thing left to guard the specific information or data communication. Depending on the Key length it's more effective and there are three Key lengths options available and they are 128bit, 192bit and 256bit pivotal length. Greater the Key length more time is demanded to break the system or to hack the system. AES performs four different functions or transformations and they are as follows: Sub-bytes, Shift-rows, and Mix-column & Add round Key. By using Pipelined Architecture and LUT greater speed can be achieved. The proposed architecture is formed on optimizing timing and this is done by using verilog HDL.

Keywords: AES (Advanced Encryption Standard), FPGA (Field Programmable Gate Array), LUT (Look Up Table), Mix (Mix-column) Shift (Shift-rows), Sub (Sub-bytes).

I. INTRODUCTION

Internet communication plays a valuable thing in transferring the huge amount of information and data in several ways. Some of the information can be transferred from the unprotected pathway to the receiver from the sender. There are different ways to protect the data from the middle person. By using the concept of cryptography we can protect the secret information from the other person. We can protect the information from the other person by the process like Encryption and Decryption. Encryption is converting data such that it should not be known to the other person. Original information or the data will already be known to the Transmitter and the Receiver itself. This can achieve in the AES where it follows the block cipher method.

The block cipher will perform its function only for the particular unchanged key length groups or the blocks. There are three types of key lengths available and they are 128bit, 192bit, and 256bit key length. According to the block cipher method the plain text size and the cipher key length should be same. The four transformations are used in both the encryption as well as decryption process. If we are performing encryption in AES we have to use same length of plain text and the cipher key. For the encryption it performs certain round of operations which depends upon Key. Fig.1 shows Architecture of 256bit Advance Encryption Standard. This architecture is based on the symmetric key cryptography that means using similar Key for the two process i.e. Encryption & Decryption. For Encryption procedure the input plain text is changed to output cipher text where as for Decryption procedure it's vice versa. For encryption in AES the input taken will be the 128bit plain text along with it a cipher key of about 128bit is also used. Firstly it performs add round key operation. In this XOR operation takes place.

The corresponding output is sent to first round. In first round it performs four transformations they are: Sub-bytes, Shift-rows, and Mix-column & Add-round Key. These transformations are performed for 10 rounds since the plain text size and key size is 128bit. In the case of last round only three transformations will be performed. Here mix column will not be performed and output of the shift-rows will be given to Add-round Key. After completion of Add Round Keys corresponding output which we get will be the 128bit cipher text. Similar operation takes place in case of 192bit and 256bit plain or the cipher text along with same length of key.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

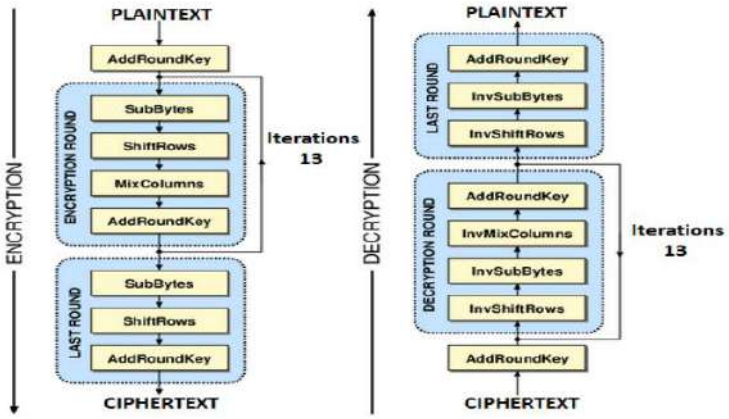


Fig. 1 Architecture of 256 AES Algorithm

AES is the replacement of DES. AES algorithm in each round contains 4 Transformation:-

1: Sub-Bytes Transformation: AES defines 16x16 matrices of byte variables known as S-Box which contains the permutation of the possible 256 i.e. 8bit values. Each & every individual byte of the state matrices is mapped to the new byte as shown in the following way: The left side 4bits of byte which we use as a row number and right side 4bits are used as a column number. The rows & columns digits are treated as indexes from the S-box to be selected an individual 8bit output number.

