IJRSET April 2016 Volume 3, Issue 4 Pages: 12-24
ISSN 2394-739X
International Journal for Research in Science Engineering and Technology

# AREA & POWER OPTIMIZATION OF AES ALGORITHM USING MODIFIED MIXCOLUMN WITH COMPOSITE S-BOX

**[1] S. Harihara sutharsan, [2] Aby. K.Thomas**
[1] M.Tech (VLSI), [2] Professor,
[1, 2] Department of ECE,
[1, 2] Hindustan university,
[1, 2] Chennai,

## Abstract:-

AES algorithm has attracted from various departments since it gives a very high level of security and can be implemented easily. Cryptographic applications are based on application-specific- integrated circuit (ASIC) technology, and it is to provide sufficient security level. AES Encryption and Decryption of Cryptographic algorithm plays an vital role in mail delivery system and banking due to increasing demand for secure transformation and transactions respectively. In this article, the proposed enhanced Inverse Mix-Column with Composite S-Box is designed for AES encryption and decryption. In existing AES Mix-Column, more number of logic gates are used to perform the multiplication of input stage bytes. In order to reduce this problem, the proposed enhanced Inverse Mix-column with composite S-Box is designed. This method which is used to reducing the logic gates. In addition, the proposed Inverse MixColumn is integrated into AES decryption for improving the performance of the architecture. The enhanced Inverse Mix-Column transformation with composite S-Box is to reduce the latency, area and power consumption and also reducing the hardware complexity of AES.

**Keywords:** - Advanced Standard Encryption (AES), Substitution Box(S-Box), Application-Specific-Integrated-Circuit (ASIC).

## 1. INTRODUCTION

Advanced Encryption Standard or Rijndael algorithm is a symmetric block cipher used to protect classified information and is executed in software and hardware to encrypt sensitive data. Rijndael algorithm that can process 128 bits of data blocks, using cipher keys with lengths of 128, 192, and 256 bits, which is specified by the flips standard. In 2001, Advanced Encryption Standard is designed for encryption of electronic data established by US NIST. AES is based on Rijndael Cipher which was discovered by two Belgian cryptographers, Vincent Rijmaen and Joan Daemen. These two peoples submitted a proposal to NIST at the time of AES selection process. Rijndael is a family of ciphers with different block sizes and key. The block size is 128 bits in AES. And three different cipher key lengths such as 128 , 192 and 256bits.

AES fast implementation algorithm proposed by Daemen, which takes 32-bit

data as the basic data unit of AES algorithm operation, can improve the execution performance. Intel half put forward a solution to realize AES algorithm by using AES-NI extended instruction set, which can greatly enhances the performances of AES algorithm. In 1999, first changed the normal advanced encryption standard ([AES-FIPS 197] in the form of power analysis attacks. Later, the differential power analysis (DPA) attack was introduced as one of the most capable power analysis attack. Numerous efforts have devoted to the development of efficient AES implementations against differential power analysis attack. Two representatives are mainly used in AES that are multiplicative masking and Boolean masking.

Rapid development of communication via open networks such as Internet has constructed a growing need to protect sensitive or confidential data before their transfer. This is often ensured by using encryption algorithm methods, whose key sizes are incrementing with the technology development which provides to code breakers more important computing means to search information without having the decryption key. To ensure certain security level and performance of execution, encryption and decryption algorithms require more computing and more memory that a purely software implementation cannot satisfy especially for applications such as embedded systems. In addition, a cryptographic application running on a processor with other applications, risks that other programs access to its parameters, to its variables or to its working memory areas and retrieve the used secret key or the cipher texts stored on disk to make attacks. To achieve a execution rate is fast while maintaining a high security level, it is desirable to perform crypto algorithms in hardware or jointly software and hardware implementations on SOC or PSOC platforms.

In this paper, the composite S-Box is realized to reduce the hardware complexity and power consumption. In addition, enhanced Inverse Mix-Column transformation is used to improve the performance of MixColumn transformation. The proposed enhanced Inv MixColumn with composite S-Box is implemented through Very Large Scale Integration (VLSI) system design environment. Finally, both enhanced MixColumn and minimized Composite S-Box are incorporated into AES Encryption and Decryption.

## 2. LITERATURE REVIEW

[Viktor Fischer et al, 2005] presented the MixColumn and inverse MixColumn operations used in AES. This architecture based on the Inv Mixcolumn transformation with byte-level resource. Cryptographic applications are based on application specific integrated circuit (ASIC) technology is believed to provide the security level. Elementary operations are represented at the byte level. Each byte is assumed as a polynomial with coefficients in Galois field GF (2). Hardware implementation of AES decryption is larger and slower than for encryption.

[Anitha.s. et al, 2015] explained the improved storage area network using novel mix column transformation in masked AES. Fixed coefficient technique is applied in the mixcolumn process to reduce the area, similarly for inverse mixcolumn reduced 09,0b,0dand 0e technique is designed by using shrunk time. These shrunk times is applied in the inverse mixcolumns process to analyze the performance. The AES can be used in protection and security applications such as satellite communication, Net banking and ATM.

