Research Journal of Applied Sciences, Engineering and Technology 12(1): 52-62, 2016
DOI:10.19026/rjaset.12.2303
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: August 21, 2015 Accepted: September 11, 2015 Published: January 05, 2016

Research Article

Design and Implementation of Enhanced Affine and Inverse Affine Transformation Based Composite S-box for AES Encryption and Decryption

[1]M. Vaidehi and [2]B. Justus Rabi
[1]Karpagam University, Coimbatore,
[2]Shri Andal Alagar College of Engineering, Chennai, TN, India

Abstract: The Substitution Box (S-Box) and Inverse MixColumn (Inv MixColumn) forms core building blocks in Advanced Encryption Standard (AES) based Security Algorithm. This study presents full custom design of Composite S-Box by reducing the composite field arithmetic of Multiplication Inverse (MI) and Affine/Inverse Affine Transformation. Design of proposed new MI and Affine/Inverse Affine Transformation techniques are integrated in Composite S-Box of both AES Encryption and Decryption. Very Large Scale Integration (VLSI) System design environment is considered in this research work to measure the performance improvement. High Speed, less area utilization and Lower power consumptions are the important parameter in VLSI System design environment. Hence, the main goal of this research work is to reduce the hardware complexity, Power and Delay consumption of AES Encryption and Decryption process. The principle of reducing the redundant functions is used in both MI and Affine/Inverse Affine Transformation of Proposed Composite S-Box design for reducing the hardware complexity and power consumption. Proposed new Composite S-Box design offers 6.52% reduction of Slices, 5.68% reduction of Look up Tables (LUTs), 2.24% reduction of delay and 6.15% reduction of Power consumption than traditional Composite S-Box design. Further Proposed new composite S-Box design is integrated into both AES encryption and AES decryption process to improve the performance evaluation of AES algorithm.

Keywords: Advanced Encryption Standard (AES), Affine/Inverse affine transformation, Inverse Mixcolumn, Multiplicative Inverse (MI), Very Large Scale Integration (VLSI)

INTRODUCTION

Advanced Encryption Standard (AES) algorithm provides best security in the field of wireless transmission mechanism. With ever increasing the 3G and 4G mobile products, low on-chip and high speed cryptography algorithms are necessary. For instance, in a banking system (net banking, mobile banking, ATM) and mail delivery system, secure crypto mechanism is essential with high speed and low-on chip processors. AES meets advantages in both Security and Hardware Performance Improvement. Hence, a lot of researches have been suggested the AES processes for transferring different kinds of things in Wireless. In general AES Encryption and Decryption has four steps for producing Chiper data and reconstructing the plain data.
They are:

- Substitution Box (S-Box)
- Shift rows transformation
- MixColumn transformation
- Add round key transformation for AES Encryption
- Inverse shift rows transformation
- Inverse S-Box (Inv S-Box)
- Inverse MixColumn transformation
- Add Round Key transformation for AES Decryption, respectively

Each and every transformation has critical logics developed by hardware for making secure transformation. AES is a Rijndael algorithm selected for encrypting and decrypting the digital information by National Institute of Standards and Technology (NIST) in 1997 (Jamil, 2004). A single Multi-Core architecture is developed in Wang *et al*. (2010) for performing AES Encryption and Decryption process. Similarly, Dual Single-Core architecture has been developed in Li and Li (2008). In those research works, single VLSI based core architecture controls the operation of S-Box/Inv S-Box, MixColumn/Inv MixColumn, Shift Rows/Inv Shift Rows and Add Round Key transformation.

Among all those transformation, S-Box/Inv S-Box and MixColumn/Inv MixColumn transformation has more hardware complexity than other transformations like Shift Rows/Inv Shift Rows and Add Round Key Transformation. In general, Add Round Key transformation performs N rounds of redundant operation of S-Box/Inv S-Box, Shift Rows/Inv Shift Rows and MixColumn/Inv MixColumn transformation. Hence, any architectural improvements in S-Box/Inv S-

Corresponding Author: M. Vaidehi, Karpagam University, Coimbatore, TN, India

Box or MixColumn/Inv MixColumn would also helps to improve the performances of Add Round Key transformation.

In this study, a new Affine Transformation (AT)/Inv Affine Transformation (IAT) techniques is proposed by realizing the architecture of traditional AT/IAT and eliminating the redundant logics by solving the arithmetic logics. Also Multiplication Inversion (MI) unit has been realized and improved. Proposed new AT/Inv AT and MI unit has been incorporated into Composite S-Box for improving the performance of AES Encryption and AES Decryption.

LITERATURE REVIEW

The Rijndael algorithm is used for encrypting and decrypting the data. AES standards are selected for encrypting the data by National Institute of Standards and Technology (NIST) in 1997 (Jamil, 2004). AES Rijndael proposal has been implemented in Sklavos and Koufopavlou (2002) with the help of VLSI System design environment. It consists of flexible architecture for both encrypting and decrypting the data (Li, 2006). In order to perform four types of functions (S-Box/Inv S-Box, Shift Rows/Inv Shift Rows, MixColumn/Inv MixColumn, Add Round Key) Dual Core architecture has been proposed in 2008 (Li and Li, 2008). Further in 2010, (Wang *et al*., 2010) Single-Multi Core has been proposed for integrating four types of functions in single architecture. Also it is possible to design the fault detection scheme for transformation of data in AES Encryption and AES Decryption process. Mozaffari-Kermani and Reyhani-Masoleh (2010), a Fault detection technique has been proposed with the help of hamming codes for AES Encryption and AES Decryption. Further to improve the fault detection technique of AES Encryption and AES Decryption, Pipelining technique has been introduced in Liu *et al*. (2015).

Ahmad and Rezaul Hasan (2013), Composite S-Box has been designed to reduce the number of LUTs. The composite S-Box has been designed with the help of Affine Transformation (AT) and Multiplicative Inverse (MI) operation. Function of AES Composite S-Box has been controlled by digital logic circuits. When compared to Affine Transformation technique, MI unit has more critical computational paths. Hence, a lot of research works have been worked on reducing the MI unit of AES Composite S-Box. In Sandyarani and Nirmal Kumar (2014), Novel MixColumn transformation and Sub Bytes techniques has been proposed, in which XOR gate reduction techniques are used for improving the AES Encryption and AES Decryption process. Thillaikkarasi and Vaishnavi (2014), Optimum Composite AES S-Box has been proposed for reducing the critical path of data flow architecture in S-Box architecture. Further Wave Pipelining Techniques (WPT) is used in AES S-Box to increase the speed and throughput of the system. Sandhya and Deepa (2013) and Sandhyarani and Nirmal Kumar (2014a, 2014b), Composite S-Box based AES Encryption and Decryption techniques are integrated into Cipher Block Chaining (CBC) and Counter Mode (CM) mode of AES Encryption and Decryption. In addition to Composite S-Box, Inverse MixColumn transformation also plays a vital role in Performance Improvement. In order to improve the architectural performances of Inverse MixColumn transformation, Optimized MixColumn has been designed in Balamurugan and Logashanmugam (2015) and Khose and Raut (2014).