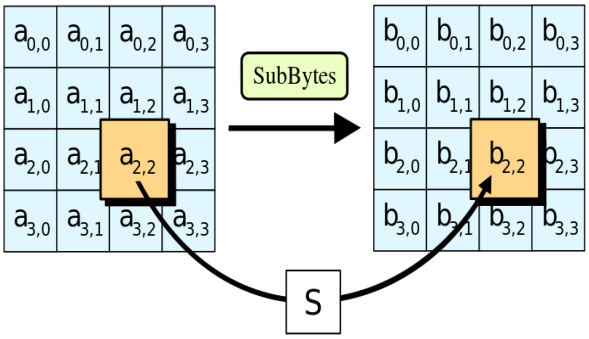


Fig. 2 Sub bytes Transformation

2: Shift-Row Transformation: While performing shift row Transformation shifting is done towards the left. Number of shifts depends on Number of rows of the output/state matrix. Important Rules to be followed while shifting-rows are: Initially while doing shifting row 1 = No need to do shifting leave it constant. Next while shifting row 2 = Only 1 byte shifting is
done i.e. towards left shift. Further while shifting row 3 = Here 2 byte shifting is done towards left shift & finally while shifting row 4 = Here 3 byte shifting is done towards left shift.

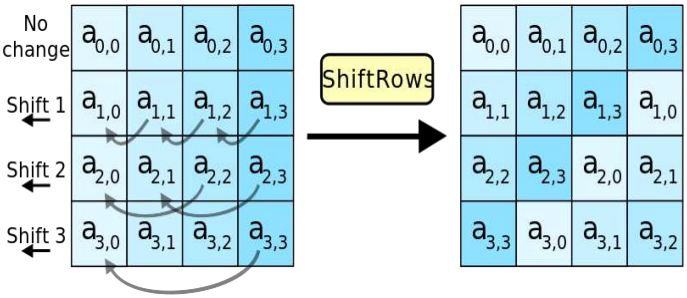


Fig. 3 Shift rows Transformation

3: Mix Column Transformation:

Mix-Columns Transformations called mix columns perform operations on each column separately. Individual bytes of column are again mapped to a new variables & that is task of four bytes in the columns. In last round mix-columns

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

transformation will not be performed since it is omitted.
Example

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

This can also be seen as the following:

$$\begin{aligned} b_0 &= 2a_0 + 3a_1 + 1a_2 + 1a_3 \\ b_1 &= 1a_0 + 2a_1 + 3a_2 + 1a_3 \\ b_2 &= 1a_0 + 1a_1 + 2a_2 + 3a_3 \\ b_3 &= 3a_0 + 1a_1 + 1a_2 + 2a_3 \end{aligned}$$

Where "+" indicate xor operation.

Example:

State	MixColumn Matrix	Mixed
d4 e0 b8 1e	02 03 01 01	04 e0 48 28
bf b4 41 27	01 02 03 01	66 cb f8 06
5d 52 11 98	01 01 02 03	81 19 d3 26
30 ae f1 e5	03 01 01 02	e5 9a 7a 4c

Fig. 4 Mix column Transformation

4: Add round Transformation: In the Add-Round keys transformation known as Add-Round Key. 128bits state matrices variables are bitwise EX-OR operation is performed using 128bits round Key. Add-round Key proceeds one column at time.

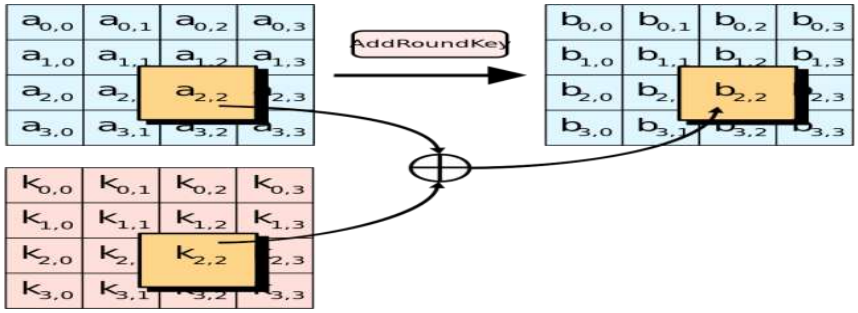


Fig. 4 Add round Key Transformation

II. OBJECTIVE

1. To design a verilog HDL code for various modules of AES Encryption.
2. To integrate all the modules in order to generate a code for AES Encryption.
3. Similarly designing a verilog HDL code for Decryption.
4. To design the verilog HDL code for pipelined AES to enhance speed.
5. To Design the verilog code for LUT based Mix columns Crypto multiplication.
6. To carry out thorough Functional and timing simulation of the Code.
7. To Perform Synthesis using Xilinx ISE tool for Spartan6 Family of FPGA.
8. To analyse the timing, area reports for the design.
9. To compare the synthesis results of conventional AES, single stage pipelined AES, fully pipelined AES with respect to speed.

