iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** V **Month of publication:** May 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Design Of AES-128 Bit For Enhanced And Sophisticated Security Application

Ms. Monika Badki[#1], Prof. (Mrs.) A.H. Charkhawala[*2]
[#]*Department Of Electronica and Telecommunication, Priyadarshini College Of Engineering, Nagpur University*

Abstract— This paper describes a detailed description about the two main processes includes in Advanced Encryption Standard (AES) which are encryption process and decryption process. This includes the working of these two processes using AES algorithm for 128 bit. AES relies on the Rijndael cipher technology which is simply an algorithm that create the encoding and reversing decoding.
Keywords— Aavanced Encryption Standard (AES), Encryption , Decryption, Rijndael Algorithm, Cipher.

I. INTRODUCTION

The Advanced coding customary (Rijndael Block Cipher) became the new U.S.A. Federal informatics customary on November 26, 2001[1] in order to replace the Data Encryption Standard (DES) that was used for over 20 years as a standard key block cipher for FIPS. After that, several hardware implementations for FPGA and ASIC have been introduced [2], [3], [4].
The current trend in client merchandise is toward an increasing would like of secure protocols and algorithms, which can be often implemented in hardware on reconfigurable platforms, due to their advantageous benefit/cost ratio for low volumes. The Advanced Encryption Standard (AES) [1] is the de facto standard and many FPGA-based implementations have been proposed. Security is the most important part in electronic communication system, where more randomization in secret keys will increase the safety in addition as complexity of the cryptography algorithms. Data security is an essential objective for the military and diplomatic services that have several business uses and applications like electronic banking, electronic mail, internet network service, electronic communication networks etc. As an economical and cost-efficient science algorithmic rule AES [1] algorithmic rule has broad applications, which includes smart cards and telephones, web servers and automated teller machines (ATMs). This paper describes the design-flow of the AES algorithmic rule for each encoding and decoding method.
The AES design are often employed in any application that needs protection of data while transmitting through the communication network, including applications such as electronic commerce transactions, ATM machines, wireless communication, Virtual Private Networks (VPN), and many others. Our AES cores can be used as a part of hardware or hybrid implementation of all major security protocols, such as IPSec, SSL, IEEE 802.11a, and therefore the ATM Forum Security Specification [1].

II. CIPHER DESCRIPTION

An AES, additionally referred to as Rijndael, may be a block cipher adopted as an encoding standard by the United States government that specifies an encoding rule [4]. The AES rule is capable of victimization cryptologic keys of 128, 192, and 256 bits to encipher and decipher data in blocks of 128 bits sequence. In this paper 128 bits secret key is used for 128 bits data block. The input, output and cipher key bit sequences are treated as arrays of bytes that are organized by dividing these sequences into set of eight contiguous bits to create arrays of bytes. The various transformations operate on the intermediate result, referred to as the state, which is the intermediate cipher result [5], [6].

This method of conversion of plaintext into cipher text and once more cipher text into plain text is shown in simple block diagram as in Fig. 1 and Fig. 2.

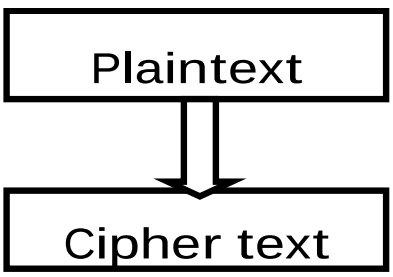


Fig. 3 Block Diagram (Encryption Process)

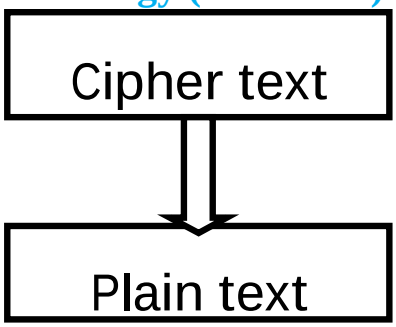


Fig. 2 Block Diagram (Decryption Process)

III. WORKING OF AES ALGORITHM

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text [6]. Decryption of the cipher-text converts the data back into its original form, which is called plain-text.