RIJINDAEL AES ALGORITHM

Rijndael AES Algorithm processes data blocks of fixed size using cipher keys of length 128, 192 and 256 bits. 128-bit AES Encryption has been widely used for encryption and decryption of AES. Both Encryption and Decryption has four transformations for encrypting and decrypting the data. General data flow structure for 128 bit AES Encryption and Decryption is illustrated in Fig. 1.

As shown in Fig. 1, Final Round of both AES Encryption and AES Decryption doesn't have MixColumn and Inv MixColumn transformation function respectively. It has 10 Number of rounds for exhibiting cipher data from Encryption process and plain data from Decryption process.

Sub-Bytes transformation (): In Sub-Bytes transformation, substitution techniques are involved with the help of Substitution tables (LUTs/ Memories/ROM). In general, substitution box has been generated by two important transformation techniques:

- **Multiplicative Inverse (MI) transformation:** Taking MI given input state bytes in Galois Field GF (2^8).
- **Affine Transformation (AT):** Taking affine transformation of MI outputs. In this operation, XOR functions can be performed with in combined input bits itself.

Similarly, Inverse Sub-Bytes transformation performs inverse operation.

Shift rows transformation (): In Shift Rows transformation, single bytes of every row have been shifted for increasing the security. Similarly, reverse process has been followed in Inv Shift Rows transformation.

MixColumn transformation (): In MixColumn multiplication, state bytes are treated as a four-term

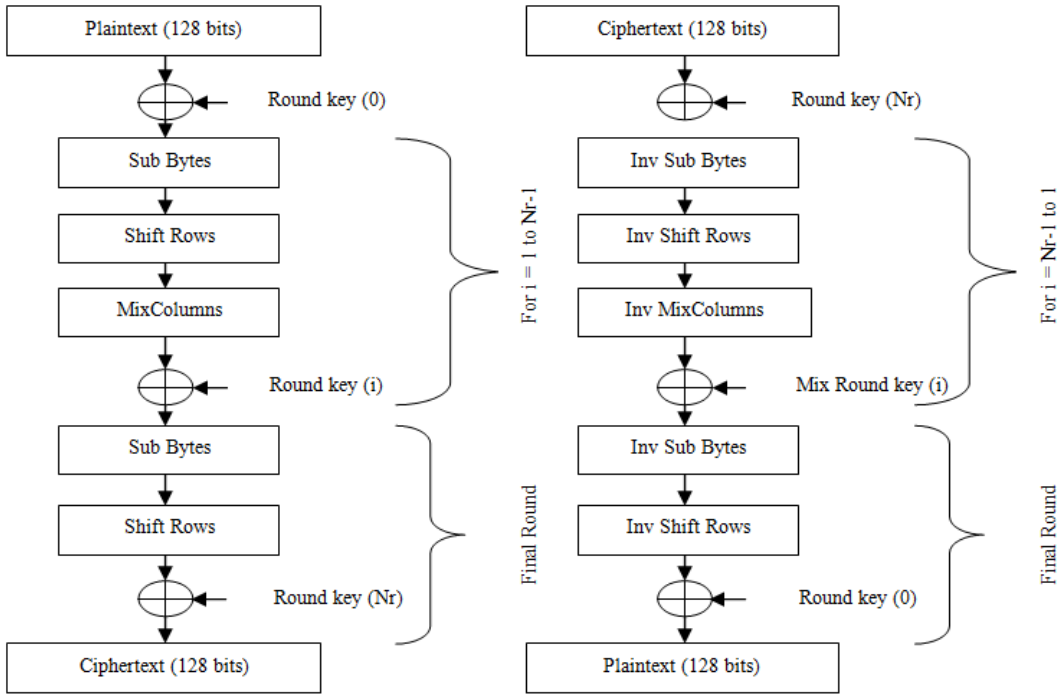

Fig. 1: General data flow structure for 128 bit AES; (a): Encryption; (b): decryption

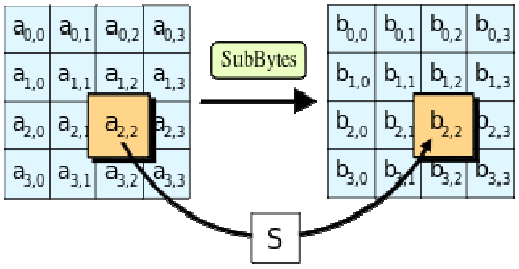

Fig. 2: Block diagram of sub bytes transformation

polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo x^4+1 with a fixed polynomial. Similarly, in Inv MixColumn transformation, reverse process has been followed (i.e.,) state bytes are multiplied with another fixed polynomials.

Add round key transformation (): In the Add Round Key () transformation, a Round Key is added to the State by a simple bitwise XOR operation. Each Round Key consists of Nb words. Those Nb words are added into the columns of the State.

DESIGN OF AES COMPOSITE S-BOX

In Rijndael AES Algorithm, Substitute Box (S-Box) is the first step of encryption operation. In S-Box transformation, Substitution bytes are used to replace the state bytes. Symbolic block diagram of Sub Bytes transformation is illustrated in Fig. 2. The Substitution table has been generated by performing two functions on input state bytes. They are:

- Multiplicative inverse
- Affine transformation

In MI unit, inverse multiplication has been performed with the help of digital logics and affine transformation has been performed by taking exclusive operation MI outputs with 63. Block diagram of Composite S-Box is illustrated in Fig. 3, which performs both Sib-Bytes and Inv Sub-Bytes transformation by switching combinational logics using multiplexers. To make a compact AES implementations composite field inversions are extended to GF $(((2^2)^2)^2)$ from GF (2^8). This approach is used to reduce the chip size to design. The column vector of the input State matrix first goes into isomorphic transformation from GF (2^8) into the composite field GF $(((2^2)^2)^2)$ followed by inversion in composite field and inverse isomorphic transformation. Finally, an affine transformation is carried out to create the cipher data.