III. PROBLEM FORMULATION

Everyday millions of people create large amounts of information in different fields like telecommunication, financial services, banking etc. It is very essential to protect this information at the time of transmission so AES is used for the securities taken at the input data stream of 128bits and encrypt it to get the result cipher of 128bits.It maintains 3

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

different key lengths and with every key different round is related. For a 256bit key it has 20 rounds and for 192bit key it has 12 rounds and for 256bit key it has 14 rounds. The four Transformation's in Advance Encryption Standard & i.e. Substitution-bytes, Shift-rows, and Mix-columns & Add-round Key. This design make use of an effective way of implementing the 256bit key AES encryption by performing multiple stages of pipelining for both AES Rounds, Key Expansions and By building an LUT bases Architecture to handle Mix Columns block which deals with crypto multiplication. Hence increase the operational frequency of the AES Architecture which is needed for the encryption. This decreases critical path delay for the overall design and hence improves overall timing performance.

IV. METHODOLOGY

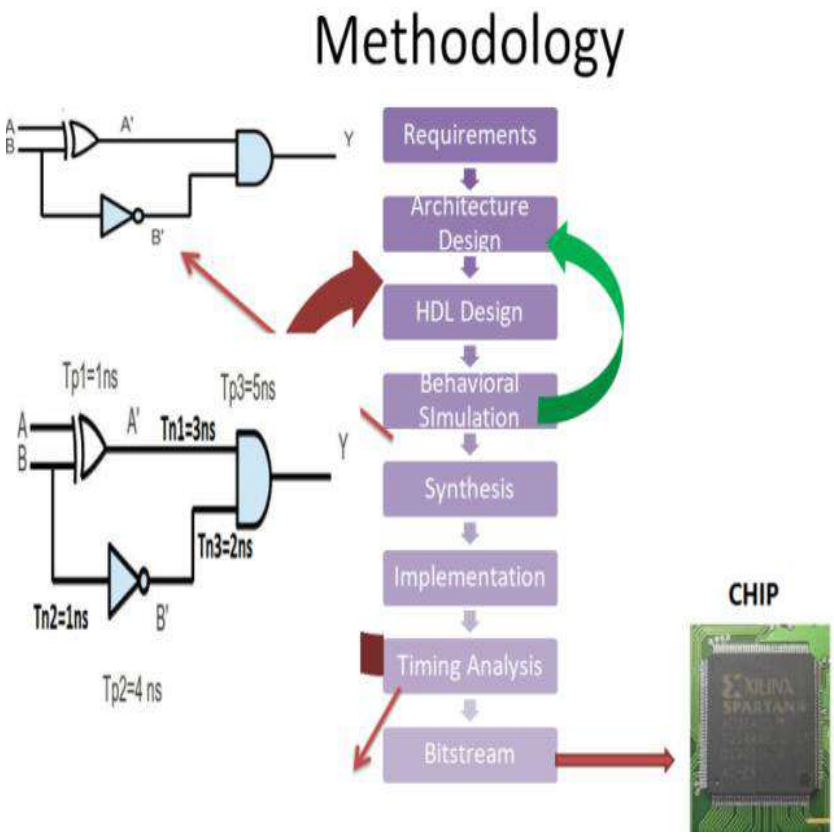


Fig. 5 Methodology

Our proposed methodology mainly follows the standard methodology that is industry based Field Programmable Gate Array methodology. The above shows the self explanatory flow of methodology which is followed in our project in-terms of VLSI. According to this methodology based on literature survey of AES algorithm implementation our project proposes a modified AES architecture. In our work, high speed implementations of Advance Encryption Standard Architecture is Proposed. The high speed can be achieved by thought of pipelining & look up table Based Architecture.

Distributed LUT based architecture: AES was integrated with S-box as a single look up table based on designing. width of about 8bits, depth of about 256. 2byte input data was addressed to look up table which is nothing but a ROM. Output of ROM is the value which substitutes the input. One of the most critical blocks of AES is S-box module which is employed in both sub-bytes and key expansion.

