Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.10 n.48, p. 105-114. July/August., 2024.
DOI: https://doi.org/10.5935/jetia.v10i48.915

RESEARCH ARTICLE OPEN ACCESS

AREA AND SPEED EFFICIENT FPGA DESIGN OF S-BOX AES-256 GALOIS FIELD APPROCH BASED ON LOGIC

K Janshi Lakshmi[1], G Sreenivasulu [2]

[1,2]Research Scholar, Department of Electronics and Communication Engineering,
Sri Venkateswara University Collegeo f Engineering,
SriVenkateswara University, Tirupati, Andhra Pradesh, India.

[1] https://orcid.org/0000-0001-9543-9478, [2]https://orcid.org/0009-0008-9025-177

Email: * jansikaramala@gmail.com, gunapatieee@rediffmail.com.

ARTICLE INFO

Article History
Received: October 03th, 2023
Revised: July 08th, 2024
Accepted: Month 15th, 2024
Published: Month 30th, 2024

Keywords:
AES (Advanced Encryption Standard),
S-box Optimization,
Galois Field Arithmetic,
FPGA (Field-Programmable Gate Array),
Delay Reduction.

ABSTRACT

This paper Provides Compared S-box Galois Field Approach Based on LUT and Logic Gates for AES in terms of decreased chip size and decreased delay, which enhances performance. Data security is a fundamental requirement in the digital age. Modern cryptography encryption techniques are essential for creating secure communication. The Advanced Encryption Satandard (AES) is widely regarded as the cryptography field's strongest encryption technique. The Operate Three Stage Pipeline process to reduce delay of S-box AES-256 using logic gates Galios Field approch. So,accordingly increse speed. Additionally, results of the suggested and existing approaches were compared. The proposed approch simulated and sythesised with Virtex-5 FPGA device along with design in verilog code in xilinx 14.7 software.

I. INTRODUCTION

Numerous Platforms One Family. Four new platforms are available from the Virtex--5 family of FPGA, each of which offers an optimised mix of embedded processing, signal processing, high-performance logic, and serial connection. The Virtex line of FPGAs are built using Configurable Logic Blocks (CLBs), where each CLB is comparable to a number of ASIC gates. Numerous slices, which vary in design between Virtex families and make up each CLB, are used. Military-grade specialised processor is used by Virtex. The product's primary market is high-latency broadband applications, which necessitate processing a lot of data with little latency.

Digital data is protected using cryptography. Changing plaintext into illegible text and conversely is mention to as cryptosystem. It is an approach of transferring and saving data in a particular way so that just those who are meant to access and analyse it may do so. It is focuses on converting data into formats that can't be understood by unauthorised users. A message the fact that has been encrypted along with which the letters have been converted to other characters is an instance of basic cryptogsystem see the Fig.1. The main cryptography principles are Data Confidentiality, Data Integrity, Authentication and Non-repudiationin modern day.

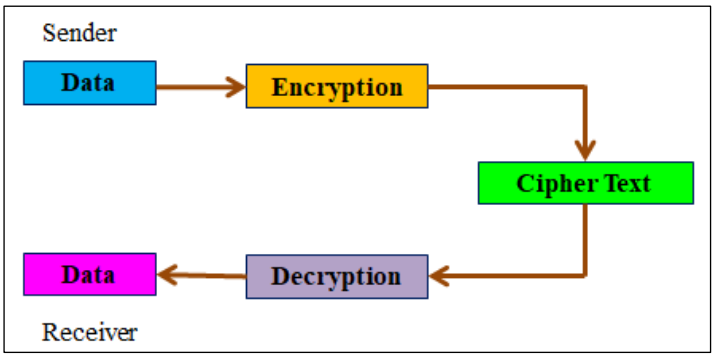

Figure 1: Cryptography Process.
Source: Authors, (2024).

The following elements make up a fundamental cryptosystem-Plaintext is information must be kept secure,The encryption algorithm is a mathematical formula that receives, plaintext as input and outputs ciphertext. This is the plaintext's encrypted or unintelligible counterpart, or ciphertext. Cryptography can be broken down into 3 different groups: secret key encryption, public key encryption, and hash functions. Encryption by employing a Symmetric Key, Asymmetric Key Encryption.Symmetric Key Encryption: from the Fig. 2. For the encryption and decryption have given same secret key that is secret key 1 and secret key 1. But Asymmetric Key Encryption for the encryption and decryption have given different secret key that is secret key 1 and secret key 2 see the Fig. 3.

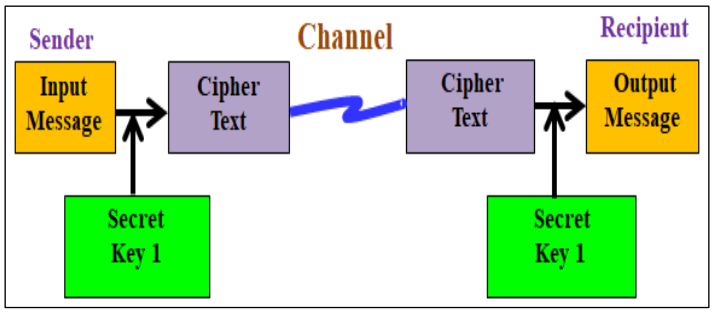

Figure 2: Symmetric Key Encryption.
Source: Authors, (2024).

Messages can be securely encrypted and decrypted utilising asymmetric cryptography, referred to as public-key cryptography, effectively prevents them towards unauthorised access and utilisation. A public key and a private key that are linked jointly are utilised. A single key is used for both encryption and decryption in symmetric encryption, which speeds up the encryption process. Asymmetric encryption uses two separate keys, one of which is related to the other by a challenging mathematical process, hence the encryption process is slower when using this method. Network security authentication and digital signature applications: Date and time stamping, electronic currency, The encryption and decryption of email, encrypting WhatsApp, The encryption of Instagram, Identification with a SIM card.

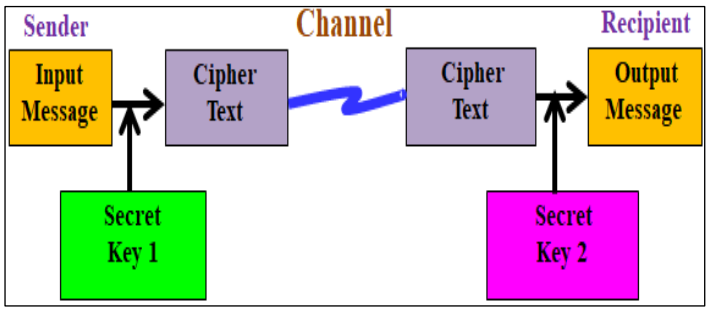

Figure 3: Asymmetric Key Encryption.
Source: Authors, (2024).

