A Low Power, Area Efficient Implementation of AES Algorithm

S.Neelima, R.Brindha

Abstract*: Encryption is a procedure of convert readable information into encoded appearance so that it can't be interpreted by the intruder. Paper presents the FPGA implementation of a low power, neighborhood efficient AES algorithm for encrypting data. From the results it has been experimental that the enhanced technique has reduced the power consumption and area compared to the existing methods. The implementation is done in 90 nm and 65 nm CMOS technology using Quartus for Cyclone II and Cyclone III.

Keywords* : Advanced Encryption Standard, Security, FPGA implementation, Low power, and area efficiency.

I. INTRODUCTION

Encryption is the procedure of convert the information from understandable arrangement into encoded format. There are two types of encryption processes named as keyed and keyless. The keyed algorithms are most used algorithms for encryption process and they are of two types namely symmetric and asymmetric key algorithms. The symmetric uses same key for both encryption and decryption. The most advanced symmetric key algorithm is AES. AES is an iterative process .This algorithm follows different input block sizes such as 128, 192, and 256 and follows the same processes (sub bytes, shift rows, mix columns) for number of rounds. The set of rounds is depending on block size of the input data that is AES-128, AES-192 and AES-256 performs 10, 12 and 14 rounds respectively. The AES algorithm is fully based on permutation and substitution network. It performs a series of linked operations that involves replacing input values by predefined fixed values (sub bytes) and also performs shuffling (shift row, mix column) operation. The input data are considered in terms of bits (128bits, 192bits, 256bits). But the operations performed on input are in terms of bytes. For example, AES considers 128bit data as 16 byte which are arranged as matrix which consisting of four rows and four columns. Thus 192bit and 256bit are considered as 24, 32bytes respectively.

In this paper, an AES algorithm which provides high security and consumes less power is presented. The remaining paper is organized in the following manner: The subsequently division a detailed literature survey covering the past research carried out on various encryption standards is presented. The background methodology is existing in section III. Section IV presents the enhanced structure and usage of the procedure. Section V talks about the exploratory arrangement and results acquired pursued by the end and reference.

II. LITERATURE REVIEW

Raeed et al (2015) [1] proposed an Anti-Collision improvement of a SHA-1 Digest by AES Encryption using LABVIEW. Implementation of cryptography hash function was done in LabVIEW. The presented 128 bits code generation gives strong code for both encryption and decryption . Even though the labview based method was effective in the implementation of the method, the hardware part is difficult. In recent years, the AES along with Galois Counter Mode (GCM) operation was adopted for high throughput network in authenticated encryption. But there was other systems in literature implemented using different versions of AES-GCM Core. Buhrow et al (2015) [2] proposed a similar hybrid AES-GCM Core for authentic encryption of 400 Gb/s network protocol in highly parallel architecture. The implementation was based on AES with GCM parallelization to widen the data interface. This approach can accommodate larger designs but have I/O port limitation when implemented in FPGA. Dalakoti et al (2015) [3] proposed efficient AES for image processing with high throughput and implementation was carried out based on parallel processing with key expansion technique. The advantage of the presented method was less number of hardware usages in the implementation of AES. The implemented design has high speed, less complex and high throughput for image encryption. Deshpande et al (2015) [4] presented a task level parallelism which achieves less area and high throughput by operating three concurrently working AES modules. With the area optimization algorithms, the system works with the frequency of about 239.648MHz and through put of 5.751Gbps.AES encryption engines using parallel, pipeline and sequential technique were implemented in FPGA. The FSM method proves to be very proficient in terms of area and throughput. To implement this algorithm Zinc device was utilized and tested on Zedboard. The limitations were low speed and delay.

A similar pipelining and enhanced key expansion method was proposed by Liu et al (2015) [6] with high throughput and safe AES. The difficulty of cracking keys was

Revised Manuscript Received on July 22, 2019.

S.Neelima, R.Brindha
Asst Professor, Gandhiji Institute of Science and Technology, Jaggayyapet. Krishna District, Andra Pradesh.
Former Professor in ECE, Faculty of Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamilnadu
Corresponding Author email: seethala.neelimakesav@gmail.com,

increased up to 2(N − 1) times in N-round AES using additional nonlinear operations. The key expansion was tested in a two-stage pipelining blueprint and deep pipelining blueprint. From the analysis the deep pipelining was found favorable and was compared with other AES implementations. Mateur et el also implemented a pipelining hardware implementation of AES algorithm in Spartan 3 in 2016. The author achieved a throughput of 6900 mbit/s.