Pipelining in AES Encryption:

The architecture which is used in AES encryption is the modification of iterative loop architecture. Here records are used and these records are added in between the two rounds. The records help us to achieve the channeling of AES. Pipelining is nothing but processing of information/data which is given as input in Continuous way without even waiting for current processing input to finish over. Idea of Pipelining can be seen in various processors. In our Architecture records are being used to save outputs of the current Round which is running. Output of each round can be stored in these records instead of directly passing these records to the next rounds.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

V. DESIGN & IMPLEMENTATION

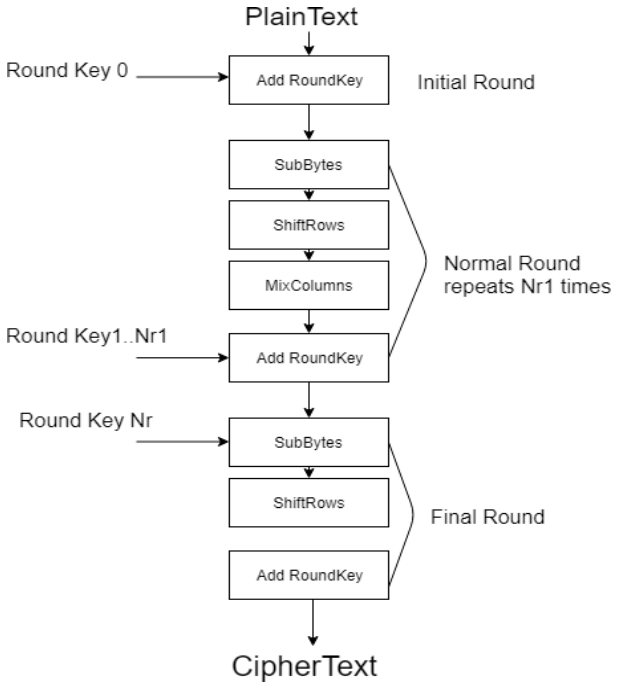


Fig. 6 Hardware I/O of AES

The above figure shows the hardware I/O of AES. It contains Four Transformations and they are Sub-Bytes, Shift-Rows, and Mix-Column & Add-Round Keys. If input given is plain text along with cipher key then we get cipher text as the output and vice versa. The size of plain text as well as the key size must be same and this should satisfy block cipher technique. The available key sizes in AES are 128bit, 192bit and 256bit. The Control Signal's are Start-Encryption, Start-Decryption & Start Key-Generation & Encryption/Decryption. The above given Control Signal's are utilized for Control of Proper sequence arrangement in AES. By using controller we will control these control signals. The controller is associated with key expander, encryption and decryption. The input information along with the key is processed and the corresponding output is obtained at output terminal.

Proposed substitute box design (S-BOX) Design & Implementation:

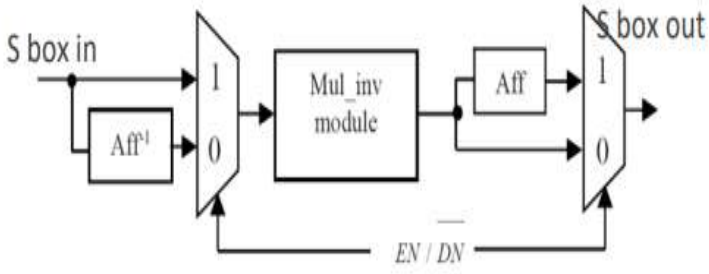


Fig. 7 Block diagram of S-box

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

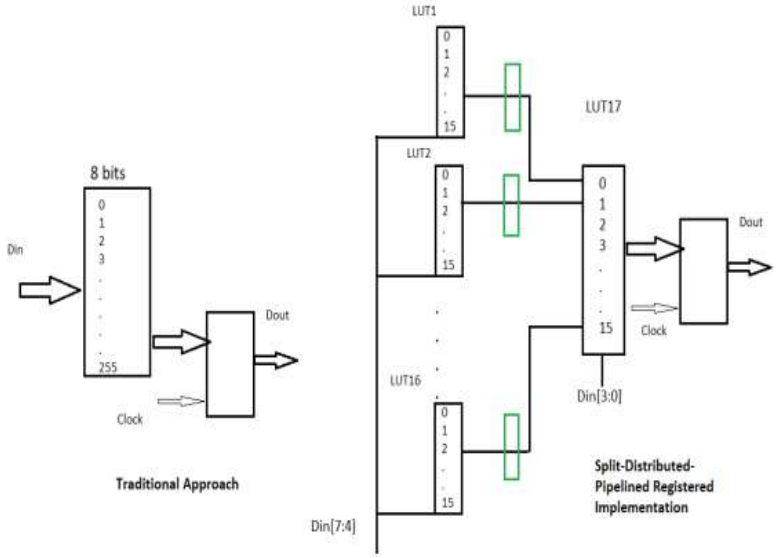


Fig. 8 S-BOX is implemented via a ROM

The above figure represents the S-box. The s-box serves as a look up table. S-box is implemented via a ROM design where all the values are stored and fetched on Input which acts as Address, nothing but the input value.