In S-Box Transformation, Isomorphic Mapping function (q) has performed initially on the input State Bytes (b). Secondly, Inverse Multiplication (q') has been performed on isomorphic output data. Then inverse isomorphic mapping function (b') has been performed on q'. Finally, affine transformation function (b'') has been carried out by inverse

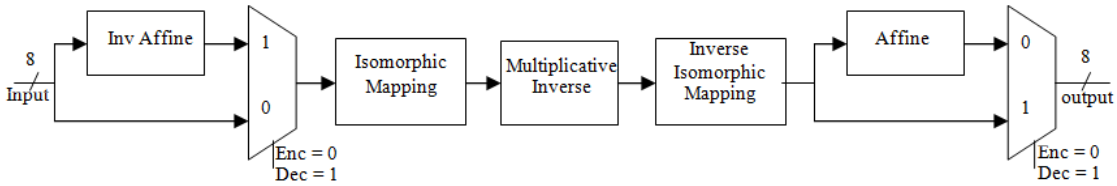

Fig. 3: Block diagram of composite S-box

isomorphic functions. In general affine transformation (AT) function involves the multiplication with 63.

In Inv S-Box Transformation, Inv Affine Transformation (q') function has been performed initially on the given input bytes (q''). Secondly, Multiplicative Inverse function (q) has been performed on IAT outputs. Finally, inverse isomorphic mapping function has been performed on MI outputs to reconstruct the original input State-Bytes.

The State matrix representation of Affine and Inverse Affine transformation techniques is demonstrated as follows:

$$[\text{AT (b')}] = \begin{bmatrix} b_o'' \\ b_1'' \\ b_2'' \\ b_3'' \\ b_4'' \\ b_5'' \\ b_6'' \\ b_7'' \end{bmatrix} = \begin{bmatrix} \overline{b_0' \wedge b_4' \wedge b_5' \wedge b_6' \wedge b_7'} \\ \overline{b_0' \wedge b_1' \wedge b_5' \wedge b_6' \wedge b_7'} \\ b_0' \wedge b_1' \wedge b_2' \wedge b_6' \wedge b_7' \\ b_0' \wedge b_1' \wedge b_2' \wedge b_3' \wedge b_7' \\ b_0' \wedge b_1' \wedge b_2' \wedge b_3' \wedge b_7' \\ \overline{b_1' \wedge b_2' \wedge b_3' \wedge b_4' \wedge b_5'} \\ \overline{b_2' \wedge b_3' \wedge b_4' \wedge b_5' \wedge b_6'} \\ b_3' \wedge b_4' \wedge b_5' \wedge b_6' \wedge b_7' \end{bmatrix} \quad (1)$$

$$[\text{IAT (q'')}] = \begin{bmatrix} q_o'' \\ q_1'' \\ q_2'' \\ q_3'' \\ q_4'' \\ q_5'' \\ q_6'' \\ q_7'' \end{bmatrix} = \begin{bmatrix} \overline{q_2'' \wedge q_5'' \wedge q_7''} \\ q_0'' \wedge q_3'' \wedge q_6'' \\ \overline{q_1'' \wedge q_4'' \wedge q_7''} \\ q_0'' \wedge q_2'' \wedge q_5'' \\ q_1'' \wedge q_3'' \wedge q_6'' \\ q_2'' \wedge q_4'' \wedge q_7'' \\ q_0'' \wedge q_3'' \wedge q_5'' \\ q_1'' \wedge q_4'' \wedge q_6'' \end{bmatrix} \quad (2)$$

Similarly, State matrix transformation of isomorphic and inverse isomorphic transformation is demonstrated as follows:

$$[\text{ISO (b)}] = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \quad (3)$$

$$[\text{ISO (b)}] = \begin{bmatrix} b_0 \wedge b_2 \\ b_1 \wedge b_6 \wedge b_7 \\ b_2 \wedge b_5 \\ b_1 \wedge b_3 \wedge b_6 \wedge b_7 \\ b1 \wedge b5 \wedge b7 \\ b_1 \wedge b_4 \wedge b_5 \wedge b_6 \\ b_1 \wedge b_3 \wedge b_2 \wedge b_5 \wedge b4 \wedge b_6 \\ b_5 \wedge b_7 \end{bmatrix} \quad (4)$$

$$[\text{ISO}^{-1}\text{ (b)}] = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} \quad (5)$$

$$[\text{ISO}^{-1}\text{ (b)}] = \begin{bmatrix} q_0 \wedge q_1 \wedge q_3 \wedge q_5 \wedge q_6 \\ q_4 \wedge q_7 \\ q_1 \wedge q_3 \wedge q_5 \wedge q_6 \\ q_1 \wedge q_3 \\ q_1 \wedge q_5 \wedge q_7 \\ q_1 \wedge q_2 \wedge q_3 \wedge q_5 \wedge q_6 \\ q_2 \wedge q_3 \wedge q_4 \wedge q_5 \wedge q_6 \\ q_1 \wedge q_2 \wedge q_3 \wedge q_5 \wedge q_6 \wedge q_7 \end{bmatrix} \quad (6)$$

The block diagram of Multiplicative Inverse (MI) unit is illustrated in Fig. 4. It performs the inverse operation of input x (i.e., 1/x). It consists of

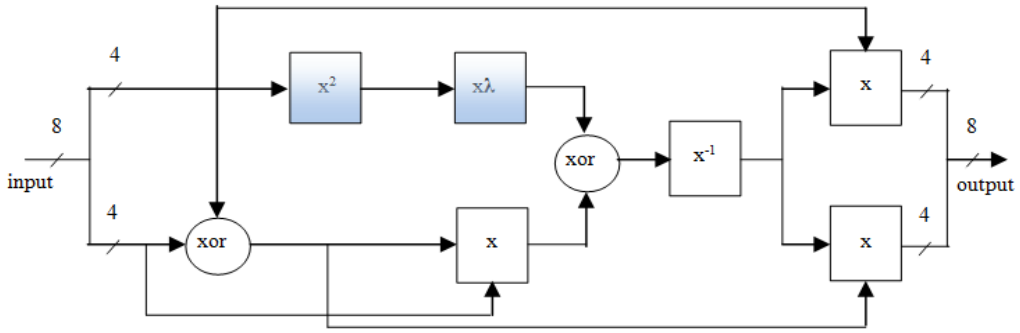

Fig. 4: Block diagram of multiplicative inverse

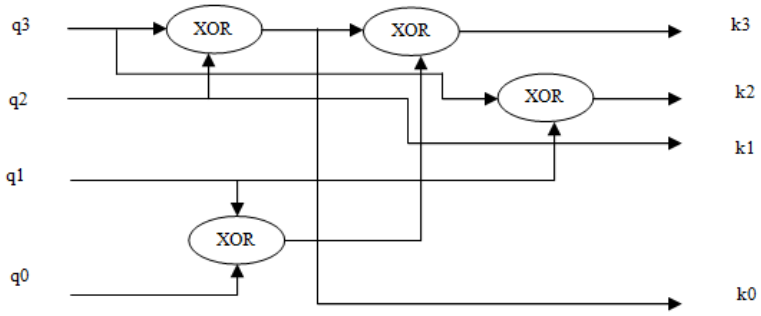

Fig. 5: Multiplication of x λ

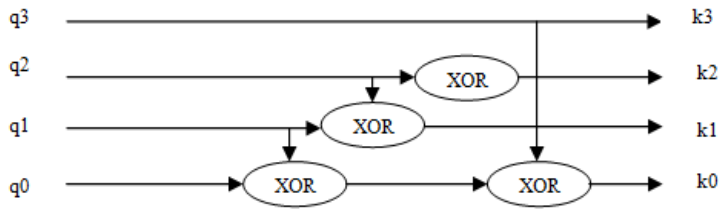

Fig. 6: Multiplication of x^2

mathematical block x λ and x^2. Both these blocks have certain kinds of Boolean logics for supporting the inverse multiplication block. Multiplication of x λ and x^2 are illustrated in Fig. 5 and 6 respectively.