[M.Senthil Kumar, et al, 2014] described the incorporation of wave pipelined techniques into composite S-Box and AES architectures. Enhancing the S-Box structure using pipelining enhances the speed of operations along with the security. The wave

pipelined technique is introduced in every round of the AES architecture along with the improved S-Box. Modified composite S-Box with WPT composed of 5 clock gating structures to reduce the delay and security. Clock gating structure is designed using one register and any one basic gate. Switching activity is reduced to increase the speed of the S-Box and overall AES architectures.

[K.Sandyarani, et al, 2014] described the implementation of the Sub Bytes transformation. This work avoids use of LUTs and use of composite field data path for the Sub Bytes and InvSubBytes transformations. Two Galois Fields are isomorphic, the complexity of the field operations may heavily depend on the representation of the field elements. The symmetric block cipher that can process data blocks of 128 bits using the cipher keys with lengths of 128-192-256 bits. The original message is taken to 10 round operations which produces the cipher text.

## 3. AES ALGORITHM

AES is based on Rijndael algorithm selected for data encryption standard by National Institute of Standards and Technology (NIST) in 1997. It processes data blocks of fixed size using cipher keys of length 128, 196 and 256 bits. AES-128 bit cipher keys are widely used for both encryption and decryption. Encryption side of AES can perform 4 discrete transformations in specific order like S-Box, Shift Rows, MixColumn and Add Round Key. Three numbers of rounds have to be performed in order of 10, 12 and 14 for AES-128, AES-196 and AES-256 bits respectively. S-Box is generated by taking multiplicative inverse of data input in the finite Galois Field GF ($2^8$) and it followed by an affine transformation. The irreducible polynomial of data input is represented as follows:

$$m(z) = z^8 + z^4 + z^3 + z + 1 \qquad (1)$$

Shift Rows transformation can be performed by shifting the last three rows of the state cyclically over different offsets. In MixColumn transformation, matrix multiplication can be performed over GF ($2^8$) with output of Shift Rows transformation. The matrix multiplication of MixColumn of AES is illustrated as follows,

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 01 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \qquad (2)$$

Add Round Key function can be performed in a 10 rounds of transformations can be performed, since here AES-128 bit length is considered for both input and output. The generalized flow chart of AES encryption and decryption is analyzed in fig. 1.

In Decryption side of AES, reverse operations can be performed in specific order, (i.e.) first Inv Shift Rows transformation can be performed instead of Inv S-box. Next to Inv Shift Rows, Inv S-Box can be performed. The final round of the algorithm is similar to the standard one, except that it does not have MixColumn operation. Matrix multiplication of Inv MixColumn of AES is illustrated as follows,

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \qquad (3)$$

Multiplication of Inv MixColumn requires more hardware complexity, since large number of values such as 0e, 0b, 0d and 09 are involved in Inv MixColumn transformation.