AES operates on a *4x4* array of bytes (referred to as "state"). The algorithm consists of four different simple operations. These operations are:
Sub Bytes
Shift Rows
Mix Columns
Add Round Key

A. Encryption Process

The Encryption and decryption process consists of a variety of various transformations applied consecutively over the information block bits, in a particular number of iterations, known as rounds. The amount of rounds depends on the size of the key used for the encoding method [6], [8]. For key length of 128 bits, the amount of iteration needed are 10 *(Nr = 10*). As shown in Fig. 1(a), every of the primary *Nr-1* rounds consists of four Transformations: *SubBytes()*, *ShiftRows()*, *MixColumns()* and *AddRoundKey()*.

1) *Sub Byte Transformation:* It is a non-linear substitution of bytes that operates severally on every byte of the state employing a substitution table (*S*-box). This invertible *S*-box is made by initially taking the multiplicative inverse within the finite field GF (2^8) with irreducible polynomial $m(x) = x8 + x4 + x3 + x + 1$. The component {00} is mapped to itself [6]. Then affine transformation is applied (over GF (2)).
2) *Shift Row Transformation:* Cyclically shifts the rows of the state over completely different offsets. The operation is nearly constant within the decoding method aside from the very fact that the shifting offsets have completely different values.
3) *Mix Columns Transformation:* This transformation operates on the state column-by-column, treating every column as a four-term polynomial. The columns area unit thought about as polynomials over GF (2^8) and increased by modulo $x4 + 1$ with a set of fixed polynomial $a(x) = \{03\} x3 + \{01\} x2 + \{01\} x + \{02\}$.
4) *Add Round Key Transformation:* In this transformation, a Round Key is combined with the state by performing a straightforward bitwise XORing. Each Round Key consists of *Nb* words from the key extension. Among those entire *Nb* words each one is combined with the columns of the state [6]. In decoding technique addition of the key is the same as in encoding method.
5) *Key Expansion:* Every round key is a 4-word (128-bit) array composed as a product of the earlier round key, ongoing that changes all rounds, and a series of *S*-Box lookups for each one of 32-bit word of the key. The Key schedule Enlargement generates a complete of $Nb * (Nr + 1)$ words.

Volume 3 Issue V, May 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

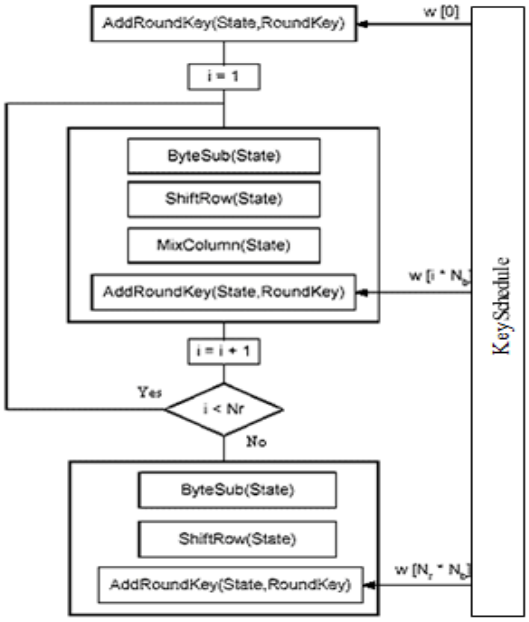


Fig. 3 Encryption Flowchart

B. Decryption Process

For decryption, an equivalent method happens merely in reversal manner, taking the 128-bit block of cipher text and changing it to plaintext by the appliance of the inverse of the four operations. AddRoundKey is that the same for both the encoding and decoding process technique [7]. But the 3 other functions have inverses employed in the decoding process: Inverse SubBytes, Inverse ShiftRows, and Inverse MixColumns. This method is direct inverse of the encryption method. All the transformations applied within the the cryptographic encoding method is reciprocally applied to decoding or decryption process. Therefore the last round values of each the information and key are first round inputs for the decoding technique and follows in decreasing order as shown Fig. 4.