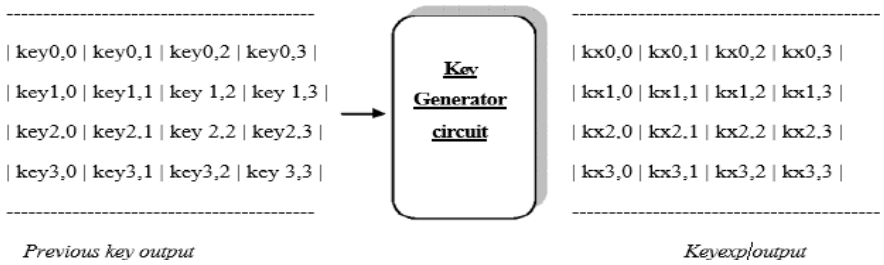


Fig. 9 Key generator circuit

Key generator is the circuit which performs the functions to obtain the unique key for each and every round. In figure 9 we can see how key generator circuit is added in between the rounds. By key generator circuit security level will be increased and attackers who try to hack the system will not be able to break the system.

Introduction to image processing

Another essential part in communication and multimedia world is image data security. The data of image can be protected by converting the original data into an unknown form. The data may be in the form of image or information or signal etc. For image data security cryptography is one of the best methods. Encoding means it is the conversion of Original Data to Unknown format. Restoring Encrypted Data into Original form known as Decoding.

Implementation of AES algorithm can be done by using MATLAB. The image will be sent as Input & by Applying Advance Encryption Standard Algorithm it generates a Cipher-Image. Cipher-Image is Input to Decryption Algorithm that Reconstruct's back to the Original-Image.

By MATLAB Coding & Implementation of Advance Encryption Standard Algorithm Design is Synthesized & Simulated in both Encryption & Decryption.

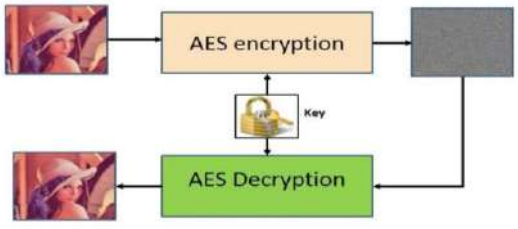


Fig. 10 Image Encryption/Decryption Process

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

Blocks are given which is implemented step by step by using AES algorithm for image encryption and decryption. It converts plain text into cipher text can be applied using this algorithm and can also convert back to the original text

VI. RESULTS AND DISCUSSION

AES 256 Bit Encryption results

Table 5.1 of Test Case-1

Input	FF112233445566778899aabbccddeeff
Key	000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Expected output	8f87a14f92531d3ba2950f2276ad4786
Result Obtained	8f87a14f92531d3ba2950f2276ad4786
Status	Pass