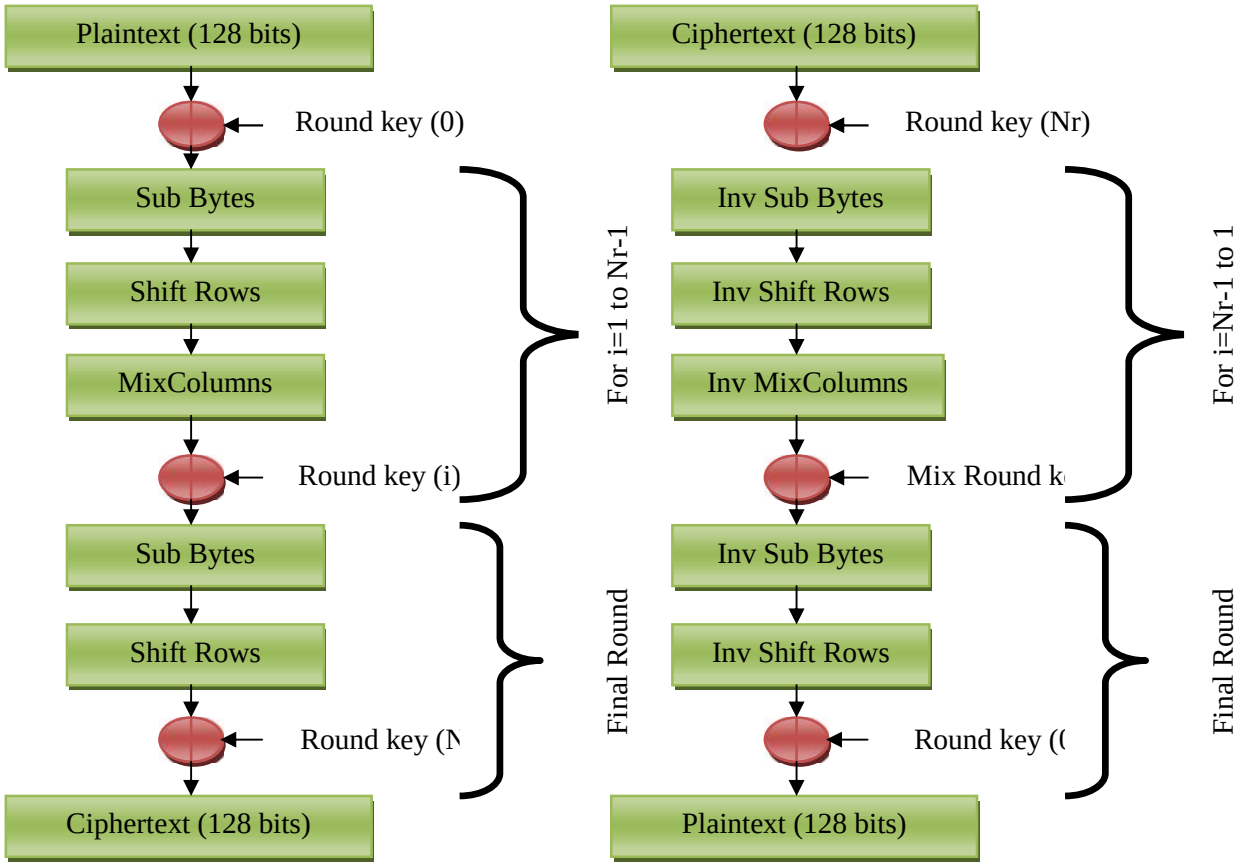


**Figure. 1 Generalized AES structure (a) Encryption (b) Decryption**

## 4. EXISTING INVERSE MIX-COLUMN TRANSFORMATION

The Mix-Column transformation operates on the State column-by-column, treating each column as a four term polynomial. These columns are considered as polynomials over GF ($2^8$) and multiplied by modulo $z^4+1$ with a fixed polynomial x (n) is given by,

$$z(n) = \{03\}z^3 + \{01\}z^2 + \{01\}z + 02 \quad (1)$$

This can be written as a matrix multiplication, s' (z) = z (n)*s (z). Where s (z) represents the state bytes from Shift Row methods. These can be represented as,

$$\begin{bmatrix} S'_{0c} \\ S'_{1c} \\ S'_{2c} \\ S'_{3c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad for\ 0 \le c < Nb(2)$$

Similarly, Inverse Mix-Column can be calculated by using $z^{-1}$ (n). The columns are considered as polynomials over GF ($2^8$) and multiplied modulo $z^4+1$ with a polynomial $z^{-1}(n)$.

$$z^{-1}(n) = \{0b\}z^3 + \{0d\}z^2 + \{09\}z + \{oe\} \quad (3)$$

Therefore $s'(z) = z^{-1}(n) \oplus s(z)$ The matrix for InvMix-Column can be represented as,

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad \text{for } 0 \leq c < Nb \quad (4)$$

The multiplication of input state bytes with fixed pre-defined polynomials can be processed by Xtime (Both input and output word length of multiplication is same) multiplication. In each multiplication steps, set of inputs are logically surrounded by EX-OR gate. The process of Xtime multiplication for InvMix-Column transformation is illustrated in fig. 1. Matrix for InvMix-Column requires more number of logic gates to perform multiplication than Mix-Column due to long word length. Typically 24 EX-OR gates are used to perform the Inv Mix-Column operation in AES.

Due to utilizing more number of logic gates, silicon area for existing Inv Mix-Column consumes large silicon area and delay. In order to reduce this problem, Circuits for InvMix-Column is realized in this paper. In next section, optimization of InvMix-Colum is described in a detailed manner.

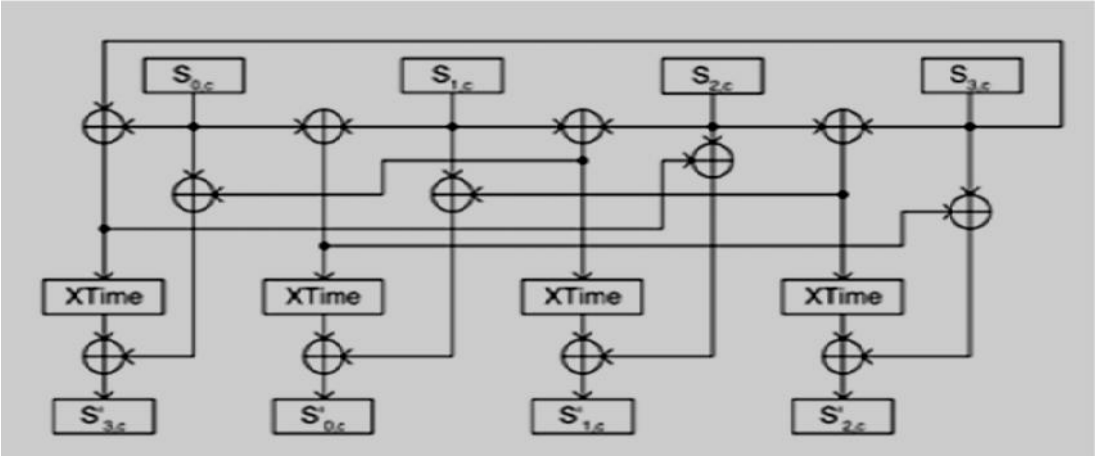


**Figure. 2 Xtime multiplication circuit for Inv Mix-Column Transformation**

## 5. PROPOSED ENHANCED INV MIX-COLUMN WITH COMPOSITE S-BOX

In this proposed work, redundant operation of Inv MixColumn is identified and eliminated to further reduce the hardware complexity of AES algorithm. Instead of using redundant methods availability resources are utilized effectively.