Nadjia et al (2015) [8] proposed AES Intellectual Property core for hybrid cryptosystem RSA-AES. The paper presents three AES hardware architectures in Serial/Serial, Parallel /Serial and Parallel/Pipelined structures. This structural design allows four 32-bit words to be performed jointly and round is execute in pipeline, which gives a higher encryption/decryption rate. A similar work also implemented by Prachi V Bhalerao et el "Hardware accomplishment of cryptosystem by AES algorithm using FPGA" (11) by using triple key AES algorithm implemented in Spartan 3 to improve the security of the data.Pravin B. Ghewari et el(12) proposed "Efficient hardware design and implementation of AES cryptosystem" in Virtex and achieved the better throughput of 352mbit/s.

A similar work was proposed with AES by Khedlekar et al (7). The presented smart secure system using parallel AES provides security in the transmission of information. The advantage of the presented method was to implement a secure smart system to store the files of any size on a server which was totally confidential and secure. The results show that the computational time of parallel computation was low when compared to the sequential computing. Ashwini et al (2014) proposed an FPGA based accomplishment of the AES for electronic data transmission using S-Box. The advantage of the proposed system was the clock speed. Shihai zhu et el(13) proposed "Hardware uses of AES encoding and decoding scheme based on FPGA" in 2015 using pipeline structure using Quartus II and synthesized using modelsim software and achieved a neutral speed of the device.

III. BACKGROUND METHODOLOGY

A. AES

AES is the algorithm for the encoding of electronic information from different source. The algorithm changes over the information into encoded position with the assistance of a key; the encrypted information can't be perused by a third individual. The recipient can peruse the encrypted information after the unscrambling is perform. The means followed in the algorithm are given bellow,

Table 1 Algorithm steps for AES Encryption standard

Initial round	Add round key
First round	Sub bytes
	Shift rows
	Mix column
	Add round key
Final round (no mix columns)	Sub bytes
	Shift rows

During the time spent encryption and decoding the algorithm pursues same advances more than once. Each progression is measured as round. The quantity of rounds relies upon the square size of info information to be altered. The size of information square shifts dependent on the client prerequisite.

- Add round key

Include round key is the progression which is finished all round of encryption and decoding process, where the info and key produced from key age algorithm are consolidated together. For that the algorithm perform bitwise XOR task on info and key.

This process is clearly expressed in following mathematical Equation (1),

$$\mathbf{S_{i,j} = A_{i,j} \oplus k_{i,j}} \quad (1)$$

Where A is the Information matrix given in 8X8 and K is the key matrix given in 8X8.

S is the substitution box matrix.

- Sub bytes

Information bytes are delineated as 4x4 networks. The information matrix is likewise called as state matrix. This is put away regarding lookup table. The look into table is name as S-BOX (Substitution Box). In the previously mentioned venture every byte in info matrix is supplanted by every byte from the query table. This is appeared in figure.1. The state matrix is expressed in Equation (2).

$$A_{i,j} = \begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{bmatrix} \quad (2)$$

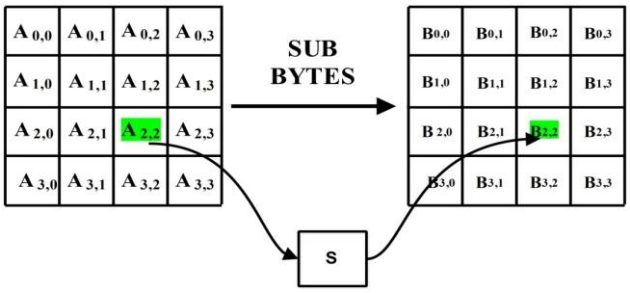


Figure 1. Performing substitution bytes from S-Box

- Shift rows

In this progression every byte in the condition matrix is moved in a specific technique, the moving strategy is clarified in Figure.2.

$$B'_{r,c} = B_{r,(c+shift(r,N_b))mod N_b} \quad (3)$$

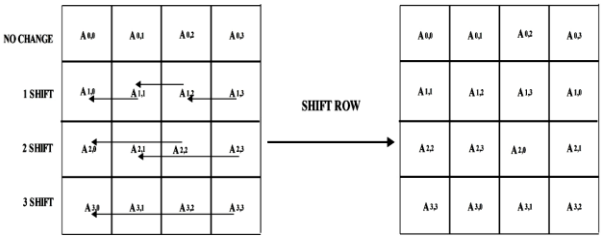


Figure.2 Performing shifting of rows.