Finally, the multiplexer circuit is used to select S-Box/Inv S-Box outputs based on the select line. Hence, this S-Box circuit is referred as Composite S-Box. Instead of using direct Look up Tables (LUTs), this Composite S-Box gives more advantage in terms of less chip utilization, lower delay and power consumptions.

PROPOSED ENHANCED AFFINE AND INVERSE AFFINE TRANSFORMATION BASED COMPOSITE S-BOX

In this study, redundant logic functions of Composite S-Box is realized with the help of solving the Boolean logics. As stated earlier, architecture of Composite S-Box has different kinds of blocks like Affine Transformation (AT), Inverse Affine Transformation (IAT), Isomorphic Mapping (ISO), Inverse Isomorphic Mapping (Inv ISO) and Multiplicative Inverse (MI) for performing inverse multiplication of input. Each and every block has digital logics to support the function of inverse multiplication. Redundant logic function of each and every block has identified with the help of Boolean logic expressions.

Equation (1) establishes the Affine Transformation techniques. In this equation, there are four redundant logic functions are identified:

Redundant Function_AT1 = $b'_6 \wedge b'_7$
Redundant Function_AT2 = $b'_4 \wedge b'_5$
Redundant Function_AT3 = $b'_0 \wedge b'_1$
Redundant Function_AT4 = $b'_2 \wedge b'_3$

Hence, equations of Affine Transformations can be reduced as follows:

AT [0] = ~ (b'_0 ^ Redundant Function_AT2 ^ Redundant Function_AT1)
AT [1] = ~ (Redundant Function_AT3 ^ b'_5 ^ Redundant Function_AT1)

AT [2] = Redundant Function_AT3 ^ b'_2 ^ Redundant Function_AT1
AT [3] = Redundant Function_AT3 ^ Redundant Function_AT4 ^ b'_7
AT [4] = Redundant Function_AT3 ^ Redundant Function_AT4 ^ b'_4
AT [5] = ~ (b'_1 ^ Redundant Function_AT4 ^ Redundant Function_AT2)
AT [6] = ~ (Redundant Function_AT4 ^ Redundant Function_AT2 ^ b'_6)
AT [7] = b'3 ^ Redundant Function_AT2 ^ Redundant Function_AT1

When compared to traditional AT technique, 12 gates are reduced in proposed Enhanced AT techniques. The circuit diagram of AT technique is illustrated in Fig. 7. Similarly, Redundant Functions of Inverse Affine Transformation techniques are identified from Eq. (2) as follows:

Redundant Function_IAT1 = q'_2 ^ q'_5
Redundant Function_IAT2 = q'_3 ^ q'_6
Redundant Function_IAT3 = q'_4 ^ q'_7

Hence, equations of Inverse Affine Transformation can be reduced as follows:

IAT [0] = ~ (Redundant Function_IAT1 ^ q'_7)
IAT [1] = q'_0 ^ Redundant Function_IAT2
IAT [2] = ~ (q'_1 ^ Redundant Function_IAT3)
IAT [3] = q'_0 ^ Redundant Function_IAT1
IAT [4] = q'_1 ^ Redundant Function_IAT2
IAT [5] = q'_2 ^ Redundant Function_IAT3
IAT [6] = q'_0 ^ q'_3 ^ q'_5
IAT [7] = q'_1 ^ q'_4 ^ q'_6

When compared to traditional IAT technique, 3 gates are reduced in proposed Enhanced IAT technique. The circuit diagram of proposed Enhanced IAT technique is illustrated in Fig. 8. Equation (4)

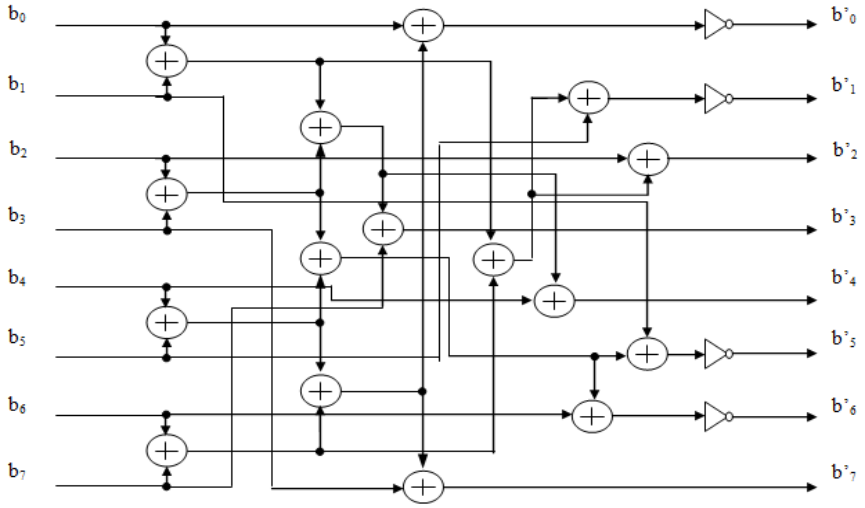

Fig. 7: Circuit diagram of proposed enhanced AT technique

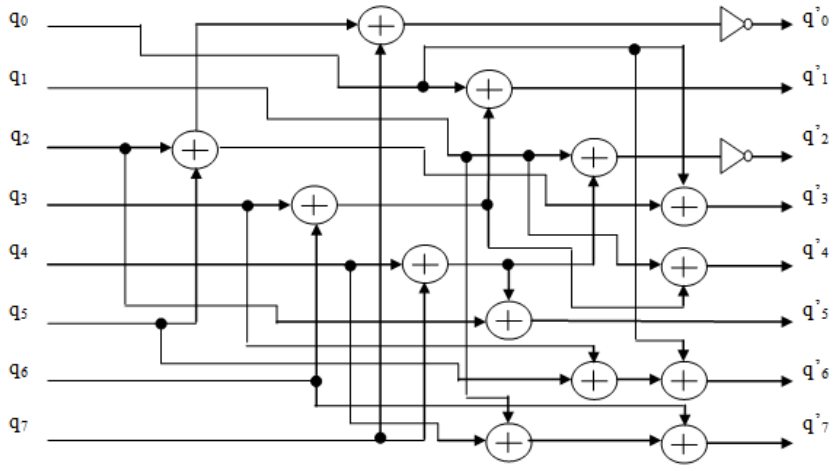

Fig. 8: Circuit diagram of proposed enhanced IAT Technique

establishes the Isomorphic Mapping Transformation. In this equation, five redundant functions are identified:

Redundant Function_ISO1 = b_6 ^ b_7
Redundant Function_ISO2 = b_5 ^ b_7
Redundant Function_ISO3 = b_2 ^ b_5
Redundant Function_ISO4 = b_1 ^ b_3
Redundant Function_ISO5 = b_4 ^ b_6

Hence, equations of Isomorphic Transformation can be reduced as follows:

ISO [0] = b_0 ^ b_2
ISO [1] = b_1 ^ Redundant Function_ISO1
ISO [2] = b_2 ^ b_3 ^ Redundant Function_ISO2
ISO [3] = Redundant Function_ISO2
ISO [4] = Redundant Function_ISO4 ^ Redundant Function_ISO1
ISO [5] = b_1 ^ b_5 ^ Redundant Function_ISO5
ISO [6] = Redundant Function_ISO3 ^ Redundant Function_ISO4 ^ Redundant Function_ISO5
ISO [7] = Redundant Function_ISO2

When compared to traditional ISO technique, 5 gates are reduced in proposed Enhanced ISO technique. The circuit diagram of proposed Enhanced ISO technique is illustrated in Fig. 9. Similarly, redundant functions of Inverse Isomorphic Transformation are identified from Eq. (6) as follows:

Redundant Function_InvISO1 = q_1 ^ q_3
Redundant Function_InvISO2 = q_5 ^ q_6
Redundant Function_InvISO3 = q_7 ^ q_5

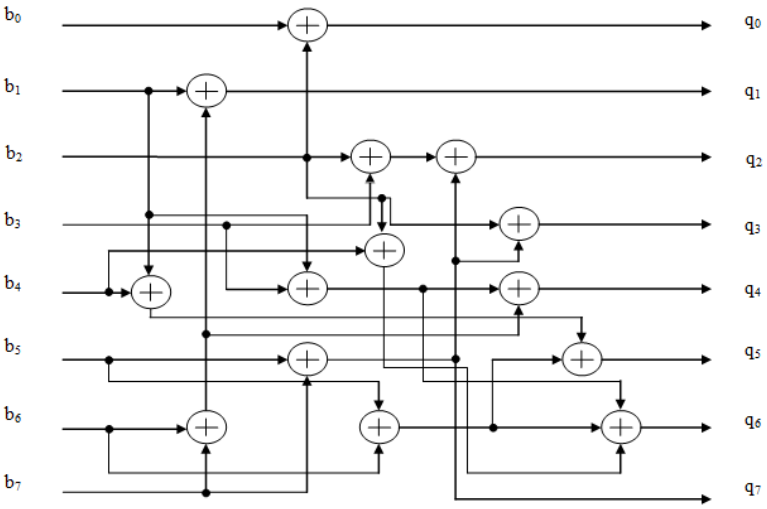

Fig. 9: Circuit diagram of proposed enhanced ISO technique

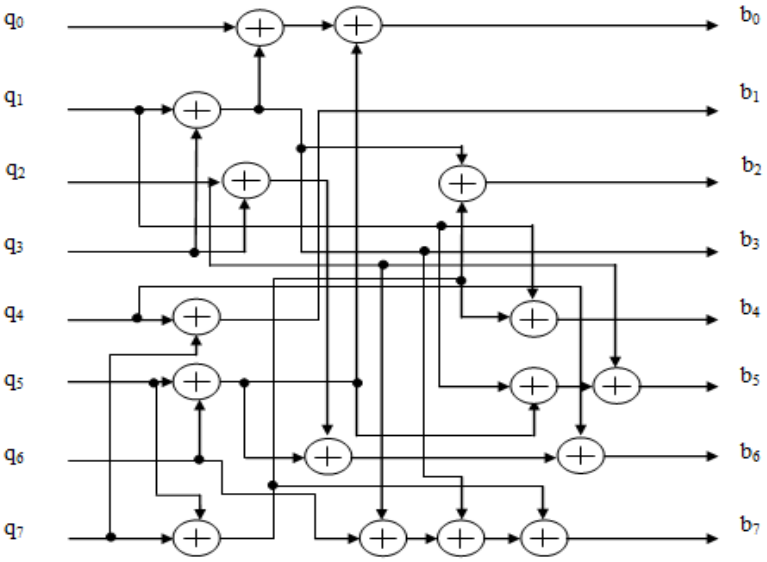

Fig. 10: Circuit diagram of proposed enhanced inverse ISO technique

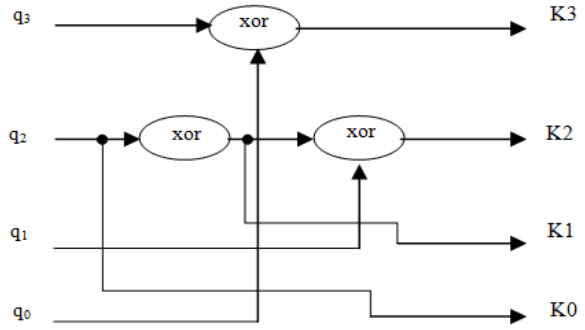

Fig. 11: Circuit diagram of proposed combined multiplication of xλ and x^2

Hence, equations of Inverse Isomorphic Transformation can be reduced as follows:

Inv ISO [0] = q_0 ^ Redundant Function_ISO1 ^ Redundant Function2
Inv ISO [1] = q_7 ^ q_5
Inv ISO [2] = Redundant Function_InvISO1 ^ Redundant Function_InvISO3
Inv ISO [3] = Redundant Function_InvISO1
Inv ISO [4] = q_1 ^ Redundant Function_InvISO3
Inv ISO [5] = q_2 ^ Redundant Function_InvISO1 ^ Redundant Function_InvISO2
Inv ISO [6] = q_2 ^ q_3 ^ q_4 ^ Redundant Function_InvISO2
Inv ISO [7] = q2 ^ q7 ^ Redundant Function _InvISO1 ^ Redundant Function_InvISO2

When compared to traditional Inv ISO technique, 8 gates are reduced in proposed Enhanced Inverse ISO technique. The circuit diagram of proposed Enhanced Inverse ISO technique is illustrated in Fig. 10. In addition, the circuits of multiplication of x λ and x^2 are realized and re-designed by eliminating the unwanted redundant functions. Hence a new architecture has been proposed for Inverse Multiplication unit. The circuit diagram of Enhanced combined multiplication of x λ and x is illustrated in Fig. 11 by using following expressions:

$$K3 = q0 \oplus q3 \tag{7}$$

$$K2 = q1 \oplus h \tag{8}$$

$$K1 = h \tag{9}$$

$$K0 = q2 \tag{10}$$

where,

$$h = q2 \oplus q3$$

RESULTS AND DISCUSSION

Design of Enhanced AT, IAT, ISO, Inverse ISO and combined multiplication of x λ and x^2 are designed

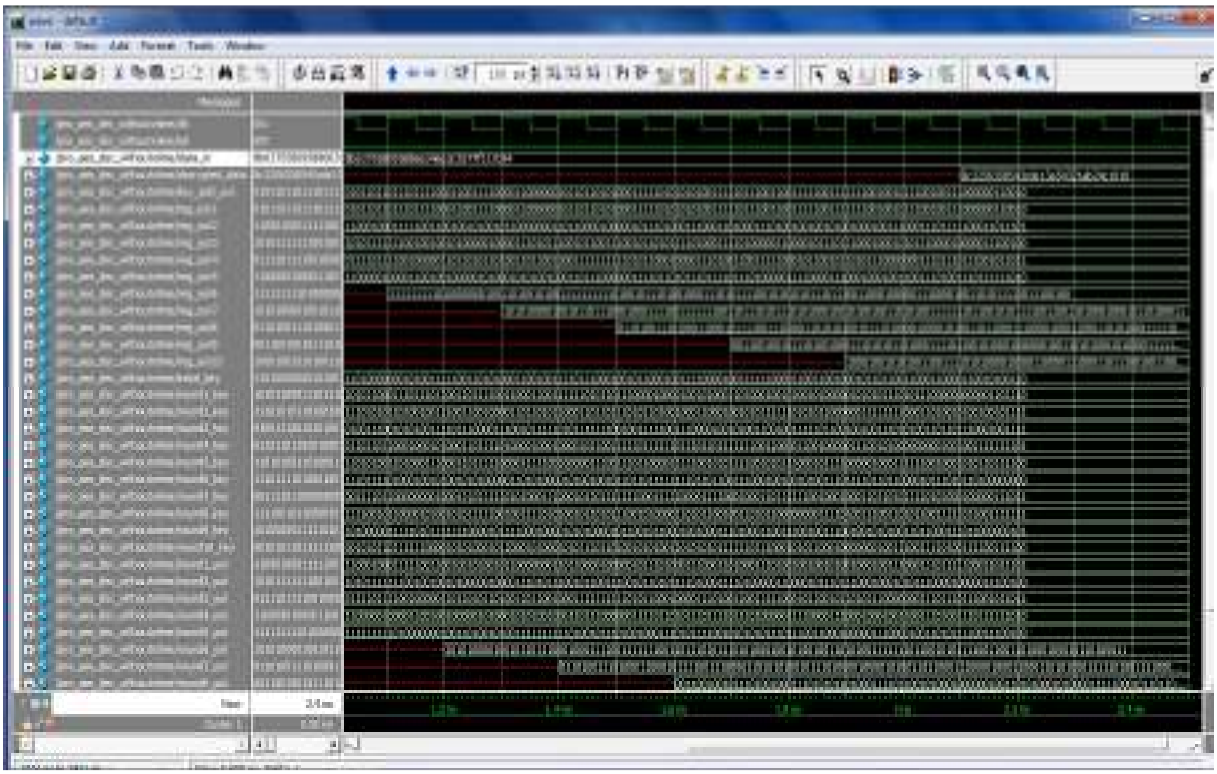

Fig. 12: Simulation result of proposed enhanced composite S-box based 128-bit AES encryption

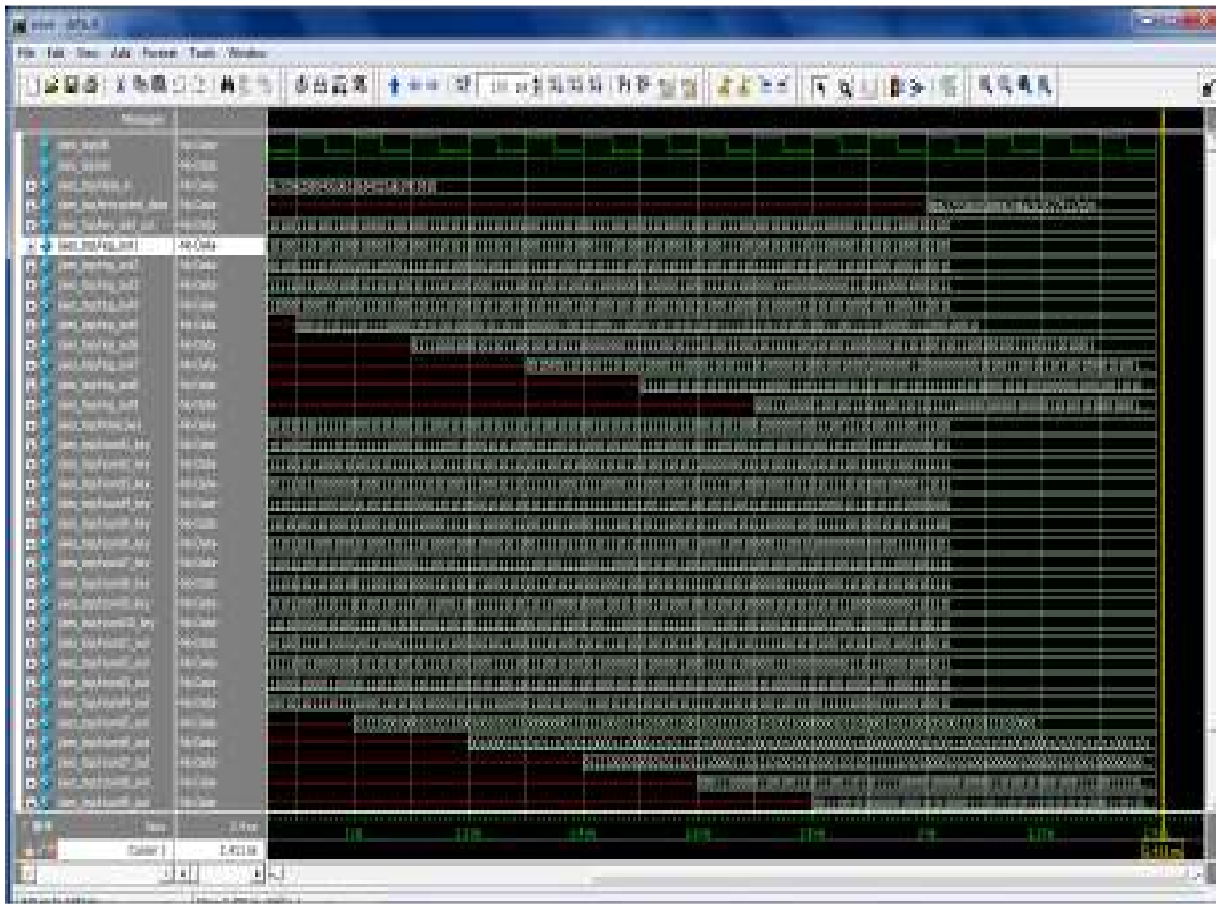

Fig. 13: Simulation result of proposed enhanced composite S-box based 128-bit AES decryption

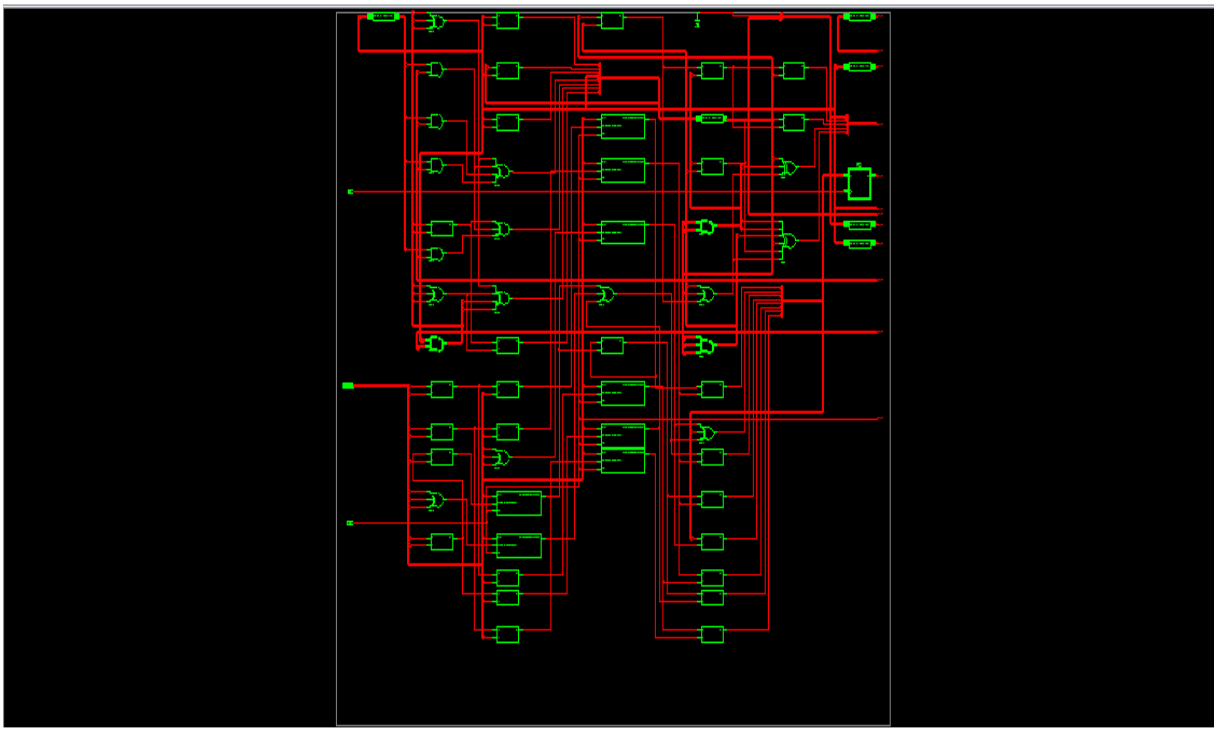

Fig. 14: RTL view of traditional composite S-box

through Verilog Hardware Description Language (Verilog HDL). The Simulation result of proposed Enhanced techniques are validated by using ModelSim 6.3C and Synthesis results are evaluated by using Xilinx 10.1i (Family: Virtex 4, Devices: Xc4vlx25, Package: FF668, Speed: -12) design tool. In this research work, 128 bit AES is considered for realizing the proposed Enhanced techniques of Composite S-Box. Simulation result of proposed Enhanced Composite S-Box based 128-bit AES Encryption is illustrated in Fig. 12. Figure 12, 128-bit input data is given as "8c3256358543cde13a54321ab24c1910". Encrypted output is obtained as "8b37558 09588 66346e3c3577f117c94" through proposed Enhanced Composite s-Box. Simulation result of proposed Enhanced Composite S-Box based AES Decryption is illustrated in Fig. 13. Again the original input data is obtained in decryption (reconstructed) output through proposed Enhanced techniques.

RTL View of traditional and proposed Composite S-Box is illustrated in Fig. 14 and 15 respectively. Performance Evaluation of both traditional Composite

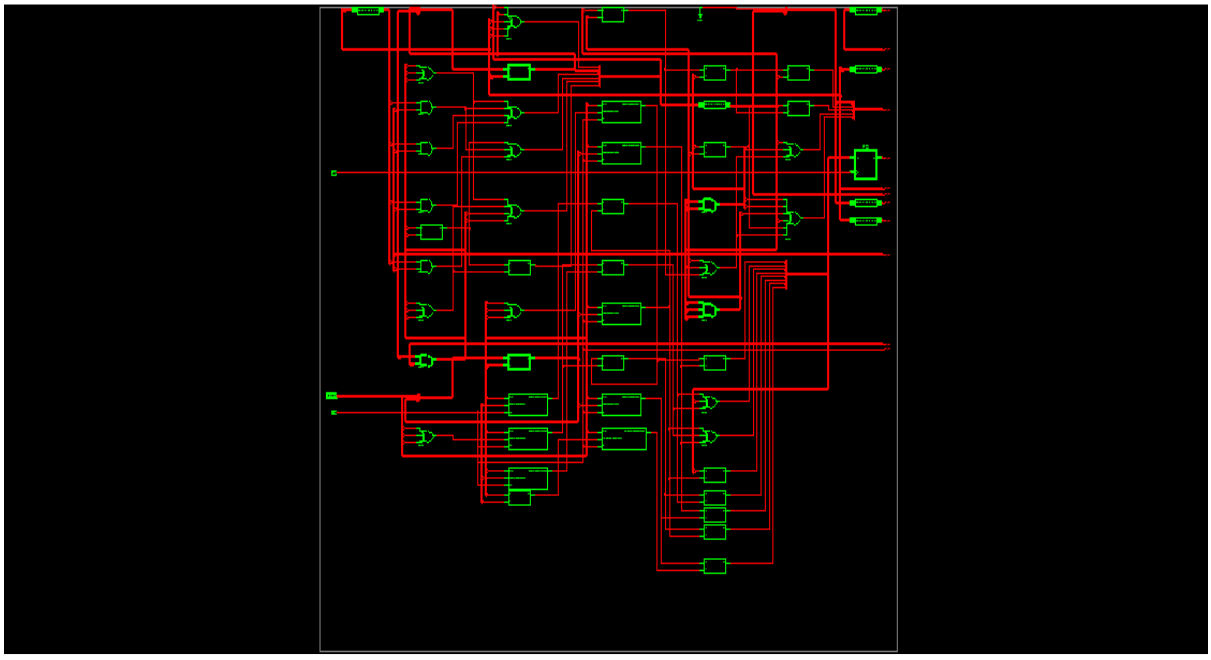

Fig. 15: RTL view of proposed composite s-box

Table 1: Performance evaluation of both traditional composite s-box and proposed composite s-box