1) *Add Round Key:* AddRoundKey is its own contrary function as a result of the XOR function is its own inverse. The round keys ought to be designated in reverse manner [7]. The outline of remaining transformations will be given as follows.
2) *Inv Shift Row Transformation:* InvShiftRows precisely functions an equivalent as ShiftRows, in the counter direction. The primary row is not shifted, whereas the second, third and fourth rows are shifted right by one, two and three bytes appropriately.
3) *Inv Sub Bytes Transformation:* The InvSubBytes transformation is finished employing a once predetermined substitution table referred to as InvS-box [7]. That InvS-box table contains 256 numbers (from 0 to 255) and their equivalent values.
4) *Inv Mix Column Transformation:* The inverse mix columns transformation is counter measure technique of mix columns transformation.

IV. SOME COMMENTS ON AES

A. A repetitious instead of Feistel cipher

B. Key extended into array of 32-bit words

1) Four words form round key in every round

C. 4 totally different stages are used as shown

D. Has a simple structure

Volume 3 Issue V, May 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

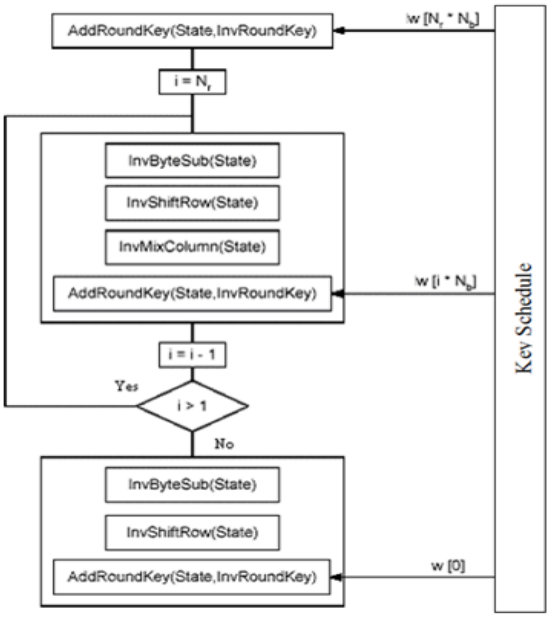


Fig. 4 Decryption Flowchart

E. Only AddRoundKey uses key

F. AddRoundKey a form of Vernam cipher

G. Each stage is easily reversible

H. Decryption uses keys in reverse order

I. Decryption does recover plaintext

J. Final round has only 3 stages

V. APPLICATION AREA OF AES

The AES design can be used in any application that requires protection of data during transmission through the communication network, including applications such as electronic commerce transactions, ATM machines, wireless communication, Virtual Private Networks (VPN), and many others. Our AES cores can be used as a part of hardware or hybrid implementation of all major security protocols, including IPSec, SSL, IEEE 802.11a, and the ATM Forum Security Specification [10], [11].

VI. RESULT

A. Simulation 1:

Following waveform in Fig. 5 showing the simulation result of sub-byte transformation,

1) When reset = 1

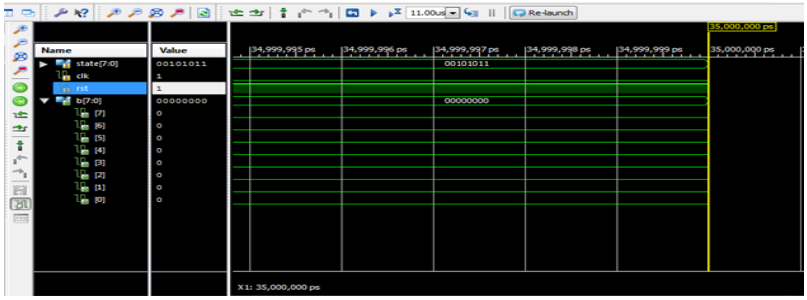


Fig. 5 Simulation Result (rst=1)