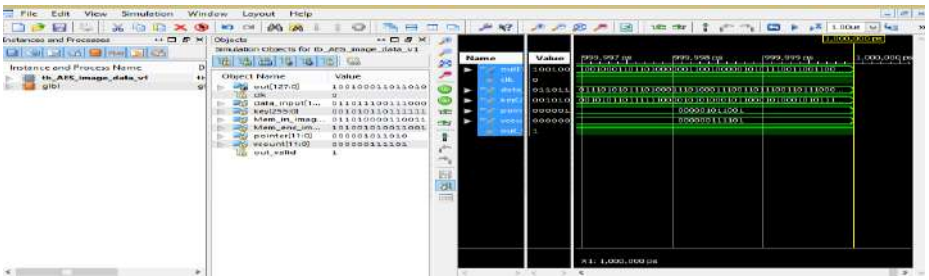


Fig. 11 AES Data Image

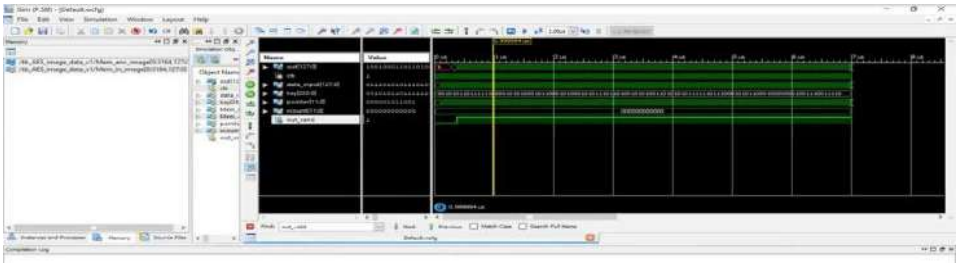


Fig. 12 Memory in Image

Inputs

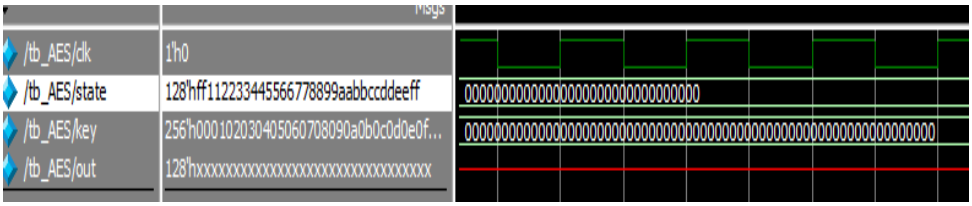


Fig. 13 Input waveform

Outputs

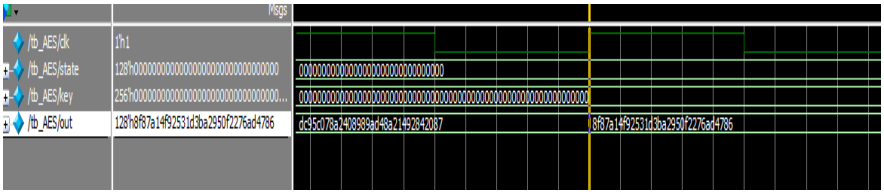


Fig. 14 Output waveform

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

AES Verification with Online Calculator

Fig. 15 AES verification using online calculator

Key generation Outputs

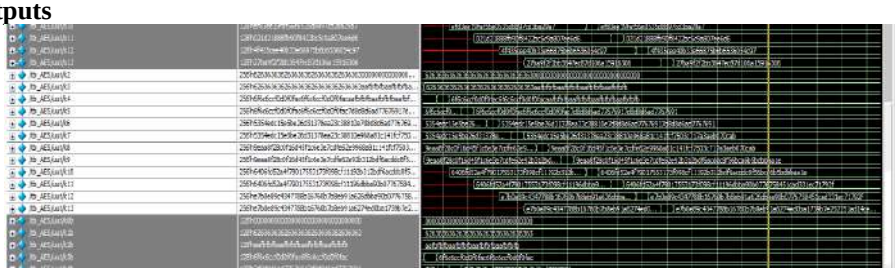


Fig. 16 Key generation output waveform

Each Round Outputs

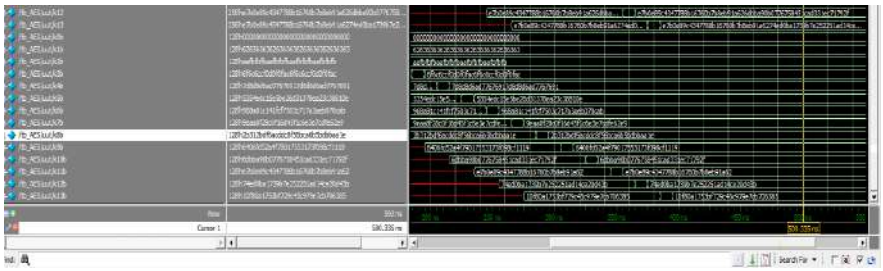


Fig. 17 Each round output waveform

Method	Existing	Proposed
Total cycles for encryption	56	25
Frequency(MHz)	161	445
Throughput(Mbps)	278	528
On chip power(w)	0.58	0.74
Slice LUT	1814	7680
Slice register	836	6484
LUT Flip flop	434	6457

From the above table we can compare the parameters with existing and proposed method. The overall speed of the design is almost increased by 3 times compared to the existing architecture which is very good improvement in terms of speed.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

Device Utilization

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	6848	607200	1%
Number of Slice LUTs	7686	303600	2%
Number of fully used LUT-FF pairs	6407	8121	78%
Number of bonded IOBs	769	600	128%
Number of Block RAM/FIFO	121	1030	11%
Number of BUFG/BUFGCTRL/BUFHCEs	1	200	0%
Number of DSP48Es	88	3800	2%

Timing

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 2.247ns (Maximum Frequency: 445.077MHz)
   Minimum input arrival time before clock: 10.299ns
   Maximum output required time after clock: 0.511ns
   Maximum combinational path delay: No path found