In figure 2 it tends to be seen to facilitate the main column continues as before during the whole procedure. The subsequent line experiences total byte move by one sub square. Essentially the 3 and 3 line experience moves 2's and 3's individually.

- Mix column

During blend section activity every segment from information state matrix is increased with the predefined steady matrix. The subsequent section is shown as the comparing segment. So each information matrix will influence every segment of coming about matrix.

$$A(x) = a(x) \otimes B(x) \bmod (x^4 + 1) \quad (4)$$

Where, a(x) - Predefined fixed matrix,
B(x) – Input state matrix,
A(x) – Output state matrix.

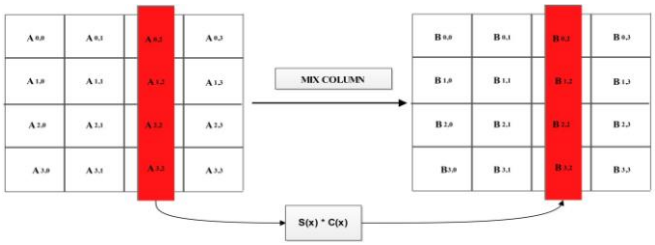


Figure.3 Process Diagram generating Mix column.

IV. PROPOSED METHOD

The blue print of the proposed design is exposed in Figure.4. The AES algorithm uses 128 bits, and the algorithm has 10 rounds. 128 bit will be considered as 16 bytes. These 16 bytes will be considered as block size (Nb). From the value of block size (Nb) the number of rounds will be calculated from the following formula. Nr= (Nb/4) +6. Here Nr stands for number of rounds. At the initial stage add round key alone performed. Calculations performed in first round to ninth round are shown in Figure.4. At the final round, except mix column remaining process were done. The outcome encryption unit highlights the 128 bit cipher text.

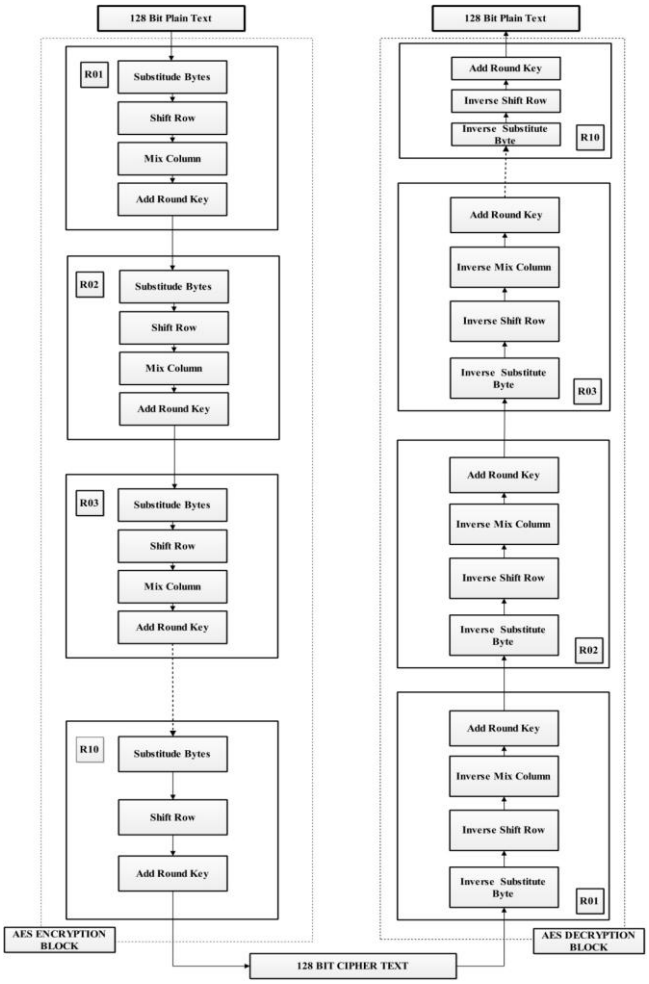


Figure.4 Block diagram of conventional AES algorithm (Block 1)

In the conventional algorithm each round has separate 16X16 Sub-Bytes. To adopt these values separate memory location is required for individual rounds. So it results in requirement of FPGA with more input pins. It leads to power and time consuming process. To reduce this, the new algorithm is proposed in terms of physical architecture. The new design consists of mono LUT; it holds the entire predefined functions. The algorithm can read the data required for Sub-Bytes from the single LUT throughout the execution of encryption process. The physical architecture of proposed algorithm is represented in the Figure 5.

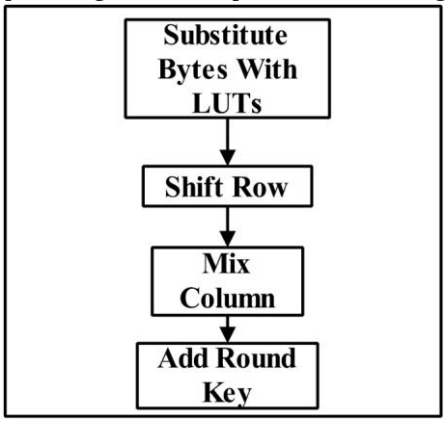


Figure.5 Block diagram of single round in proposed AES algorithm with LUT (Block 1)

V. RESULTS AND DISCUSSION

A. Results analysis of proposed method

The algorithms displayed in this paper are executed in various advancements like 65nm and 90nm. The metrics like power, postponement and LUT are estimated and classified. The Table 2 demonstrates the exhibition of the current AES algorithm. The power in 65nm is improved by about 47% in cyclone and contrasted with stratix II, the cyclone III execution was improved by about 58% . Essentially the presentation towards deferral is likewise improved by about 16%. Table 2 demonstrates the presentation of AES algorithm. Power, deferral and LUT of AES for different nanometer innovation are appeared in Table 2. At the point when power is measured, the stratix II unit gives a 20% development in power when contrasted with Cyclone II. The deferral is decreased by about 50% The RTL view of AES rounds is shown in Figure 6. From table 2 the LUT consumption is assessed with our design and the existing methods. Thus the LUT consumption has reduced compared with the other designs except the dalkoti method.

COMPARISION OF LUT WITH OTHER MRTHODOLGIES							
OUR DESIGN		AES-GSM PARALLEL CORE METHOD (BUHROW)	AES ENCRYPTION CORE PROCESS (DESHPANDE)	AES-IMAGE PROCESSING (DALKOTI)	AES-RSA (ANANE)	AES PARALLEL METHOD (PRIYA)	AES-FINE PIPELINING STRUCTURE (LIU)
CYCLONE II	CYCLONE III	VIRTEX-5					
457	459	443,724	1394	389	2534	3168	2444

Table 3. Performance analysis of AES algorithm in different technology.

Parameters	Cyclone II(90 nm)	Cyclone III(65 nm)	Stratix II(90 nm)
Power (mW)	155.48	140.6	187.50
Delay (nS)	12.503	11.555	12.007
LUT	457	459	256

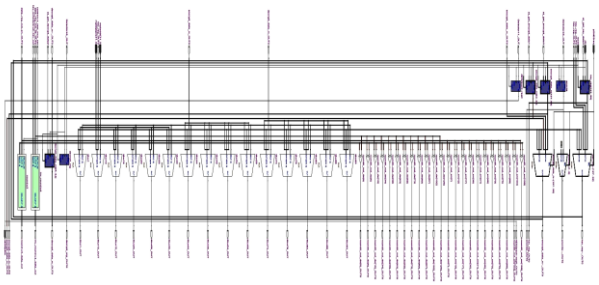


Figure.6 RTL views of the AES algorithm.

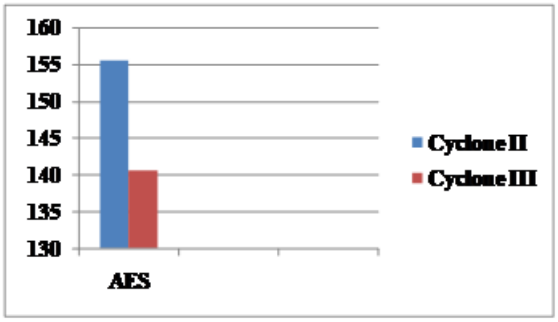


Figure.7 Comparison of Power Consumption (in milli watts) AES.