Volume 3 Issue V, May 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Following waveform in Fig. 6 showing the simulation result of sub-byte transformation,

2) When rst = 0
3) Input : 2b i.e. 0010 1011
4) Output : f1 i.e. 11110001

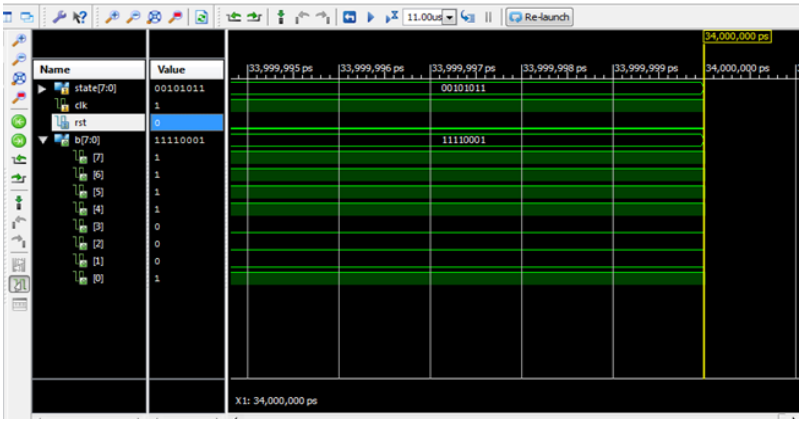


Fig. 6 Simulation Result (rst=0)