=========================================================================

Process "Synthesize - XST" completed successfully
```

Image Encryption

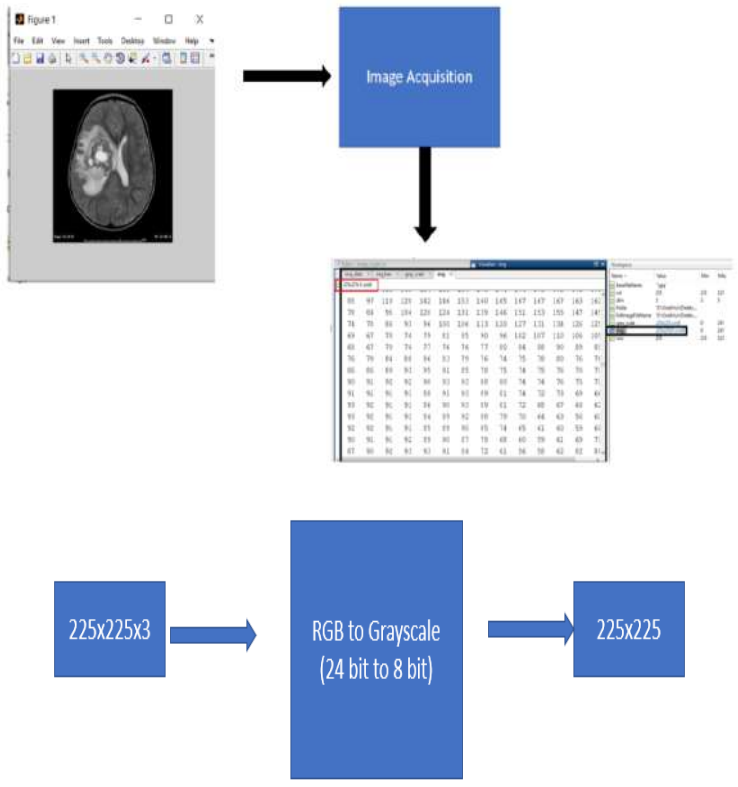


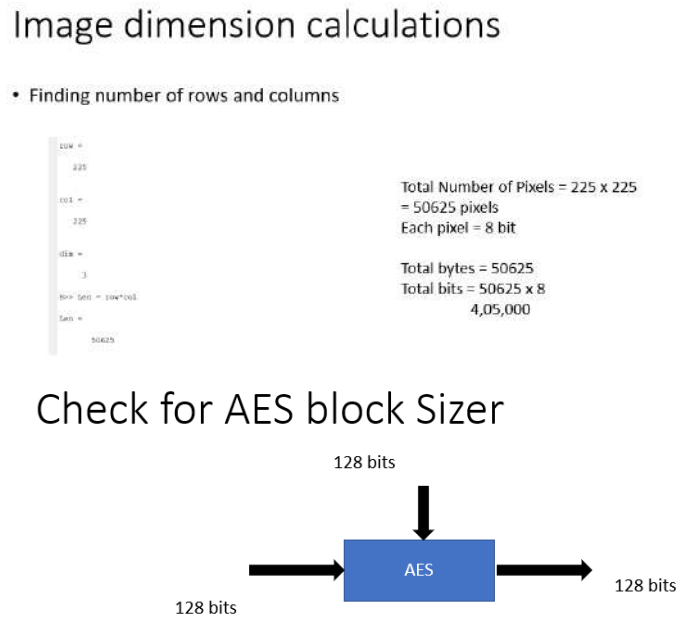
Fig. 18 Block diagram of image encryption

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755



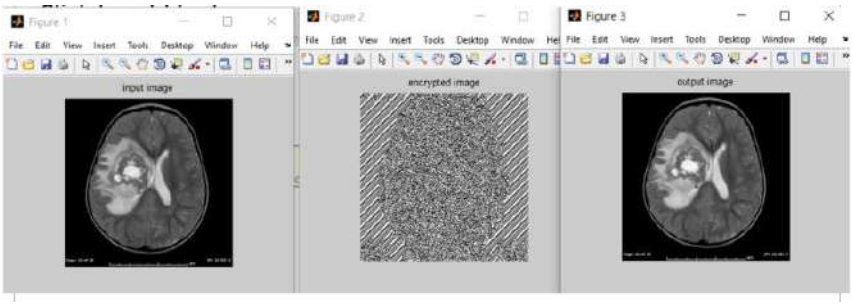


Fig. 19 Image Encryption

Correct Key

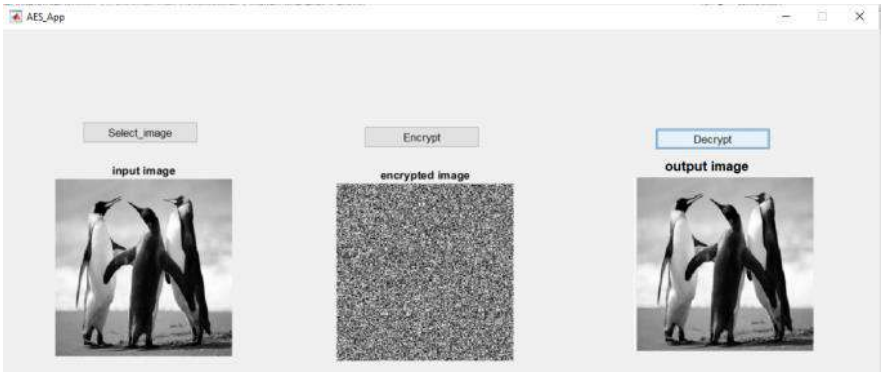


Fig. 20 output image for correct key

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⁜ Impact Factor 7.105 ⁜ Vol. 9, Issue 7, July 2022

DOI: 10.17148/IARJSET.2022.9755

Wrong Key

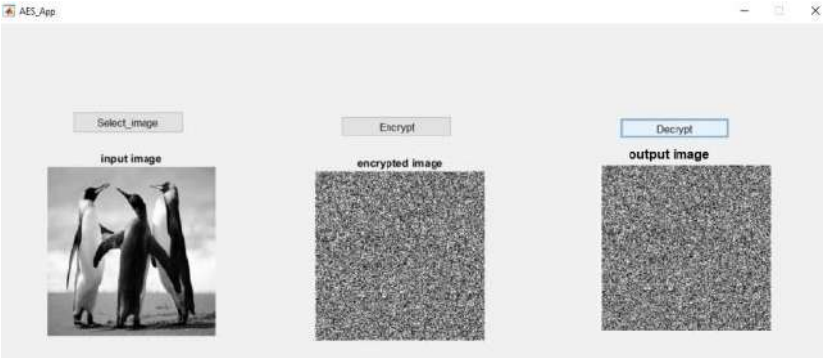


Fig. 21 output image for wrong key

VII. HARDWARE AND SOFTWARE REQUIREMENTS

Software: Xilix ISE, Xilinx VIVADO, Modelsim, MATLAB.
Langauge: Verilog HDL, MATLAB

- For Verilog HDL simulation
  Modelsim Simulator,
  Xilinx ISE, ISIM.

- For Synthesis, Hardware Generation
  Xilinx ISE 14.5
  Xilinx Vivado
  Supporting software's
  MATLAB 2013

VIII. CONCLUSION

In this paper, we have done implementation on FPGA (xc7vx485tffg1157). The speed is increased by three times when compared to existing architecture which is very good improvement in terms of speed. At the same time there is a trade off, if the speed improves the area also increased, if we try to optimize for area speed might decrease. Our objective was to obtain very high speed which is achieved, as per area is concerned there is enough space on FPGA; the requirement is to get higher speed.