The most famous and commonly employed symmetric encryption technique accessible today is AES. It is at least six times speedier to discover than triple DES. A replacement was required while the key size of DES was inadequate. As processing power spiked it became considered vulnerable to an exhaustive key search assault. Triple Data Encryption Standard was supposed to solve this disadvantage, but it was proven to be sluggish. In both hardware and software, AES is a widely used and supported encryption process. No genuine assaults employing crypt analysis has yet been carried out versus AES. The AES also has built-in flexibility for key length, which offers some "future-proofing" against improvements in the ability for thorough key searches. AES security is only made sure, though, if it is deployed properly and adequate key management is employed, just like DES. Federal government departments, non-government organisations, commercial firms, and organisations regularly use AES encryption to secure sensitive data. The AES algorithm is widely utilised in a wide range of applications, including file encryption, SSL/TLS, processor security, wireless security.

II. LITERATURE REVIEW

You-Tun Teng, Wen-Long Chın, et.al.[1] are represented the advanced encryption standard's (AES) SubBytes and Inverse SubBytes procedures implemented on hardware. To achieve this, the SubBytes (and inverse SubBytes) transformation's S-box (and inverse S-box) building blocks are all optimised using the composite field arithmetic (CFA).

Nandan And Gowrı Shankar Rao [2] are presented about Implementing the inverse of the improved affine transform is the main objective in this case. The basic zigbee S-Box approach and the Rijndael method are compared in terms of a variety of factors. This is due to the use of extra registers for the various stages.

Christian Equihua, Esteban Anides [3] are shown The Galois Field Multiplier, which is it has been determined that the operation, that is involved in the Mix-Columns and Inverse Mix-Columns transformations, is the most demanding in terms of processing performance and area consumption.

Rei Ueno and Kohei Matsuda [4] were represented of Critical components have been combined with fewer additional selectors utilizing Data path's innovative operation reordering and register-re timing algorithms.

Poonam Jindal, Aryan Kaushik [5] were research on Simulation of the Advanced Encryption Standard utilizing double 7th series capital to examine cost and performance contrasts.

Yashvir Singh Chauhan, T.N. Sasamal [6] were specified about Generates multiple s-boxes for different rounds that is s-box changes with respect to key and hence increasing the security of cipher

Cory Davis, Alekhya Muthineni [7] were presented Low power AES design based on a novel implementation for the Shift Rows operation. Designed and implemented using 45nm and 90nm technology nodes

Shivakumar V Gaded, Abhay Deshpande [8] were represented by Design an S-box that utilizes Composite Field arithmetic to significantly minimize the size relative to FPGA slices and the gate latency as well as the combinational path delay.

Santhosh Kumar R, Shashidhar R [9] were research on Pipeline design for AES algorithm to increase the Performance of Hardware

Sri perumbuduru Srilaya, Sirisha Velampalli [10] specified about Throughput of encryption and throughput of decryption of DES is more compared to AES.

III. OVERVIEW OF AES ALGORITHM

III.1 AES-256 ENCRYPTION

Advanced Encryption Standard (AES) has input Data is 128bits, Secret key is 128bits, 192bits and 256bits. Here using 256bit key algorithm. In 256bit key algorithm has total 14 rounds perform. For encryption has 14 rounds as well as for AES decryption also has 14 rounds. İt shows in Fig.4,5. Substituting bytes (SubBytes): The design includes a fixed table (S-box) that is hunted up in order to replace the 16 input bytes. The outcome is represented as a matrix with four rows and four columns. The four rows of the matrix are all shifted to the left. Any entries that 'fall off' are reinserted on the row's right side. During shift, the subsequent methods apply: The top row remains in place, The second row has been moved one (byte) position left, the third row has been moved two spaces left, whereas the fourth row has been moved three spaces left. The end result is a new matrix with the same 16 bytes but with various locations. MixColumns: Each column of four bytes is now treated to an unique mathematical operation. In order to make use of this function, four bytes from one column need to be supplied, and four completely novel bytes should be output in its place. The most recent matrix has 16 more bytes and is a fresh one. This step does not appear in the final round, it should be highlighted. Addroundkey: The 16 bytes of the matrix, which are now regarded as 128 bits, are XORed with the 128 bits of the round key. If this is the final attempt, the ciphertext will be generated. Otherwise, the outcome's 128 bits is transformed into 16 bytes, and the process is restarted. In encryption 1st round perform 5 operations i.e, Preround, shiftrow,addround key, subbyte and mixcolumn. From 2nd round to 13th round perform 4 operations i.e, subbyte, shiftrow,mixcolumn and addround key. For 14th round it perform 3 operations only i.e, subbyte,shiftrow, addround key operationsee the Fig.4.Finally, the output is 128bits cipher key generates.