From equation 1,

$$s'_{0,c} = (\{0e\} * s_{0,c}) \oplus (\{0b\} * s_{1,c}) \oplus (\{0d\} * s_{2,c}) \oplus (\{09\} * s_{3,c}) \quad (5)$$

$$s'_{1,c} = (\{09\} * s_{0,c}) \oplus (\{0e\} * s_{1,c}) \oplus (\{0b\} * s_{2,c}) \oplus (\{0d\} * s_{3,c}) \quad (6)$$

$$s'_{2,c} = (\{0d\} * s_{0,c}) \oplus (\{09\} * s_{1,c}) \oplus (\{0e\} * s_{2,c}) \oplus (\{0b\} * s_{3,c}) \quad (7)$$

$$s'_{3,c} = (\{0b\} * s_{0,c}) \oplus (\{0d\} * s_{1,c}) \oplus (\{09\} * s_{2,c}) \oplus (\{0b\} * s_{3,c}) \quad (8)$$

Further, equation (5) to (8), can be simplified as

$$s'_{0,c} = [(\{09\} * s_{0,c}) \oplus (\{04\} * s_{0,c}) \oplus (\{02\} * s_{0,c}) \oplus s_{0,c}] \oplus [(\{09\} * s_{1,c}) \oplus (\{02\} * s_{1,c})] \oplus [(\{09\} * s_{2,c}) \oplus (\{04\} * s_{2,c})] \oplus [(\{09\} * s_{3,c})] \quad (9)$$

$$s'_{1,c} = [(\{09\} * s_{0,c})] \oplus [(\{09\} * s_{1,c}) \oplus (\{04\} * s_{1,c}) \oplus (\{02\} * s_{1,c}) \oplus s_{1,c}] \oplus [(\{09\} * s_{2,c}) \oplus (\{02\} * s_{2,c})] \oplus [(\{09\} * s_{3,c}) \oplus (\{04\} * s_{3,c})] \quad (10)$$

$$s'_{2,c} = [(\{09\} * s_{0,c}) \oplus (\{04\} * s_{0,c})] \oplus [(\{09\} * s_{1,c})] \oplus [(\{09\} * s_{2,c}) \oplus (\{04\} * s_{2,c}) \oplus (\{02\} * s_{2,c}) \oplus s_{2,c}] \oplus [(\{09\} * s_{3,c}) \oplus (\{02\} * s_{3,c})] \quad (11)$$

$$s'_{3,c} = [(\{09\} * s_{0,c}) \oplus (\{02\} * s_{0,c})] \oplus [(\{09\} * s_{1,c}) \oplus (\{04\} * s_{01c})] \oplus [(\{09\} * s_{2,c}) \oplus (\{09\} * s_{3,c}) \oplus (\{04\} * s_{3,c}) \oplus (\{02\} * s_{3c}) \oplus s_{3,c}] \quad (12)$$

From equation (9) to (12), it is clear that multiplication of source signals with {09}, {04} and {02} are redundantly used for calculation of Inv MixColumn. This common resources are represented are as follows,

$$c1_{09} = \{09\} * s_{0,c} \qquad c1_{04} = \{04\} * s_{0,c} \qquad c1_{02} = \{02\} * s_{0,c}$$

$$c2_{09} = \{09\} * s_{1,c} \qquad c2_{04} = \{04\} * s_{1,c} \qquad c2_{02} = \{02\} * s_{1,c}$$

$$c3_{09} = \{09\} * s_{2,c} \qquad c3_{04} = \{04\} * s_{2,c} \qquad c3_{02} = \{02\} * s_{2,c}$$

$$c4_{09} = \{09\} * s_{3,c} \qquad c4_{04} = \{09\} * s_{3,c} \qquad c4_{02} = \{02\} * s_{4,c}$$

The hardware complexity of Inv MixColumn can be absolutely reduced, when sharing these resources to all numerical calculation of Inv MixColumn. Further the equation (9) to (12) an be simplified as,

$$s'_{0,c} = (c1_{09} \oplus c1_{04} \oplus c1_{02} \oplus s_{0,c}) \oplus (c2_{09} \oplus c2_{02}) \oplus (c3_{09} \oplus c3_{02}) \oplus (c4_{09}) \quad (13)$$

$$s'_{1,c} = (c1_{09}) \oplus (c2_{09} \oplus c2_{04}) \oplus (c2_{02} \oplus s_{1,c}) \oplus (c3_{09} \oplus c3_{02}) \oplus (c4_{09} \oplus c4_{04}) \quad (14)$$

$$s'_{2,c} = (c1_{09} \oplus c1_{04}) \oplus (c2_{09}) \oplus (c3_{09} \oplus c3_{04} \oplus c3_{02} \oplus s_{2,c}) \oplus (c4_{09} \oplus c4_{02}) \quad (15)$$

$$s'_{3,c} = (c1_{09} \oplus c1_{04}) \oplus (c2_{09} \oplus c2_{04}) \oplus (c3_{09}) \oplus (c4_{09} \oplus c4_{04} \oplus c4_{02} \oplus s_{3,c}) \quad (16)$$

From equation (13) to (15), it is clear that multiplication of source input and Inv MixColumn coefficients are reduced effectively and hence the speed of Inv MixColumn can be increased successfully. Only multiplication of {09}, {04} and {02} are estimated manually, and then other calculation uses the same resources for further multiplication.