Figure 7, 8 and 9 compares the different attributes of the algorithm when implemented in different device families. Cyclone II and Cyclone III are taken into consideration for the design implementation and each device differs in its performances.

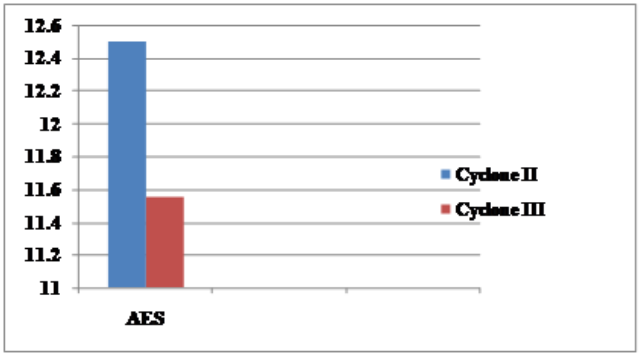


Figure.8 Comparison of Delay (in nano seconds) AES.

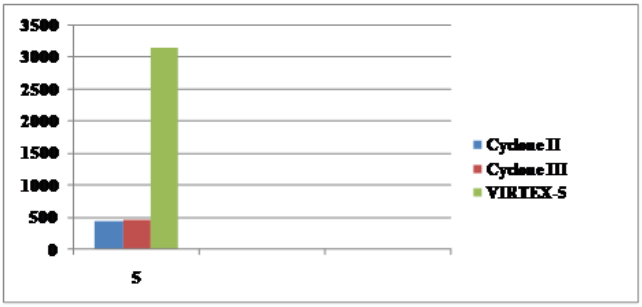


Figure.9 Comparison of LUT AES.

B. Security Analysis of Proposed Method

Security is the key term of AES. Security of AES implies how safe this framework is against dynamic or inactive assault. We determine the security dependent on three

criteria.

- Time Security
- Avalanche Effect
- Strict Avalanche Condition

- Time Security

It speaks to the "capacity of cryptographic technique towards brute force attack with various key size and time it takes to effectively mount a brute force attack. Brute force attack implies that totally checking all possible key blends until the intruder gets the first key. For a 3 bit key, Brute force attack will take most extreme 8 rounds to check each possible key course of action". Most extreme key blend for various key sizes is given in Table 3 (a). From Table 3 (b), for 128 bit key brute force attack needs to check greatest 3.403 x 1038 quantities of key mixes. Then again the brute force attack is a period based procedure. Handling speed of this attack is contrasted and the typical super PC.

Table 3 (a) Maximum Key Combination

Key size	Possible Combination
1 bit	2
2 bit	4
4 bit	16
8 bit	256
16 bit	65536
32 bit	4294967296
64 bit	1.8447×10^{19}
128 bit	3.403×10^{38}
192 bit	6.278×10^{57}
256 bit	1.158×10^{77}

Table 3 (b) Estimation of Years to Break AES

No	Key size	Years Need
1	128 bit	3.19 x 1014 years
2	192 bit	5.88 x 1033 years
3	256 bit	1.0844 x 1053 years

As shown in Table 3 (b), even with a PC, it would take 1 billion years to crack the 128-bit AES key using brute force attack.

- Hamming Distance

Hamming separation is a binary digit, which is utilized to recognize the variety between two binary strings. It is a little term utilized in the data security examination equation. "AES-128 algorithm utilizes 128 bit square of plaintext and 128 bit key. With the quickest Super Computer of this age it will take 3.19 x 1014 years to squeak a key mix through brute force attack. So it isn't possible for a human as well as for an age to break a key with checking every single key blend". In our application the two binary strings used are two different encrypted messages with small changes in plain text. The observations are given in table 4 with the avalanche effect.

Avalanche Effect

The avalanche property states that the change in a single bit of input plain text or secret key creates a significant change in the output. Then the cryptographic system becomes more effective one (Yuanqing Deng et al 2011).

If a cryptographic system does not follow the avalanche effect, the attacker can easily guess the plain text from the cipher text. The computation for avalanche effect is expressed in equation (5)

$$Avalanche\ \ Effect = ((\text{Hamming Distance}/\text{Block size}) X 100)\% \quad (5)$$

Table. 4 Avalanche Effect for fixed key 128 bit

No	Plain text (Alphabet)	Cipher text (Hex.)	Hamming Distance	Avalanche (%)
1	ABCDEFGHIJKLMNOP	9CDD85DE85B48BED892F02D8A5CBDACB	63	49.22
2	ABCDEFGHIJKLMNOQ	ACE7083761553A6B3A97BCB1740B176A		
3	ABCDEFGHIJKLMNOB	0026D76C52B61B9A76445035FD4D342B	69	53.91
4	ABCDEFGHIJKLMNOC	E930AC10030FA5DB617AF6DFA741ADE4		
5	ABCDEFGHIJKLMNOS	DA5D2C1E67818646AC2D955E0FAB4C3B	61	47.66
6	ABCDEFGHIJKLMNOR	7A6EEC02FCADA2FB323D672		

Security analysis of AES with Power analysis attack

The power analysis is carried out by calculating the power consumption at the time of encryption. There is a relation between power consumption and cryptographic process. There are two types of power analysis method is available for the encryption device such as, "Simple Power Analysis" (SPA) and "Differential Power Analysis"(DPA).

In SPA, it tries to get the secret key by understanding the power traces of the encryption device. It can be able to get the secret key at the time of execution only. But through this method it is difficult to understand the secret key, because AES-128 consists of 10 rounds, at each round it generate different key.

DPA attack uses the statistical method to identify

the secret key. By using statistical method the differences in the power traces are analysed to guess the secret key but this method is difficult due to the large number of rounds.

DPA can be performed as follows:

- Select an intermediate value that depends on data and key vi,k = f(di, k)
- Measure power traces ti,j while encrypting data di
- Build a matrix of supposed halfway qualities inside the figure for all possible keys and traces vi,k
- Using a power model, process the matrix of theoretical power utilization for all keys and traces hi,k
- Statistically assess which key hypothesis best match the deliberate power at every individual time

Implementation for finding Secret Key of AES-128 by DPA

The DPA is used to attack the AES-128. The hypothetical power consumption cannot match with the power traces of the cipher text. This is shown in Figure 10. In Figure 11, the secret key is represented in terms of bytes. Each byte carries 0x00.So the DPA had failed to guess the AES secret key.

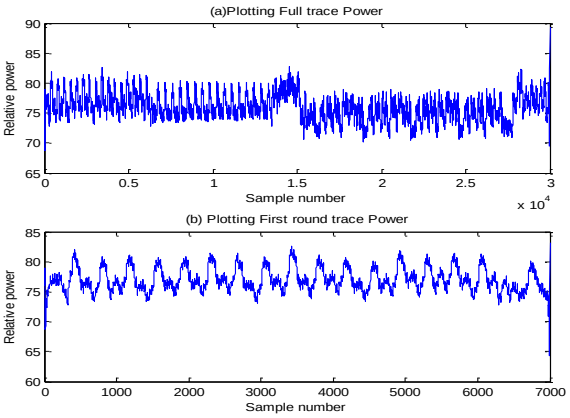


Figure 10 Power trace model

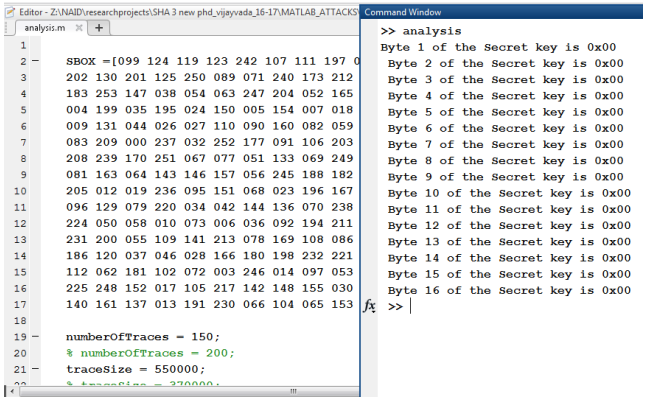


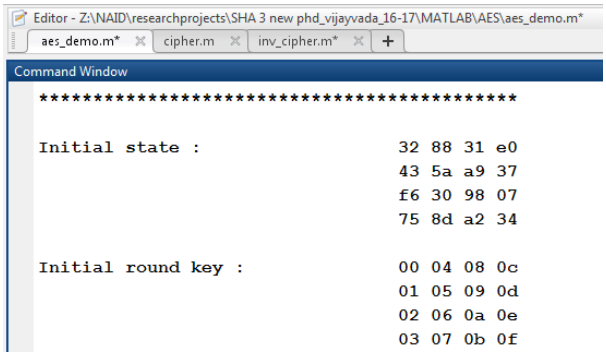
Figure 11 Deduction of Secret Key

The validation of the algorithm is checked using MATLAB. Above figure shown in figure 12 and 13 respectively. The Cipher Text used for encryption is a2 ba 77 96 a3 f1 85 ef 7e 20 c9 98 93 3b 7d 7d.

Results

ENCRYPTION PROCESS

Plain Text: '32' '43' 'f6' '75' '88' '5a' '30' '8d' '31' 'a9' '98' 'a2' 'e0' '37' '07' '34'



State at start of round 1 : 32 8c 39 ec
42 5f a0 3a
f4 36 92 09
76 8a a9 3b

State at start of round 2 : 77 18 a4 f3
3f 2b 0b e1
3b bc 0d f0
c7 c7 04 23

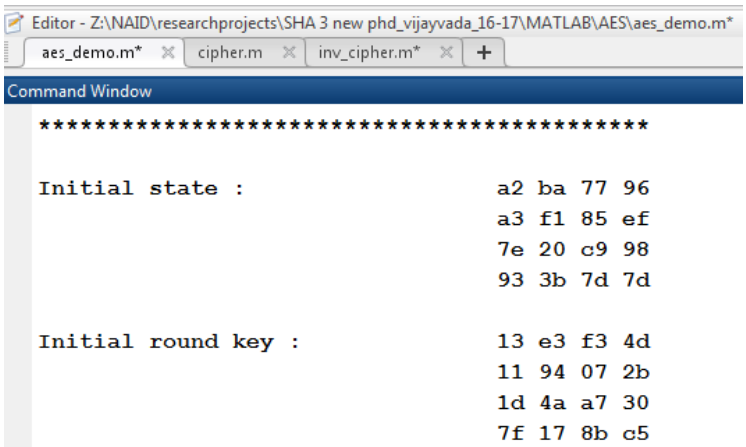
```
State at start of round 3 :      be 12 1b 7a  State at start of round 4 :      26 87 89 1a
                                 da 8f c2 8a                                   af 15 c7 bc
                                 14 69 fa 0c                                   42 15 41 13
                                 65 2d 56 06                                   27 be 7a e3

State at start of round 5 :      cb 5b 81 aa  State at start of round 6 :      24 d0 0f 30
                                 3d fe db 64                                   35 0d 0a c4
                                 77 5a 41 91                                   8f ea 88 d5
                                 46 02 40 8f                                   17 2c 1b ce

State at start of round 7 :      1f 4d 6d 8b  State at start of round 8 :      20 56 79 53
                                 66 ed ff ad                                   fe 42 f8 53
                                 fb 88 22 72                                   86 a1 41 5c
                                 39 99 d6 33                                   1a c0 54 61

State at start of round 9 :      2a f6 45 ab  State at start of final round :  56 15 4f 9f
                                 dd 53 17 49                                   88 3e bc 11
                                 2b fc bc 28                                   45 6f 00 58
                                 9d 25 f3 37                                   42 d6 9a 83
```

Cipher Text:

a2 ba 77 96
a3 f1 85 ef
7e 20 c9 98
93 3b 7d 7d

Figure 12 Encryption method

DECRYPTION PROCESS




```
State at start of round 1 :      f5 ad 49 0d  State at start of round 2 :      ae c9 af da
                                 f1 2b f8 75                                   73 25 7e 57
                                 d7 8c e2 65                                   2d fe fa f9
                                 26 c6 c6 f2                                   6f 4d d8 b1
```

```
State at start of round 3 :   f7 17 a7 a2
                              59 c6 65 79
                              83 7d 2c 59
                              11 cc ae da

State at start of round 4 :   1f 39 0c ac
                              bb b9 43 27
                              83 81 f5 be
                              73 5a 77 09

State at start of round 7 :   b7 b1 b6 ed
                              2c 41 ed bb
                              83 4a 44 32
                              ef a2 ba 20

State at start of round 9 :   b1 59 84 db
                              b2 65 82 c4
                              63 6a 6e a8
                              ec 2c f6 b8

State at start of round 5 :   36 70 76 04
                              d7 67 1c 96
                              c4 03 73 87
                              8b f0 71 af

State at start of round 6 :   c0 e3 3c 3d
                              55 16 95 33
                              93 40 0f c4
                              c3 12 ee f6

State at start of round 8 :   e5 42 6e 62
                              ed f0 3b c1
                              65 34 f1 b0
                              9a 5e 3f 0d

State at start of final round : 23 64 12 ce
                              cf e0 80 2c
                              4f 01 bf 05
                              e2 38 7e d3
```

Decrypted cipher text:

32 88 31 e0
43 5a a9 37
f6 30 98 07
75 8d a2 34

Figure 13 Decryption method

VI. CONCLUSION

An Advanced Encryption Standard has been proposed for converting readable data into encoded form. FPGA achievement of a proposed AES algorithm for encrypting data during transmission was carried out in different 95nm and 65 nm CMOS technology using Quartus for Cyclone II and Cyclone III and the LUTs consumption was reduced to 65% compared to existing methods and the delay is reduced to 50% when compared with the Stratix II. The decryption was validated using MATLAB software. In future, the work will be implemented for real time database. A System on Chip architecture will be proposed for the implementation of the proposed method with less chip area and power.

REFERENCES

[1] Raaed K. Ibraheem, Roula AJ. Kadhim and Ali SH. Alkhalid, 'Anti-Collision Enhancement of a SHA-1 Digest Using AES Encryption By LABVIEW', World Congress on Information Technology and Computer Application, 2015.

[2] Benjamin Buhrow, Karl Fritz, Barry Gilbert and Erik Daniel, 'A Highly Parallel AES-GCM Core for Authenticated Encryption of 400 Gb/s Network Protocols', International Conference on ReConFigurable Computing and FPGAs, Pp. 1 – 7, 2015.

[3] Neha Dalakoti , Nidhi Gaury and Anu Mehra z, 'Hardware Efficient AES for Image Processing with High Throughput', 1st International Conference on Next Generation Computing Technologies (NGCT), Pp. 932 – 935, 2015.

[4] Pournima U. Deshpande and Smita A. Bhosale, 'AES Encryption Engines of Many Core Processor Arrays on FPGA by Using Parallel, Pipeline andSequential Technique', International Conference on Energy Systems and Applications (ICESA), Pp. 75-80, 2015.

[5] Qiang Liu, Zhenyu Xu and Ye Yuan, 'High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion', IET Computers & Digital Techniques, Vol. 9, No. 3, Pp. 175–184, 2015.

[6] Anane Nadjia and Anane Mohamed, 'AES IP for Hybrid Cryptosystem RSA-AES', 12th International Multi-Conference on Systems, Signals & Devices, Pp. 1-6, 2015.

[7] Ankit Dhananjay Khedlekar, Tejas Mahadev Shelke, Shraddha Walhekar and Nikhita Nerkar, 'Smart Secure System Using Parallel AES', International Journal of Advance Research And Innovative Ideas In Education, Vol. 3, No. 3, 2017, Pp. 1517-1522.

[8] Ashwini R. Tonde and Akshay P. Dhande, 'Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA', International Journal of Current Engineering and Technology, Vol.4, No.2, Pp. 1048-1050, April 2014.

[9] Yang.J, Ding.J, Li.N and Guo.Y,'FPGA-based design and implementation of reduced AES algorithm', IEEE Inter.Conf. Chal Envir Sci Com Engin(CESCE).,Vol.02, No.5-6, Pp.67 70, June 2010.

[10]

[11] Mateur.K, Alareqi.M, and Elgouri.R, 'Design and hardware implementation of AES algorithm on FPGA ', WITS, The international conference on Wireless Technologies embedded and intelligent systems ENSA of kenitra, April 2016.

[12] Prachi bhalero v et el, 'Hardware implementation of cryptosystem by AES algorithm using FPGA' journal of Computer Science and information Technology IJCSMC, vol 6, issue 5, may 2017.

[13] Pravin Ghewari B, Jayamaka Patil K and Amit Chougule B 'Efficient hardware design and implementation of AES cryptosystem', International Journal of Engineering Science and Technology, Vol. 2(3), 2010.

[14] Shihai zhu 'Hardware implementation of AES encryption and decryption system based on FPGA' The open cybernetics & Systemic journal, 2015.