REFERENCE

[1] S. Oukili and S. Bri, "High speed efficient advanced encryption standard implementation," 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, 2017, pp. 1-4.
[2] S. Oukili, S. Bri and A. V. S. Kumar, "High speed efficient FPGA implementation of pipelined AES S-Box," 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, 2016, pp. 901-905.
[3] R. R. Rachh, P. V. AnandaMohan and B. S. Anami, "High-speed architecture for Advanced Encryption Standard Algorithm," 2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, Hyderabad, 2012, pp. 167-172.
[4] S. Burman, P. Rangababu and K. Datta, "Development of dynamic reconfiguration implementation of AES on FPGA platform," 2017 Devices for Integrated Circuit (DevIC), Kalyani, 2017, pp. 247-251.
[5] W. Wang, J. Chen and F. Xu, "An implementation of AES algorithm Based on FPGA," 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, Sichuan, 2012, pp. 1615-1617.
[6] S. N. Dhanuskodi and D. Holcomb, "An improved clocking methodology for energy efficient low area AES architectures using register renaming," 2017 IEEE/ACM International Symposium on Low Power Electronics and Design
[7] Y. Wang, A. Kumar and Y. Ha, "FPGA-based high throughput XTS-AES encryption/decryption for storage area network," 2014 International Conference on Field-Programmable Technology (FPT), Shanghai, 2014, pp. 268-271.