The proposed model of Inv MixColumn multiplication is illustrated in fig. 4. According to equation (13) to equation (15), multiplication of three coefficients such as {09}, {04} and {02} with corresponding inputs are estimated manually by using reduced Xtime based multiplication and their values are re-used for further multiplication whenever it required. Numbers 1, 2, 3 and 3 are used to represent the resources re-used for $s_{o,c}$, $s_{1,c}$, $s_{2,c}$ and $s_{3,c}$ respectively. Therefore, we require reduced Xtime multiplication for {09} and {04}. Multiplication {02} results are obtained by simply left shifting one bit.

State bytes $T_9$ and $T_4$ are evaluated by as follows.Multiplication of {09} with state-byte,

$t_7 = 0$ , $t_6 = b_7$ , $t_5 = b_6 \oplus b_7$ , $t_4 = b_5 \oplus b_6$ , $t_3 = b_5 \oplus b_7$ , $t_2 = t_5$ , $t_1 = t_4$ , $t_0 = b_5$

Multiplication of {04} with state-byte,

$t_7 = 0$ , $t_6 = 0$ , $t_5 = b_7$, $t_4 = b_7 \oplus b_6$ , $t_3 = b_6$, $t_2 = b_7$, $t_1 = t_4$, $t_0 = b_6$

Where, $t_0$ to $t_7$ indicates the coefficient equation and $b_0$ to $b_7$ indicates the data bits. Finally, XOR gates are used to determine the Reduce Xtime multiplication values. Enhanced this types of Inv MixColumn transformation is incorporated into AES decryption process to improve the performances of AES decryption.

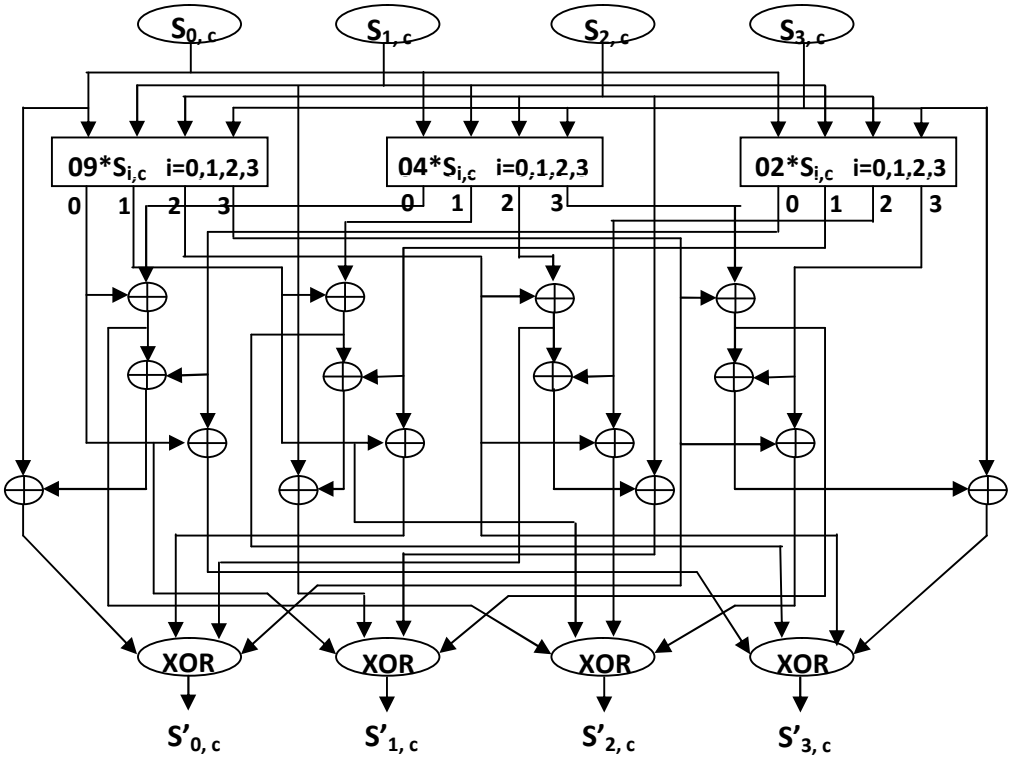


**Figure. 3 Enhanced Inv MixColumn Multiplication**

Composite AES S-Box for both AES encryption and AES decryption is represented in fig. 4. In that architecture Multiplicative Inverse unit is recognized as a high potential for more hardware complexity. MI architecture consists of multiplication of $X^2$ and multiplication of $X^a$. In this proposed model, redundant operation of both multiplication of $X^2$ and multiplication of $X^a$ are identified to further reduce the hardware complexity of Composite S-Box. Redundant operations of multiplications are identified through following equations.