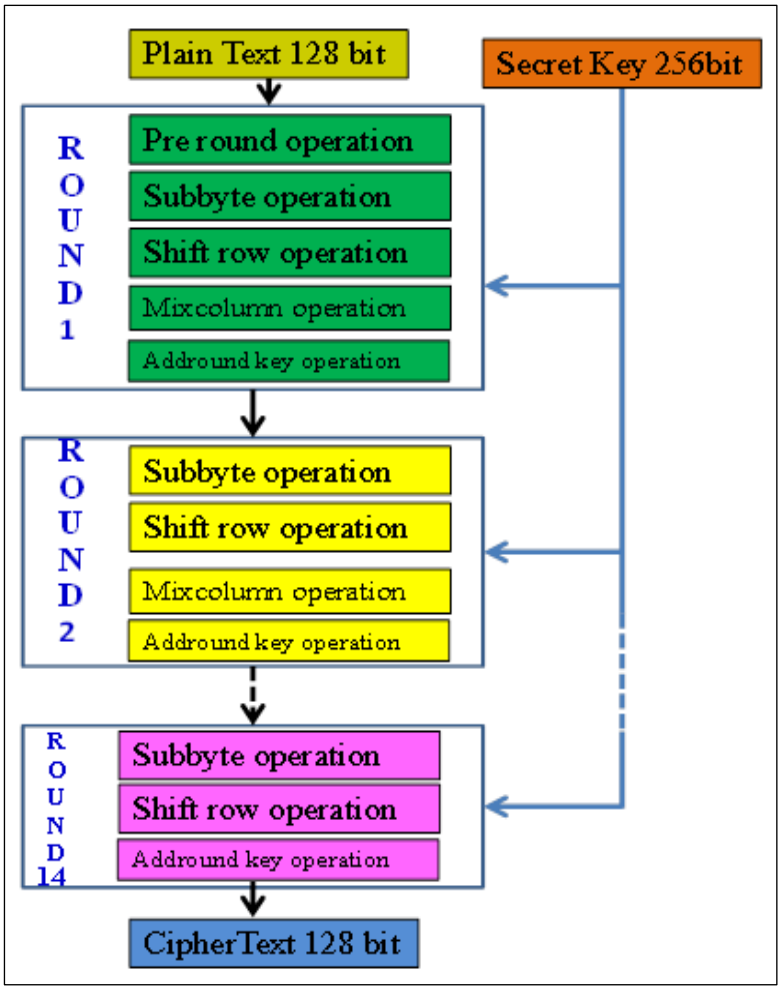

Figure 4: AES-256 Encryption.
Source: Authors, (2024).

III.2 AES-256 DECRYPTION

AES-256 decryption has input is cipher key 128bits, cipher key means combination of input data and secret key. Secret key is 128bits, 192bits and 256bits. Here using 256bit key algorithm. In 256bit key algorithm has total 14 rounds perform. For decryption has 14 rounds İt shows in Fig. 5. The reversal of the encryption process is used in the decryption of an AES cipher text. The four procedures are carried out each round in reverse sequence. Inverse Byte replacement, Inverse Byte Mix, Inverse Shift, and Inverse Add round key. Despite having extremely close relationships, the encryption and decryption algorithms must be implemented independently since each round's sub-processes function backwards. In decryption 1st round perform 5 operations i.e, Preround, Invshiftrow,addround key, Invsubbyte and Invmixcolumn. From 2nd round to 13th round perform 4 operations i.e, Invsubbyte, Invshiftrow, Invmixcolumn and addround key. For 14th round it perform 3 operations only i.e, Invsubbyte, Invshiftrow, addround key operationsee the Fig.5.Finally, the output is 128bits original data or input data generates.

IV. PRAPOSED METHODOLOGY

IV.1 S-BOX USING LOGIC GATES

The Example for Computing the Sub Byte Operation with Logic Gates in S-Box is show in Fig. 8, the dissemination of the data input of $(04)_{16}$ into a composite field based Substitute Box. The precomputed values were formerly kept in a ROM-based lookup table, which was one of the most popular and simple implementations of the StandardBox for the SubByte operation (LUT). In this approach, the input byte would be connected to the ROM's address bus and all 256 values would be kept in a ROM. The multiplicative inversion of the supplied data is the first step. The four bit numbers outside of the logical blocks show the new values that are applied to the high and low nibbles. Since the results containing for GF (2^4) multiplication and multiplicative inverses are provided, the example can be completed by hand. Following the multiplicative inversion module's inverse isomorphic mapping operation, the To create the S-Box replacement value for the input, an affine transformation is applied to the multiplicative inverse. of $(CB)_{16}$. Finally the output is $(F2)_{16}$, which is consistent with the S-Box table given. Combinational logic is a more sophisticated method of implementing the S-Box, and this kind of S-Box has the benefit of having low space occupancy or less area occupied of system [11], [12]. Due to this smaller size, Computing the Sub Byte Operation using Logic Gates was implemented in S-Box in instead of LUT. Moreover, in order to take advantage of the FPGA's reduced path delay, three pipeline stages S-Box were developed employing logic gates. It enhances optimisation and performance. These two-design shown in Fig.8 and 10.

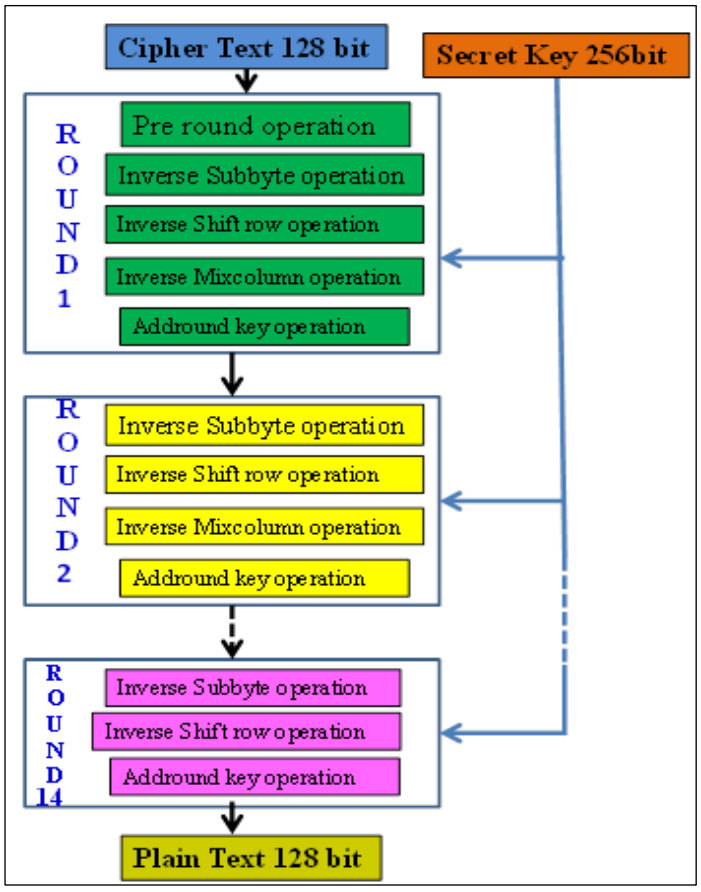

Figure 5: AES-256 Decryption.
Source: Authors, (2024).

IV.2 ISOMORPHIC AND INVERSE ISOMORPHIC MAPPING

The logical XOR operation can be used to translate the matrix multiplication. Below is a diagram of the matrices' logical form.

$$\delta \times j = \begin{pmatrix} 1&0&1&0&0&0&0&0\\ 1&1&0&1&1&1&1&0\\ 1&0&1&0&1&1&0&0\\ 1&0&1&0&1&1&1&0\\ 1&1&0&0&0&1&1&0\\ 1&0&0&1&1&1&1&0\\ 0&1&0&1&0&0&1&0\\ 0&1&0&0&0&0&1&1 \end{pmatrix} \times \begin{pmatrix} j_7\\ j_6\\ j_5\\ j_4\\ j_3\\ j_2\\ j_1\\ j_0 \end{pmatrix} \quad \delta^{-1} \times j = \begin{pmatrix} 1&0&1&0&0&0&0&0\\ 1&1&0&1&1&1&1&0\\ 1&0&1&0&1&1&0&0\\ 1&0&1&0&1&1&1&0\\ 1&1&0&0&0&1&1&0\\ 1&0&0&1&1&1&1&0\\ 0&1&0&1&0&0&1&0\\ 0&1&0&0&0&0&1&1 \end{pmatrix} \times \begin{pmatrix} j_7\\ j_6\\ j_5\\ j_4\\ j_3\\ j_2\\ j_1\\ j_0 \end{pmatrix}$$

(1)

The above matrices simplified then becomes

$$\delta \times j = \begin{pmatrix} j_7 \oplus j_5\\ j_7 \oplus j_6 \oplus j_4 \oplus j_3 \oplus j_2 \oplus j_1\\ j_7 \oplus j_5 \oplus j_3 \oplus j_2\\ j_7 \oplus j_5 \oplus j_3 \oplus j_2 \oplus j_1\\ j_7 \oplus j_6 \oplus j_2 \oplus j_1\\ j_7 \oplus j_4 \oplus j_3 \oplus j_2 \oplus j_1\\ j_6 \oplus j_4 \oplus j_1\\ j_6 \oplus j_1 \oplus j_0 \end{pmatrix} \quad \delta^{-1} \times j = \begin{pmatrix} j_7 \oplus j_6 \oplus j_5 \oplus j_1\\ j_6 \oplus j_2\\ j_6 \oplus j_5 \oplus j_1\\ j_6 \oplus j_5 \oplus j_4 \oplus j_2 \oplus j_1\\ j_5 \oplus j_4 \oplus j_3 \oplus j_2 \oplus j_1\\ j_7 \oplus j_4 \oplus j_3 \oplus j_2 \oplus j_1\\ j_5 \oplus j_4\\ j_6 \oplus j_5 \oplus j_4 \oplus j_2 \oplus j_0 \end{pmatrix}$$

(2)

lois Field corresponds to the addition of two elements.

IV.3 ADDITION OF GF(24)

The basic bit wise XOR operation between two elements in a Galois Field corresponds to the addition of two elements.4.4 Squaring of GF (2^4)

IV.4 SQUARING OF GF(24)

Let $b = j^2$ whereb and j are elements of GF(2^4), represented by the binary number of $\{b_3\ b_2\ b_1 b_0\}_2$ and $\{j_3 j_2\ j_1 j_0\}_2$ respectively.It shows in Fig. 6.

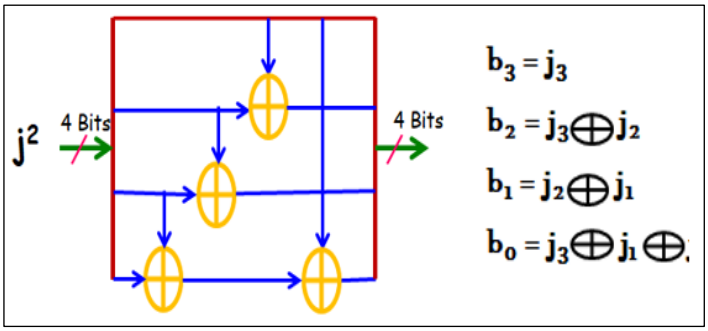

Figure 6: Diagram of squaring GF (2^4).
Source: Authors, (2024).

IV.5 MULTIPLYING WITH Λ

λ is constant, Let $b = j\lambda$, where $= \{b3b2b1b0\}_2$, $j = \{j3j2j1j0\}_2$ and $\lambda = \{1100\}_2$ are elements of GF(2^4).It shows in Fig. 7.

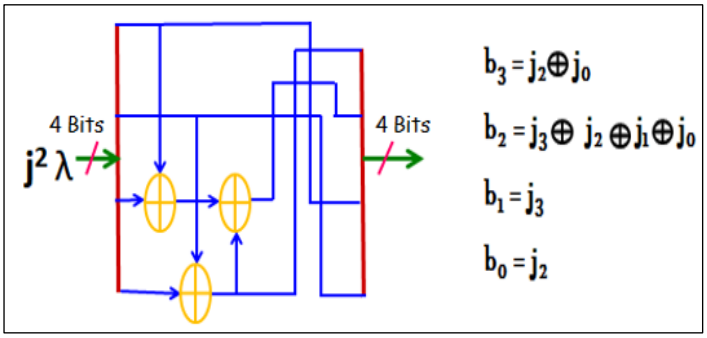

Figure 7: Diagram of multiplying gf (2^4).
Source: Authors, (2024).

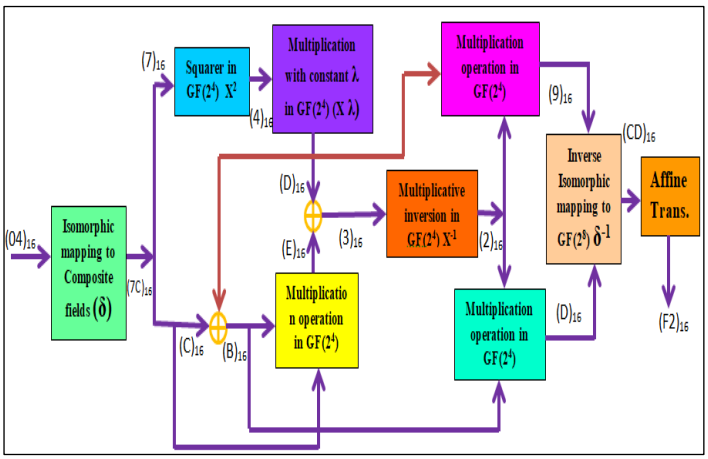

Figure 8: The Example for computing the sub byte operation with logic gates in S-Box.
Source: Authors, (2024).

IV.6 MULTIPLYING WITH Λ

λ is constant, Let $b = j\lambda$, where $= \{b3b2b1b0\}_2$, $j = \{j3j2j1j0\}_2$ and $\lambda = \{1100\}_2$ are elements of GF(2^4).It shows in Fig. 9.

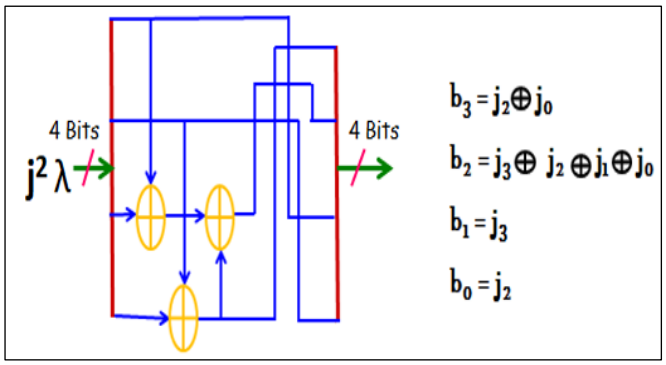

Figure 9: Diagram of multiplying GF (2^4).
Source: Authors, (2024).

V. STANDARD BOX USING THREE STAGE PIPELINING

The concept of pipe-lining is based on sending water through a pipe continuously rather than waiting for it to empty. It therefore causes an increase in speed. Pipelining is a method of implementation where several instructions are executed simultaneously. There are many phases in the computer process. Each step simultaneously completes a portion of an instruction.Instructions enter at one end, go through the stages, and emerge at the other end of the pipe that connects them. Efficient systems include three pipeline stages for the byte substitution phase, as shown in Fig. 10 The pipe registers for the three-stage S-Box are shown here in dotted lines. So reduces delay of system rather than increases speed of system.

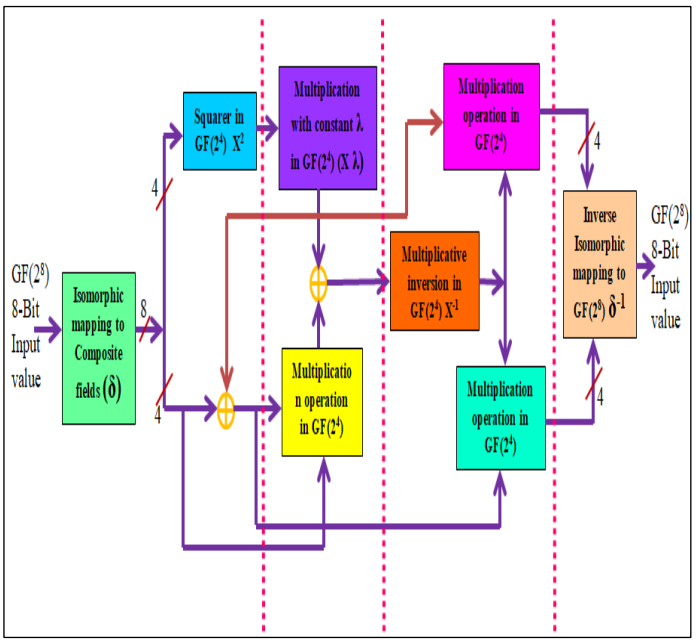

Figure 10: The diagram of three pipeline stages S-Box.
Source: Authors, (2024).

VI. AES-256 RTL SCHEMATIC DIAGRAM

AES - 256 algorithm Encryption and Decryption schematic diagram is depicted in see Fig. 11. Here taking clk, rst, enc_dec are input active low it act as Encryption. Later clk, rst, enc_dec are input active high it act as Decryption. And aesin is 128 bit input, keyin is 256bit, aesout is 128bit.

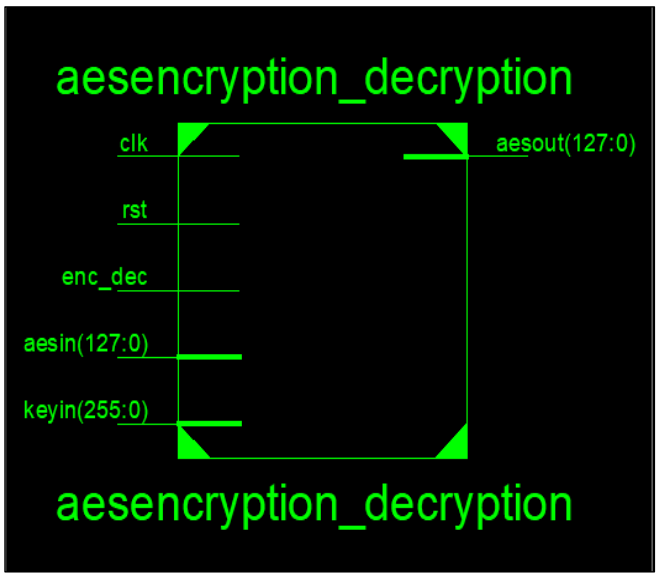

Figure 11: AES rtl schematic diagram.
Source: Authors, (2024).

VII. SIMULATION RESULTS

Table 1: AES-256 input, cıpher key and output data

AES-256 Encryption Data	
Hexadecimal Code	
aesin	4B2E4A414E534849204C414B53484D49
keyin	5352492056454E4B41544553574152412055 4E49564552534954592C20545054
aesout (Cipher Key)	8afc5aedb35ddfcae4balscf06a673c8
ASCII Code	
aesin	K.JANSHI LAKSHMI
keyin	SRI VENKATESWARA UNIVERSITY, TPT
aesout (Cipher Key)	8afc5aedb35ddfcaedbal5cf06a673c8 (Hex Code)
AES-256 Decryption Data	
Hexadecimal Code	
aesin (Cipher Key)	8afc5aedb35ddfcaedbalscf06a673c8
keyin	5352492056454E4B41544553574152412055 4E49564552534954592C20545054
Aesout	4B2E4A414E534845204C414B53484D49
ASCII Code	
aesin (Cipher Key)	8afc5aedb35ddfcaedbal5cf06a673c8 (Hex Code)
keyin	SRI VENKATESWARA UNIVERSITY, TPT
Aesout	K.JANSHI LAKSHMI

Source: Authors, (2024).

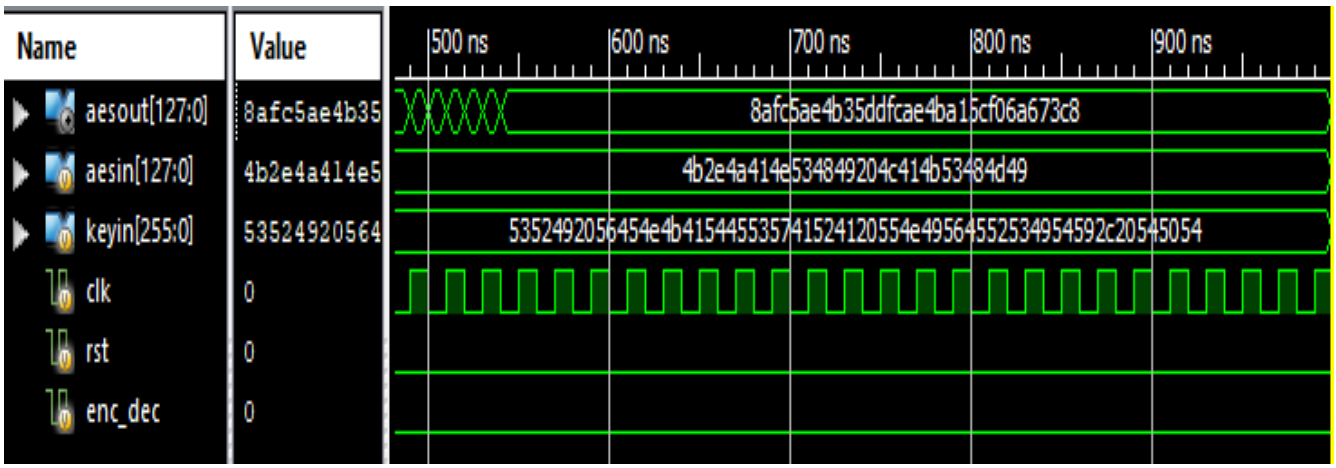

Figure 12: Aes-256algorithm Encryption (Hex Code).
Source: Authors, (2024).

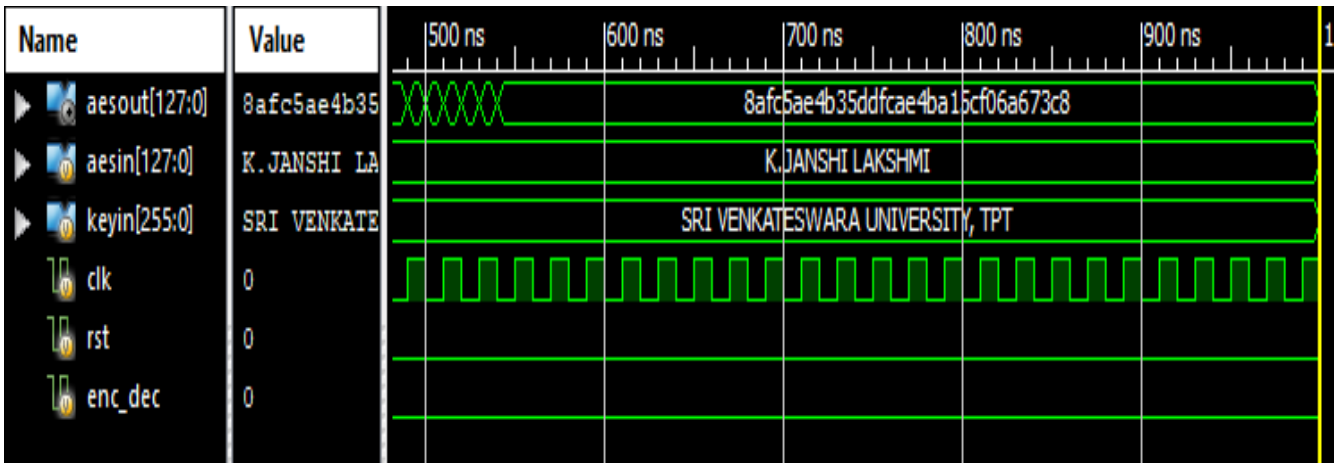

Figure 13: Aes-256 Algorithm Encryption (Ascii Code)
Source: Authors, (2024).

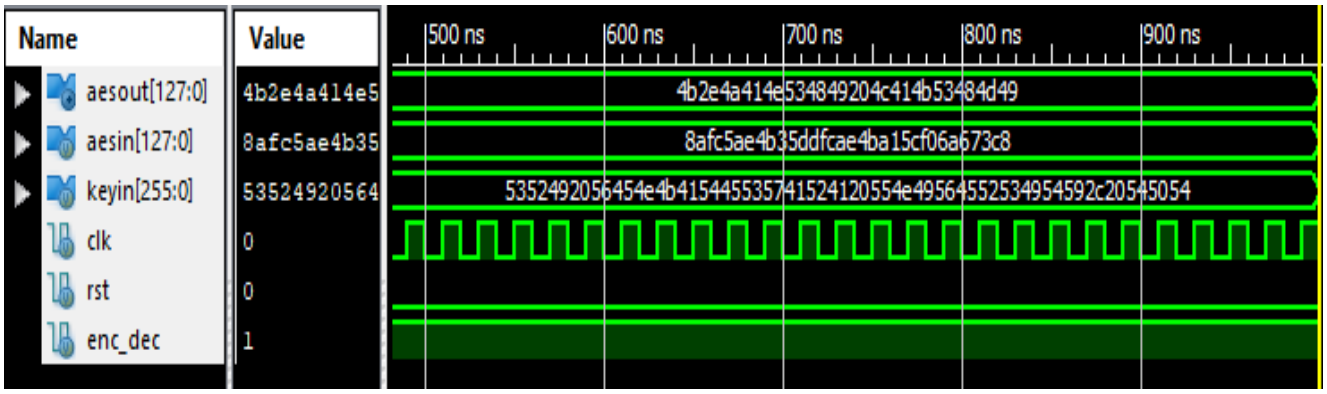

Figure 14: Aes-256 Algorithm Decryption (Hex Code).
Source: Authors, (2024).

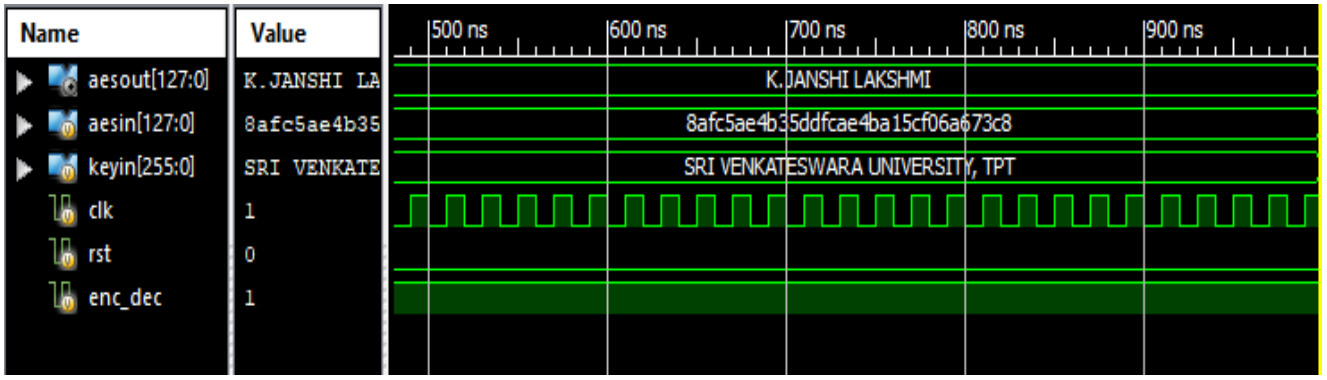

Figure 15: Aes-256 Algorithm Decryption (Ascii Code).
Source: Authors, (2024).

All of these above waveforms were implemented in Verilog - testbench and later simulated in the Xilinx ISE design suite 14.7 tool. AES-256 algorithm Encryption and Decryption simulation results Hexadecimal and ASCII code format as shown in figures are from Fig. 12. to Fig. 15. When clk, rst, enc_dec are input active low it act as Encryption. Input is aesin 128-bit, key is keyin 256bit. It generates aesout also called cipher key 128bit. This cipher key is input of decryption and give key same as encryption keyin 256bit. When clk, rst, enc_dec are input active high it act as Decryption. Later it generates original data that is aesin 128 bit. All inputs and outputs shown in above Table in form of HEX code and ASCII code forms (see Table 1.)

VIII. SYNTHESIS RESULTS

VIII.1 COMPARISON OF SYNTHESIS RESULTS OF LUT BASED AES-256 WITH AES-256 USING LOGIC GATES IN TERMS OF SIZE OR AREA

The FPGA Device Used and synthesis with Virtex- 5 ML510 Evaluation platform. After synthesis of place and route Here three parameters variations occurs in Table 2 indicated red color stars and Table 3indicated green color stars. That parameters are i) Number of LUTs ii) Number of occupied slices and iii) Average fanout of non-clock nets. These parameters reduced in table 4, because this is using logic gate using for AES- 256 algorithm. So reduces above parameters that means area reduced AES-256 using Logic Gates than LUT based AES-256. The chart representation shown in Fig. 16.

Table 2: LUT Based AES-256 device utilization summary.

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice LUTs	★ 40,791	81,920	49%	
Number used as logic	40,791	81,920	49%	
Number using O6 output only	40,791			
Number of occupied Slices	★ 15,034	20,480	73%	
Number of LUT Flip Flop pairs used	40,791			
Number with an unused Flip Flop	40,791	40,791	100%	
Number with an unused LUT	0	40,791	0%	
Number of fully used LUT-FF pairs	0	40,791	0%	
Number of slice register sites lost to control set restrictions	0	81,920	0%	
Number of bonded IOBs	513	840	61%	
Average Fanout of Non-Clock Nets	★ 7.69			

Source: Authors, (2024).

Table 3: AES-256 using logic gates device utilization summary.

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice LUTs	34,671	81,920	42%	
Number used as logic	34,671	81,920	42%	
Number using O6 output only	34,671			
Number of occupied Slices	14,808	20,480	72%	
Number of LUT Flip Flop pairs used	34,671			
Number with an unused Flip Flop	34,671	34,671	100%	
Number with an unused LUT	0	34,671	0%	
Number of fully used LUT-FF pairs	0	34,671	0%	
Number of slice register sites lost to control set restrictions	0	81,920	0%	
Number of bonded IOBs	513	840	61%	
Average Fanout of Non-Clock Nets	5.59			

Source: Authors, (2024)

Table 4: Comparison of AES-256 using LUTs (Existing Methodology) with using Logic gates galois field approach (Proposed Methodology) in terms of Size or Area.

Slice Logic Utilization	**LUTs**	**Logic gates**
Number of Slice LUTs	40,791	34,671
Number of Occupied Slices	15,034	14,808
Number of LUT Flip-flop Pair used	40,791	34671
Number of IOBs	513	513
Average Fan-out of Non-Clock Nets	7.69	5.59
Peak Memory usage	5157 MB	5171 MB

Source: Authors, (2024).

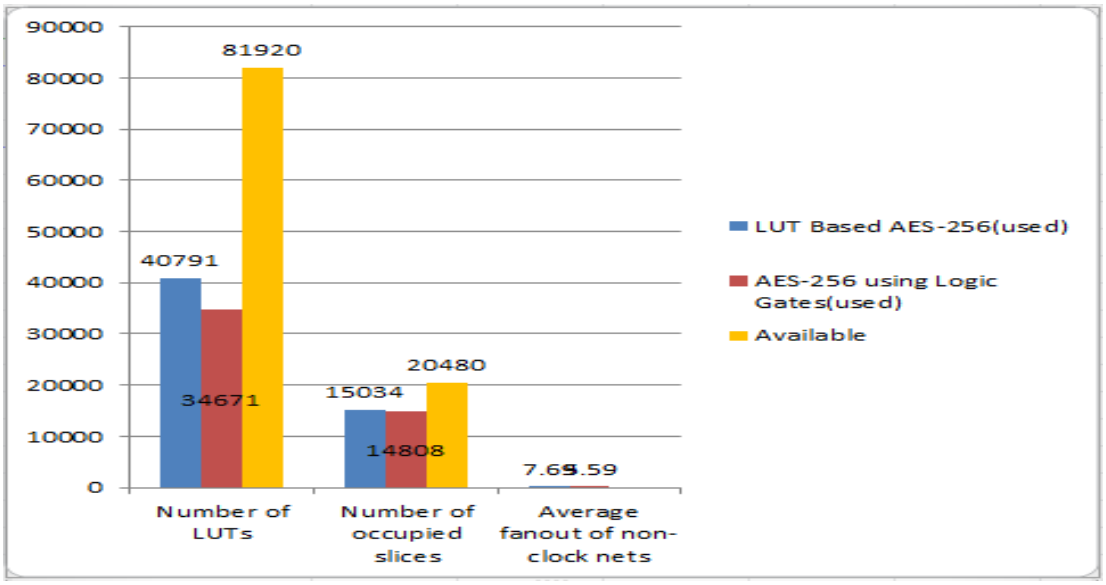

Figure. 16: Chart representation of comparing parameters of LUT and Logic Gates using AES-256.
Source: Authors, (2024)

.

Comparison of AES-256 using LUTs (Existing Methodology) with using Logic gates Galois field approach (Proposed Methodology) shows in Table 5. AES using logic gates Galois field approach is reduced area comparing Using LUT. And also it shows Fig. 16 in graphical diagram format i.e. chart representation. In Chart, Blu colour represents LUT Based AES-256 utilization, Red colour indicates AES -256 Using Logic Gates Galois field approach utilization. Yellow colour specify available in FPGA device.