B. Simulation 2:

Following Fig. 7 showing the simulation result of inverse sub-byte transform,

1) when rst = '0'
2) input = 2A
3) output = 95

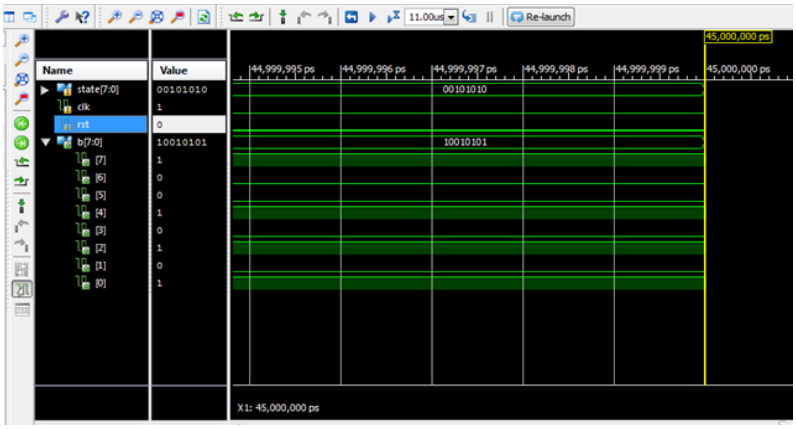


Fig. 7 Inverse Sub-Byte Transform

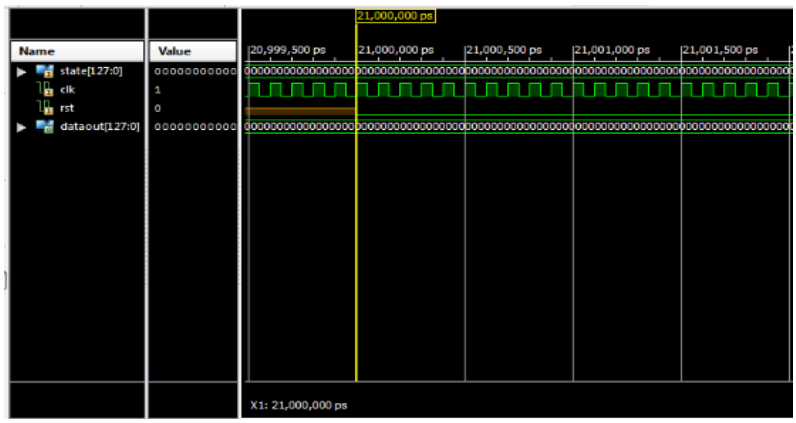


Fig. 8 Mix Column Transform

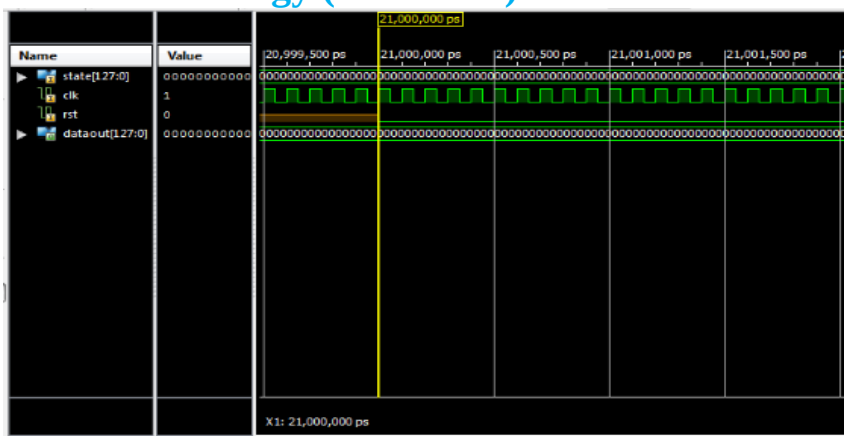


Fig. 9 Inv Mix Column Transform

VII. CONCLUSION

In this paper, in order to improve the security of the AES, we implemented the AES algorithm in encryption and decryption process. This is the work to implementing a faster cryptosystem in hardware to ensure speedy IT security. In this section performance of the proposed architectures is evaluated. The presented results are for the loop-unrolled approach where the proposed round architecture is duplicated through *10* successive units referring to the *10* rounds of the AES algorithm. Future work: an efficient FPGA implementation of 128 bit block and 128 bit key AES algorithm.

REFERENCES

[1] "Advanced Encryption Standard (AES)" Federal Information Processing Standards Publication 197, Nov. 2001.

[2] Xinmiao Zhang,and Keshab K. Parhi, "High Speed VLSI Architecture for the AES Algorithm", IEEE Transactions on very Large Scale Integration (VLSI) Systems, vol. 12, NO. 9, September 2004.

[3] A. Hodjat and I. Verbauwhede, "A 21.54 Gbit/s Fully Pipelined AES Processor on FPGA", Proceedings of the 12th Annual IEEE Symposium on Field Programmable Custom Computing Machines(FCCM 2004), pp. 308- 309,April 2004.

[4] I. Verbauwhede, P. Schaumont, H. Kuo, "Design and Performance testing of a 2.29 Gb/s Rijndael Processor", IEEE Journal of Solid-State Circuits (JSSC), March 2003.

[5] C. Yen, and B. Wu, "Simple error detection methods for hardware implementation of Advanced Encryption Standard," IEEE Transactions on Computers, Vol. 55, N°. 6, June 2006.

[6] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS Publication 197, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