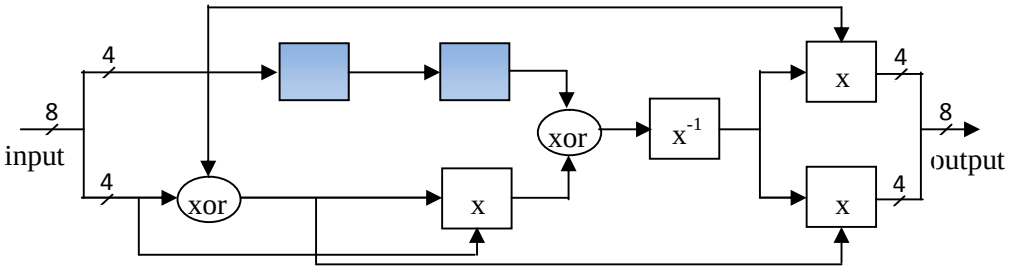

**Figure. 4 Block diagram of Multiplicative Inverse**

Let as input of multiplication $X^2$ as q3, q2, q1 and q0. Similarly, corresponding output of multiplication of $X^2$ as k3, k2, k1 and k0 respectively. Equation representation (3) to (6) gives the output of multiplication of $X^2$

$$k3 = q3 \quad (3)$$

$$k2 = q3 \oplus q2 \quad (4)$$

$$k1 = q2 \oplus q1 \quad (5)$$

$$k0 = q3 \oplus q1 \oplus q0 \quad (6)$$

Similarly, let as input of multiplication $X^a$ as q3, q2, q1 and q0, corresponding output of multiplication of $X^a$ as K3, K2, K1 and K0 respectively. Equation representation (7) to (10) gives the output of multiplication of $X^a$ .

$$K3 = q0 \oplus q1 \oplus q2 \oplus q3 \quad (7)$$

$$K2 = q1 \oplus q3 \quad (8)$$

$$K1 = q2 \quad (9)$$

$$K0 = q2 \oplus q3 \quad (10)$$

In our proposed work, we combine these two multiplications and get the minimized circuit which is obtained by following simplifications. From equation (7) and equation (3) to (6), we can also write,

$$K3 = q0 \oplus q1 \oplus q3 \oplus q2 \oplus q1 \oplus q2 \oplus q3 \oplus q3$$

We know that, A XOR A = 0. Hence, we get,

$$K3 = q0 \oplus q3 \quad (11)$$

From equation (8) and equation (3) to (6), we can also write,

$$K2 = q1 \oplus q2 \oplus q3 \quad (12)$$

From equation (9) and equation (3) to (6), we can also write,

$$K1 = q2 \oplus q3 \quad (13)$$

From equation (10) and equation (3) to (6), we can also write,

$K0 = q2 \oplus q3 \oplus q3$ Hence, we get,

$$K0 = q2 \quad (14)$$

From equation (12) and equation (13), a common factor $q2 \oplus q3$ can be repeated. Hence, we can reuse the same resource for both of the place.

Let $h = q2 \oplus q3$ Therefore finally, we get,

$$K3 = q0 \oplus q3 \quad (15)$$

$$K2 = q1 \oplus h \quad (16)$$

$$K1 = h \quad (17)$$

$$K0 = q2 \quad (18)$$

Equation (15) to equation (18) represents the proposed equations for minimized

Composite S-Box. The architecture of combined both multiplication of $X^2$ and multiplication of $X^a$ is illustrated in fig. 4. The proposed combined multiplication uses only three XOR gates to produce sufficient data suited for inverse multiplication operation.

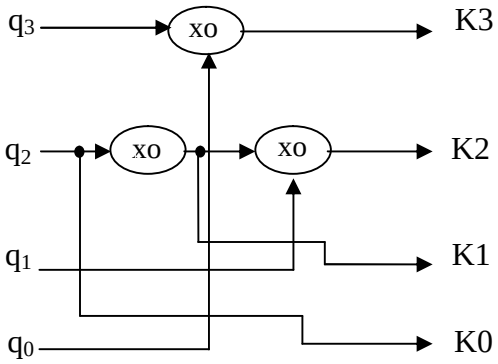


**Figure. 5 Proposed reduced structure of combined both multiplication of $X^2$ and multiplication of $X^a$**

It reduces the five XOR gates when compared to the traditional multiplication structures. Further, this structure is incorporated into Composite S-Box to reduce the hardware complexity and power consumption of AES encryption and decryption process.

## 6. RESULTS AND DISCUSSION

In this paper, the design of Enhanced Inv MixColumn with Composite S-Box is designed through Verilog Hardware Description Language (Verilog HDL) . The simulation results of minimized Composite S-Box and enhanced Inv MixColumn of AES encryption and AES decryption are validated by ModelSim 6.3C and Synthesis results are evaluated by using Xilinx 10.1i (Family-Virtex 4, Devices-XC4VLX15/XCVLX25, Package-FF668 and Speed:-12) design tool. The simulation result of AES encryption is demonstrated in fig. 5. Encryption of 128-bits data is obtained in fig. 5 though four transformations (Minimized Composite S-Box, Shift Rows, MixColumn and Add Round Key). For instance, 128-bit data input is considered as 85fc3432abcd53210be0ac125ccdb110 in hexadecimal format. Encrypted data obtained in simulation result is 9ba71628a7ee25e0416a7354a15b1321 in hexadecimal format, which is illustrated in fig. 5. Similarly, the simulation result of AES decryption is illustrated in fig. 6.

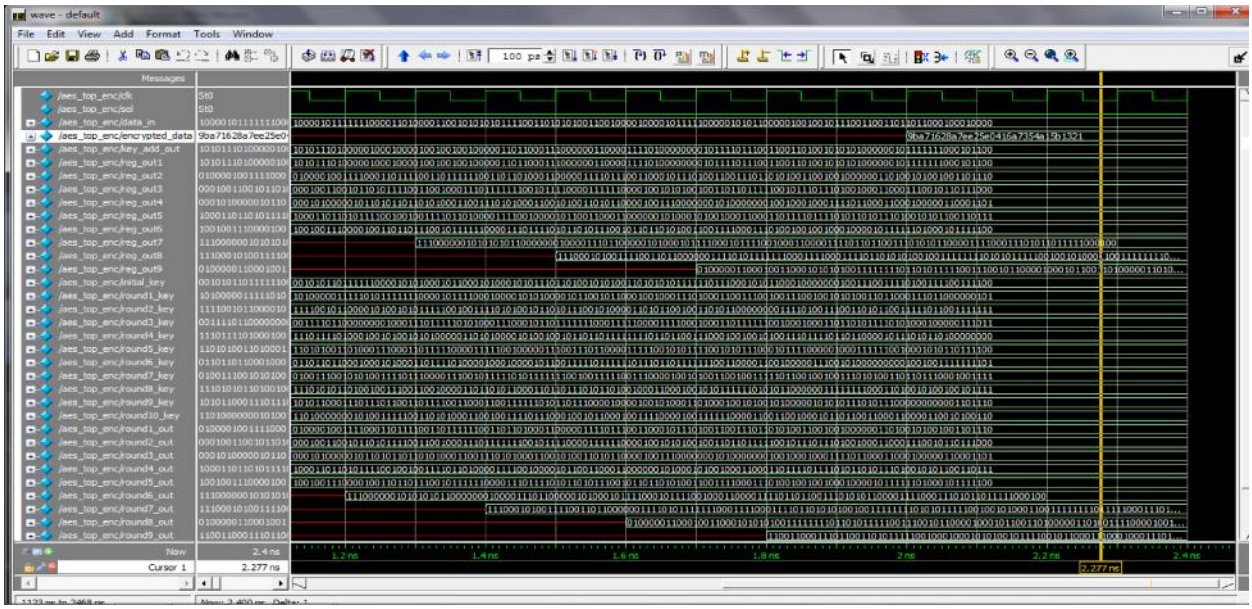

**Figure.6 Simulation result of Encryption by using Minimized Composite S-Box and Enhanced Inv MixColumn**

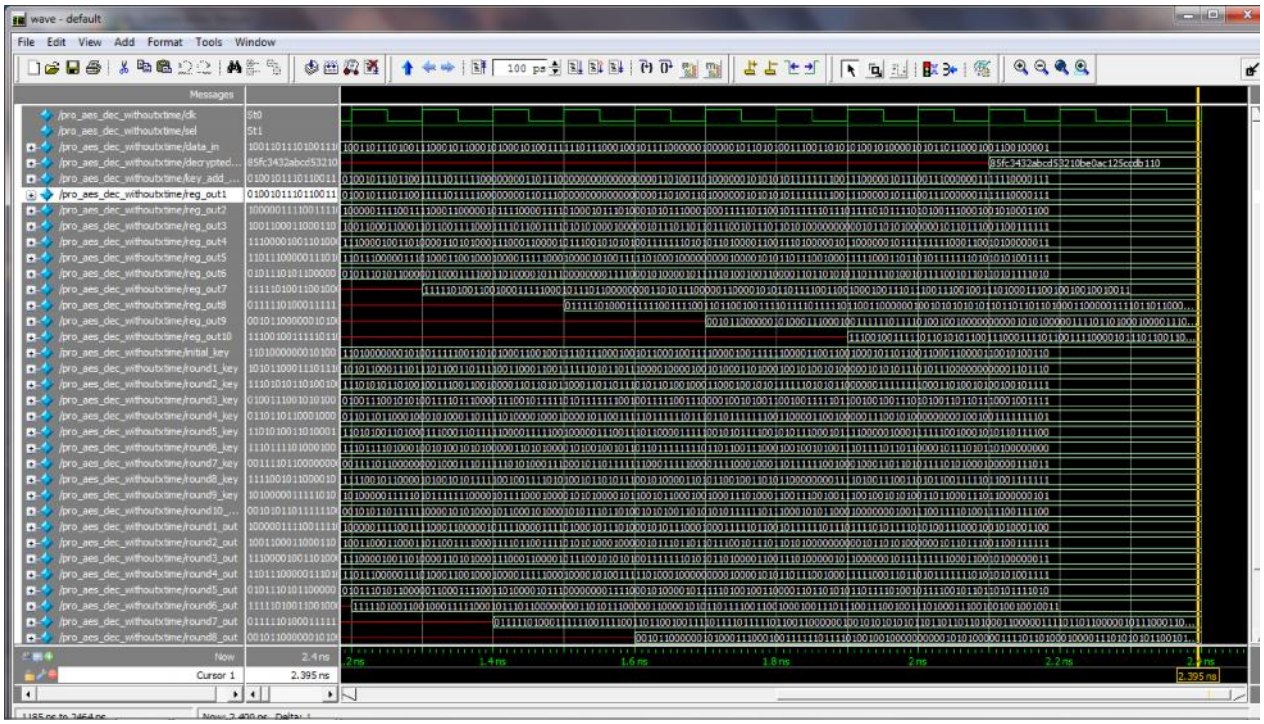

**Figure.7 Simulation result of Decryption by using Minimized Composite S-Box and Enhanced Inv MixColumn**

| Types | Slices | LUT | Delay(ps) | Frequency(MHz) | Power(mW) |
|---|---|---|---|---|---|
| **Traditional AES Decryption** | 10,307 | 19,035 | 7042 | 142.002 | 6554 |
| **Proposed Minimized** | 8539 | 15,693 | 6762 | 147.886 | 6472 |

| | | | | | |
|---|---|---|---|---|---|
| **Composite S-Box & Enhanced Inv MixColumn based AES Decryption** | | | | | |