VIII.2 COMPARISON OF SYNTHESIS RESULTS OF AES-256 USING LOGIC GATES WITH OUT PIPE-LINING WITH AES-256 USING LOGIC GATES WITH PIPE-LINING IN TERMS OF DELAY

```
Timing Summary:
---------------
Speed Grade: -2

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 299.782ns
```

Figure 17: AES-256 using logic gates with out pipe-lining.
Source: Authors, (2024).

```
Timing Summary:
---------------
Speed Grade: -2

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 239.227ns
```

Figure 18: AES-256 using Logic gates with pipe-lining.
Source: Authors, (2024).

The FPGA Device Used and synthesis with Virtex -5 ML510 Evaluation platform. After synthesis of place and route then generate timing report. In timing summary path delay generates. AES-256 using Logic gates with out pipe-lining delay is 299.782ns its represent red color in Fig. 17. Delay high means speed also high. For reduce delay using AES-256 using Logic gates with three stage pipe-lining. Used pipe-lining the path Delay is 239.227ns it represents green color in Fig. 18. This system reduced path delay is 60.555ns. So this system perform fastly compared towith out pipeline. Delay Reduced and speed increased AES- 256 using Gates with pipeline than AES- 256 using Gates without pipeline. The chart representation shown in Fig. 19.Comparison of path delay AES-256 using Logic gates without pipe-lining with Logic gates with pipe-lining shows in Table 5.

Table 5: Comparison Of Path Delay AES-256 Usinglogic Gates Without Pipe-Lining with Logic Gates with Pipe-Lining.

Timing summary	AES-256 using Logic gates without pipe-lining	AES-256 using Logic gates with pipe-lining
Path Delay (ns)	299.782	239.227

Source: Authors, (2024).

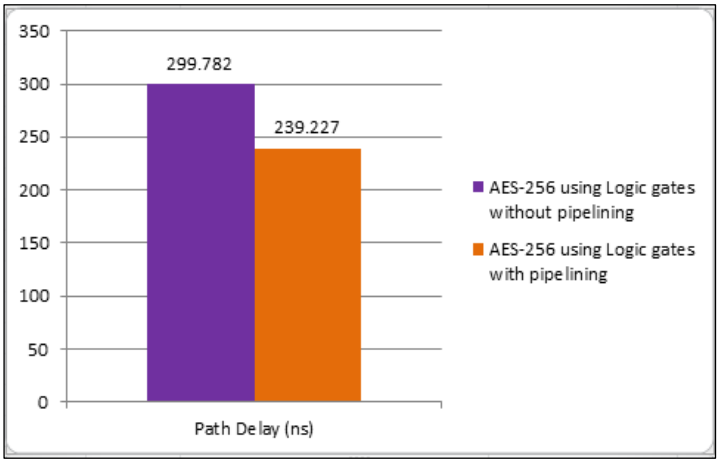

Figure 19: Chart representation of comparing path delay of aes-256 using logic gates with and without pipe-lining.
Source: Authors, (2024).

IX. CONCLUSION

This paper represented Area or size efficient of S-box AES-256 using Logic Gates Galois Field approach than S-box LUT. Discussed, compared and represent about Reduced number of LUTs and Number of Slices. So defaultly decreased area and it occupies less space in chip. later Operate Three Stage Pipeline process to reduce Delay or time of S-box AES-256 using logic gates Galois Field approach. Discussed, compared and represent about with and with out pipe-lining design. So,accordingly increase speed, decreases path delay and improve performance. The proposed approach simulated and synthesized with Virtex-5 ML510 Evaluation platform FPGA device along with design in verilog code in xilinx 14.7 software.

X. AUTHOR'S CONTRIBUTION

Conceptualization: K Janshi Lakshmi, G Sreenivasulu
Methodology: K Janshi Lakshmi, G Sreenivasulu
Investigation: K Janshi Lakshmi, G Sreenivasulu
Discussion of results: K Janshi Lakshmi, G Sreenivasulu
Writing – Original Draft: K Janshi Lakshmi
Writing – Review and Editing: K Janshi Lakshmi, G Sreenivasulu
Resources: Author Two.
Supervision: K Janshi Lakshmi, G Sreenivasulu
Approval of the final text: K Janshi Lakshmi, G Sreenivasulu

XI. REFERENCES

[1] You-Tun Teng, Wen-Long Chın, et.al. "VLSI Architecture of S-Box With High Area Efficiency Based on Composite Field Arithmetic", IEEE Access, Volume 10, 2022.

[2] Nandan, Gowrı Shankar Rao, "Low-power and area-efficient design of AES S-Box using enhanced transformation method for security application", International Journal Communication System, 2020. wileyonlinelibrary.com/journal/dac © 2020.

[3] Christian Equihua, Esteban Anides,"A low-cost and highly compact FPGA-based encryption /decryption architecture for AES algorithm", 2021, IEEE Latin America Transactions, Volume: 19, Issue: 9.

[4] Rei Ueno and Kohei Matsuda, "High Throughput/Gate AES Hardware Architectures Based on Datapath Compression", IEEE Transaction on computers,2020, Volume: 69, Issue: 4.

[5] Poonam Jindal, Aryan Kaushik, "Design and Implementation of Advanced Encryption Standard Algorithm on 7th Series Field Programmable Gate Array"

2020, 7th International Conference on Smart Structures and Systems (ICSSS), IEEE conference paper.

[6] Yashvir Singh Chauhan, T.N. Sasamal, "Enhancing Security of AES Using Key Dependent Dynamic S-box", 2019, International Conference on Communication and Electronics Systems (ICCES), IEEE conference paper.

[7] Cory Davis, AlekhyaMuthineni, "Low-Power Advanced Encryption Standard for Implantable Cardiac Devices", 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE conference paper.

[8] Shivakumar V Gaded, Abhay Deshpande, "Composite Field Arithmetic Based S-Box For AES Algorithm"2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE conference paper.

[9] Santhosh Kumar R, Shashidhar R, "Design of High Speed AES System for Efficient Data Encryption and Decryption System using FPGA", 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), IEEE conference paper.

[10] Sriperumbuduru Srilaya, Sirisha Velampalli, "Performance Evaluation for DES and AES Algorithms- An Comprehensive Overview" 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE conference paper.

[11] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[12] Marko Mali,Franc Novak and Anton Biasizzo, "Hardware Implementation of AES Algorithm", Journal of Electrical Engineering,Vol.56,No.9-10, 2005, 265-269