Types/Parameters	Slices	LUTs	Delay (ns)	Power (mW)
Traditional Composite S-Box	46	88	12.255	601
Proposed Composite S-Box	43	83	11.980	564
Percentage reduction	6.52%	5.68%	2.24%	6.15%

Table 2: Performance evaluation of both traditional composite s-box based AES encryption and proposed enhanced composite s-box based AES encryption

Types/Parameters	Slices	LUTs	Minimum Period (ns)	Minimum input arrival time before clock (ns)	Frequency (MHz)	Power (mW)
Traditional Composite S-Box based AES Encryption	8928	16880	7.638	8.329	130.916	6.775
Proposed enhanced Composite S-Box based AES Encryption	7396	13862	7.197	7.970	138.943	5.980
Percentage reduction	17.15%	17.87%	5.77%	4.31%	5.77%	11.73%

Table 3: Performance evaluation of both traditional composite s-box based AES decryption and proposed enhanced composite s-box based AES decryption

Types/Parameters	Slices	LUTs	Minimum period (ns)	Minimum input arrival time before clock (ns)	Frequency (MHz)	Power (mW)
Traditional Composite S-Box based AES Decryption	10652	20421	7.268	7.669	137.592	7.171
Proposed enhanced Composite S-Box based AES Decryption	10266	18886	6.379	7.643	156.759	6.515
Percentage reduction	3.62%	7.51%	12.23%	Slight reduction	12.23%	9.14%

S-Box and Proposed Enhanced Composite S-box is analyzed and compared in Table 1. Similarly, performance evaluation of both traditional Composite S-Box based AES Encryption and Proposed Enhanced Composite S-Box based AES Encryption is analyzed and compared in Table 2. It shows that proposed enhanced techniques are used to increase the performances in terms of less area utilization, less delay and lower power consumption. Further Performance Evaluation of both traditional Composite S-Box based AES Decryption and proposed Composite S-Box based AES Decryption are analyzed and compared in Table 3.

CONCLUSION

In this study, Enhanced Composite S-Box is designed through Very Large Scale Integration (VLSI) System design environment. Less area utilization, high speed and lower power consumption are the main key factors in VLSI System design environment. Therefore, the main goal of this research work is to increase the speed of the AES Encryption and Decryption process. In order to meet above requirements, Enhanced Affine Transformation, Enhanced Inverse Affine Transformation, Enhanced Isomorphic Mapping, Enhanced Inverse Isomorphic Mapping and combined multiplication of $x \lambda$ and x^2 are proposed in this research work. Further all the enhanced techniques are integrated in Composite S-Box. Proposed Enhanced Composite S-box offers 6.52% reduction in Slices, 5.68% reduction in LUTs, 2.24% reduction in delay and 6.15% reduction in power consumption than traditional Composite S-Box. Proposed Enhanced Composite-s-Box based AES Encryption offers 17.15% reduction in Slices, 17.87% reduction in LUTs, 5.77% reduction in Minimum Period delay and 11.73% reduction in power consumption than traditional AES Encryption. Similarly, Proposed Enhanced Composite S-Box based AES Decryption offers 3.62% reduction in Slices, 7.51% reduction in LUTs, 12.23% reduction in Minimum period delay and 9.14% reduction in power consumption than traditional AES Decryption. In future, Proposed Enhanced Composite S-Box based AES Encryption and Decryption Standard will be absolutely used in different types of wireless cryptography based applications for implementing with less hardware and high speed.

REFERENCES

Ahmad, N. and S.M. Rezaul Hasan, 2013. Low-power compact composite field AES S-Box/Inv S-Box design in 65nm CMOS using Novel XOR Gate. Integration VLSI J., 46(4): 333-344.

Balamurugan, J. and E. Logashanmugam, 2015. Design of a high speed and area efficient optimized mixcolumn for AES. Int. J. Appl. Eng. Res., 10(17): 13003-13008.

Jamil, T., 2004. The Rijndael algorithm. IEEE Potentials, 23(2): 36-38.

Khose, P.N. and V.G. Raut, 2014. Hardware implementation of AES encryption and decryption for low area and power consumption. Int. J. Res. Eng. Technol. (IJRET), 3(5): 480-484.

Li, H., 2006. Efficient and flexible architecture for AES. IEE P-Circ. Dev. Syst., 153(6): 533-538.

Li, H. and J. Li, 2008. A new compact dual-core architecture for AES encryption and decryption. Can. J. Elect. Comput. E., 33(3/4): 209-213.

Liu, Q., Z. Xu and Y. Yuan, 2015. High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion. IET Comput. Digit. Tec., 3: 175-184.

Mozaffari-Kermani, M. and A. Reyhani-Masoleh, 2010. Concurrent structure-independent fault detection schemes for the advanced encryption standard. IEEE T. Comput., 59(5): 608-622.

Sandhya, M. and S. Deepa, 2013. A high throughput CFA AES S-box with error correction capability. IOSR J. Electr. Electron. Eng. (IOSR JEEE), 5(5): 47-56.

Sandhyarani, K. and P. Nirmal Kumar, 2014a. Incorporation of composite field S-box into AES-CBC and AES-CM modes to avoid SEUs. Res. J. Appl. Sci. Eng. Technol. (RJASET), 8(12): 1424-1428.

Sandyarani, K. and P. Nirmal Kumar, 2014b. Design of high speed AES-128 using novel mix column transformation and sub bytes. J. Comput. Appl. (JCA), 7(2): 57-60.

Sklavos, N. and O. Koufopavlou, 2002. Architectures and VLSI implementations of the AES-proposal Rijndael. IEEE T. Comput., 51(12): 1454-1459.

Thillaikkarasi, R. and K. Vaishnavi, 2014. Optimum composite field s-boxes aimed at AES. Int. J. Adv. Res. Electron. Commun. Eng. (IJARECE), 3(3): 1-5.

Wang, M.Y., C.P. Su, C.L. Horng, C.W. Wu and C.T. Huang, 2010. Single-and multi-core configurable AES architectures for flexible security. IEEE T. VLSI Syst., 18(4): 541-552.