**Table.1 Comparison of Traditional Decryption and Proposed Enhanced Inv MixColumn with Minimized Composite S-Box based AES Decryption**

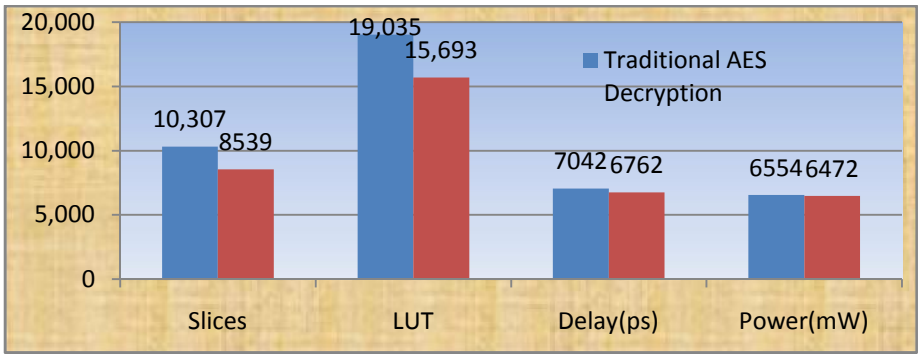


**Figure.8 Performance of Traditional AES Decryption and Proposed Minimized Composite S-Box & Enhanced Inv MixColumn based AES Decryption**

From Table 1, it is clear that Proposed Minimized Composite S-Box & Enhanced Inv MixColumn based AES Decryption offers 17.15% reduction of Slices, 17.55% reduction of LUT, 3.97% reduction of delay consumption and 1.25% reduction of power consumption than traditional AES Decryption.

## CONCLUSION

In this paper, Optimized InvMix-Column with composite S-Box of AES decryption is designed through Very Large Scale Integration (VLSI) System design environment. The structure of Xtime multiplication can be optimized by reducing the logic gates. In Optimized InvMix-Column design 9 gates are reduced to perform the Xtime multiplication. Low power consumption, high speed and less area utilization are the main key factors in VLSI System design environment. Hence, proposed model aims to reduce the hardware complexity, Power consumption and improve the speed of the System. Further Proposed Composite S-Box and Enhanced Inv MixColumn transformations are integrated into AES encryption and AES decryption process respectively. Proposed Minimized Composite S-Box & Enhanced Inv MixColumn based AES Decryption offers 17.15% reduction of Slices, 17.55% reduction of LUT, 3.97% reduction of delay consumption and 1.25% reduction of power consumption than traditional AES Decryption. In future, the proposed work helps to security Systems like Net banking for money transactions, Space and terrestrial communication.

## REFERENCES

[1] Viktor Fischer, Milos Drutarovsky, Pawel Chodowiec and Francois Gramain, "InvMixColumn Decomposition and Multilevel Resource Sharing in AES Implementations" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 13, No. 8, 2005.

[2] M. Anitha Christry and S. Sridevi Sathya Priya, "Design of Low Power Mixcolumn in Advanced Encryption Standard Algorithm" International Journal

of Scientific & Engineering Research, Vol. 5, Issue. 4, pp: 64-68, 2014.

[3] Anitha, S., and Suganya, M., 2015. "Area optimized in storage area network using Novel Mix column Transformation in Masked AES" International Journal of Engineering Trends and Technology (IJETT), Vol. 20, No. 6, pp: 275-282.

[4] M. Senthil Kumar and S. Rajalakshmi, "Incorporation of Wave Pipelined Techniques into Composite S-Box and AES Architectures" Research Journal of Applied Sciences, Engineering and Technology (RJASET), Vol. 8, No. 15, pp: 1717-1723, 2014.

[5] Sandyarani, K., and Nirmal kumar, P., 2014. "Design of high speed AES-128 using Novel Mix Column Transformation and Sub Bytes" Journal of Computer Applications (JCA), Vol. 7, Issue. 2, pp: 57-60.

[6] Advanced Encryption Standard (AES), FIPS-197, Nat. Inst. of Standards and Technol, 2001

[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in Proc. CRYPTO, 1999, vol. LNCS 1666, pp. 388–397.

[8] K. Munusamy, C. Senthilpari, and D. C. Kho, "A low power hardware implementation of S-Box for Advanced Encryption Standard" In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on (pp. 1-6). IEEE

[9] S. Messerges, "Securing the AES finalists against power analysis attacks," in Proc. FSE LNCS, 2000, vol. 1978, pp. 150–164

[10] Yi Wang and Yajun Ha, "FPGA-Based 40.9-Gbits/s Masked AES With Area optimization for Storage Area Network" IEEE Transactions n Circuits And Systems, Vol. 60, No. 1, pp: 36-40, 2013