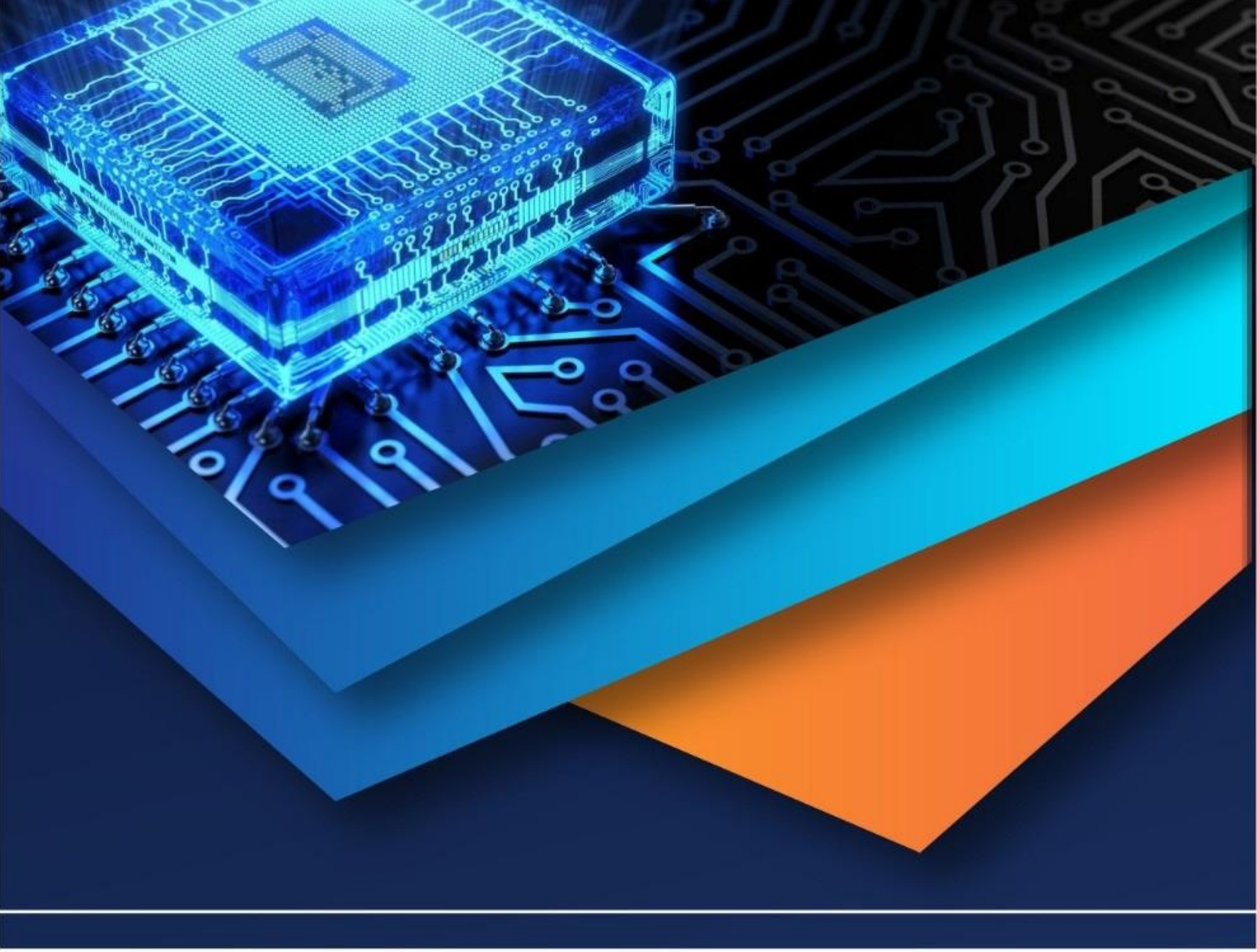
[7] J.Daeman and V.Rijmen,"AES proposal: Rijndael",september 2011. http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

[8] J. Daemon and V. Rijmen,"AES Proposal : Rijndael,"AES Algorithm Submission,Sept.1999

[9] Sujata Hiremath, M.S.Suma, "Advanced Encryption Standard Implementation on FPGA" , 2009 Second International Conference on Computer and Electrical Engineering. DOI 10.1109/ICCEE.2009.231.

[10] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS Publication 197, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[11] J. Takahashi, T. Fukunaga, K. Yamakoshi, "DFA Mechanism on the AES Key Schedule" In IEEE computer society, editor, Workshop on Fault Diagnosis and Tolerance in Cryptography, pp. 62 – 74, FDTC 2007.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY