Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

Compact AES S-Box using Gate Diffusion Input logic

[1]G.Prathipa,[2]K.Muthumeena
[1]Assistant Professor (ECE Dept), [2]PG Student (M.E-VLSI Design)
Karpaga Vinayaga College of Engineering and Technology
G.S.T Road, Mathuranthagam, Kancheepuram District, Tamil Nadu, India

Abstract— The need for security has been increasing day by day. The Advance Encryption Standard (AES) Algorithm is used for encryption and decryption purpose and it is one of the most popular Algorithm used in symmetric key cryptography. The AES specifies a FIPS approved cryptographic Algorithm that can be used to protect electronic data. The existing algorithm uses more number of transistors to implement the S-Box and INV S-Box. The heart of the S-Box is Multiplicative inverse unit. It requires eight XOR gates for Squaring Operation in GF (2^4) and Multiplication with Constant block and there are two XOR gates and one AND gate in the critical path of Multiplicative in GF (2^2) block. CMOS logic is used to design the logic gates and mux. The Squaring Operation in GF (2^4) and Multiplication with Constant block requires 48T and Multiplicative in GF (2^2) block requires 66T. This Architecture of Multiplicative inverse unit increases the area, power, critical path and transistor. In the proposed work, the compact S-Box has been designed by using reduced Multiplicative inverse unit.GDI (Gate Diffusion Input) logic is used to design the S-Box. It will be designed by reducing the number of transistors in S-Box. The TANNER EDA is used to design the proposed S-Box.

Keywords— *S-box, Composite field arithmetic, AES encryption, GDI*

I. INTRODUCTION

The worldwide communication of private and confidential data over the computer networks or the Internet, there is always a possibility of threat to data confidentiality, data integrity and also data availability. Data encryption maintains data confidentiality, integrity and authentication. Messages need to be secured from unauthorized party. Encipherment is one of the security mechanisms to protect information from public access. There were many cryptographic algorithms proposed such as Data Encryption Standard (DES), 2-DES, 3-DES, Elliptic Curve Cryptography (ECC), the Advanced Encryption Standard (AES) and other algorithms. Many investigators and hackers are always trying to break these algorithms using brute force and side channel attacks. Some attacks were successful as it was the case for the Data Encryption Standard (DES) in 1993 [1]. The Advanced Encryption Standard (AES) is considered as one of the strongest published cryptographic algorithms.

National Institute of Standards and Technology (NIST) adopted Advanced Encryption Standard (AES) as the standard for encryption and decryption of blocks of data after the failing of the Data Encryption Standard (DES). The draft is published under the name as FIPS-197 (Federal Information Processing Standard number 197) [4]. Moreover, it is used in many applications such as in ATM Machines, RFID cards, cell phones and large servers.

AES is widely used for encryption of audio/video data contents in real time. Due to the significance of the AES algorithm and the numerous real –time applications, the main concern of this thesis will be presenting new efficient hardware implementations for this algorithm. Audio/video content encryption is required in real-time in business deals via video conferencing. Therefore, dedicated hardware implementation is inevitable in such applications.

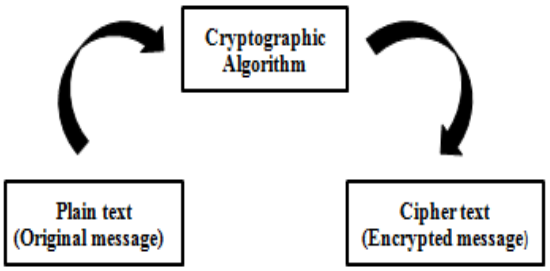
Hardware implementation can be done through different architectures trading throughput with area and power consumption. At any time, designing best architecture for a particular design with low area and low latency is a challenge. Hardware implementations of the AES algorithm [2] vary according to the application. While some applications require very high throughputs as in e-commerce servers, others require a medium throughput range as in designs for cell phones. Some others require very low area and low power implementations to be used application as RFID cards.

The AES is an iterative algorithm and uses four operations in different rounds, namely Sub Bytes, Shift Rows, Mix Columns and Key Additions transformations. Sub Bytes transformation (also called substitution) is a non-linear operation in AES wherein each byte of a state is mapped to a different value. The Sub Bytes transformation is done through S-box and it is the most complex steps in terms of cost and implementation. Use of ROM table and composite field arithmetic are two techniques to perform substitutions. The ROM based approach requires high amount of memory and also it causes low latency and low throughput because of ROM access time. Therefore, composite field arithmetic is more suitable for S-box (substitution) implementation its hardware optimization for VLSI implementation is very important to reduce the area and power of the AES architecture.

1.2 History of Cryptography

Information need to be secured from unauthorized party. Cryptography is one of the security mechanisms to protect information from public access. Cryptography is a Greek origin word which means “secret writing” to make the information secure and immune to attacks. Classic cryptography was used for top-secret communications between people. This kind of cryptography is commonly applied by replacing the message letters with other letters using definite formula. Nowadays its changed into algorithm based cryptography according to demand by the users. It has two processes encryption and decryption, in the first process encryption; the Plain text (Original message) will be converted into secured text or Cipher text (Encrypted message) using a specific algorithm which is

shown in Fig.1. The second process is decryption which is



the reverse process of encryption; here Cipher text will be converted into Plain text using all the inverse steps applied for encryption [1].

Figure.1 Basic step of Encryption in cryptography

1.3 Galois Field

The Galois field (GF) or Finite Field with a finite number of elements are extensively used in cryptography [3]. The total number of element present in GF is called the order of fields. A GF is of the form p^n, where n is a positive integer and p is a prime number also called the characteristic of the Galois field. There are many cryptographic algorithms using GF among them, the AES algorithm uses the GF (2^8). The data byte can be represented using a polynomial representation of GF (2^8). The polynomial representation of data bytes in Finite Fields as shows in equation 1.

$$a(x) = b_7 x^7+b_6 x^6+b_5 x^5+b_4 x^4+b_3 x^3+b_2 x^2+b_1 x^1+b_0 \quad (1)$$

Arithmetic operation is totally different from normal arithmetic algebra; an addition can be found using bit-wise XOR operation. In Galois field [3], the multiplication product of polynomials will be modulo an irreducible polynomial so final answer can be within the used finite field. The polynomial which cannot be factorized of two or more than two is called as irreducible polynomial.

4.1 GDI *(Gate Diffusion Input) logic*

The GDI cell is similar to a CMOS inverter structure. In a CMOS inverter the source of the PMOS is connected to VDD and the source of NMOS is grounded. But in a GDI cell this might not necessarily occur. There are some important differences between the two [10]. The three inputs in GDI are namely-

1) G- common inputs to the gate of NMOS and PMOS
2) N- input to the source/drain of NMOS
3) P- input to the source/drain of PMOS

Bulks of both NMOS and PMOS are connected to N or P (respectively) that is it can be arbitrarily biased unlike in CMOS inverter. Moreover, the most important difference between CMOS and GDI is that in GDI N, P and G terminals could be given a supply 'VDD' or can be grounded or can be supplied with input signal depending upon the circuit to be designed and hence effectively minimizing the number of transistors used in case of most logic circuits (eg. AND, OR, XOR, MUX, etc). As the allotment of supply and ground to PMOS and NMOS is not fixed in case of GDI, therefore, problem of low voltage swing arises in case of GDI which is a drawback and hence finds difficulty in case of implementation of analog circuits.

The concept of Shannon thermo could be applied with ease to design a basic GDI cell. In Shannon expansion thermo any function F can be written as:

$$F = x_1 H(x_2...x_n)+g(x2...x_n) \quad (2)$$

That is a larger function can be broken down into smaller function as shown in equation 3.

$$X1=F(1,x_2...x_n)+F(0,x_2...x_n) \quad (3)$$

Then , the smaller functions could be further broken down if possible till the time it is not further reducible. The output function of a basic GDI cell (where A, B, and C are input to GP, and, N respectively).

II. EXISTING S-BOX ARCHITECTURE

The existing algorithm uses more number of logic gates to implement the S-Box and INV S-Box. The heart of the S-Box is Multiplicative inverse unit. It requires eight XOR gates for X^2 and Xλ block and there are two XOR gates and one AND gate in the critical path of Multiplicative in GF (2^2) block. This Architecture of Multiplicative inverse unit increases the area, power and critical path

2.1 Composite Field Arithmetic S-Box

The Sub Bytes transformation, done through S-BOX mapping is computationally inefficient when implemented using a ROM. But, it is not efficient for applications requiring very high throughput [2] as ROM accessing involves one complete clock cycle for mapping one 8-bits state element and consequently 16 clock cycles are required to transform the 128 bits data (16 bytes). To increase the throughput, parallel ROMs are required resulting in large size of chip area. Therefore, a more feasible solution is to implement an S-box is by using composite field arithmetic which uses only logic elements in the implementation. The S-BOX substitution starts by finding the multiplicative inverse of the data in GF (2^8), and then applying the affine transformation. Fig.2 shows steps for the one byte forward and inverse Sub Bytes transformation using composite field arithmetic.

To find the S-BOX transformation first multiplicative inverse of GF (2^8) then affine transformation calculated. Similarly, for Inv Sub Bytes first Inv Affine transformation then multiplicative inverse has to be calculated. There is one major operation involved here, which is to find the multiplicative inverse in GF (2^8). This can be done by breaking the GF (2^8) elements in GF (2^4) and some more logical blocks. i.e., any arbitrary polynomial [6] in GF (2^8) can be represented as bx+c using an irreducible polynomial x^2+Ax+B. Here, b is the most significant nibble and c is the least significant nibble. The multiplicative inverse can be found by using the following equation 4.

$$(b x + c)^{-1}$$
$$= b (b^2 B +bcA +c^2)^{-1} x +(c +bA) (b^2 B +bcA +c^2)^{-1}$$
$$=b(b^2\lambda +c(b+c))^{-1}x+(c +b)(b^2\lambda +c(b+ c))^{-1} \quad (4)$$

Where, A=1, B=λ, as the irreducible polynomial used is $x^2+x+\lambda$. Fig.3.2 shows the block diagram to find the multiplicative inverse in GF (2^8) using GF (2^4).

Below the Fig.4 shows the meanings the symbols used in Fig.3. The mapping structure in different fields along with the irreducible polynomials is as follows.

$$GF(22) \rightarrow GF(2): x^2 + x + 1$$
$$GF((22)2) \rightarrow GF(22): x^2 + x + \varphi$$
$$GF(((22)2)2) \rightarrow GF((22)2): x^2 + x + \lambda \quad (5)$$

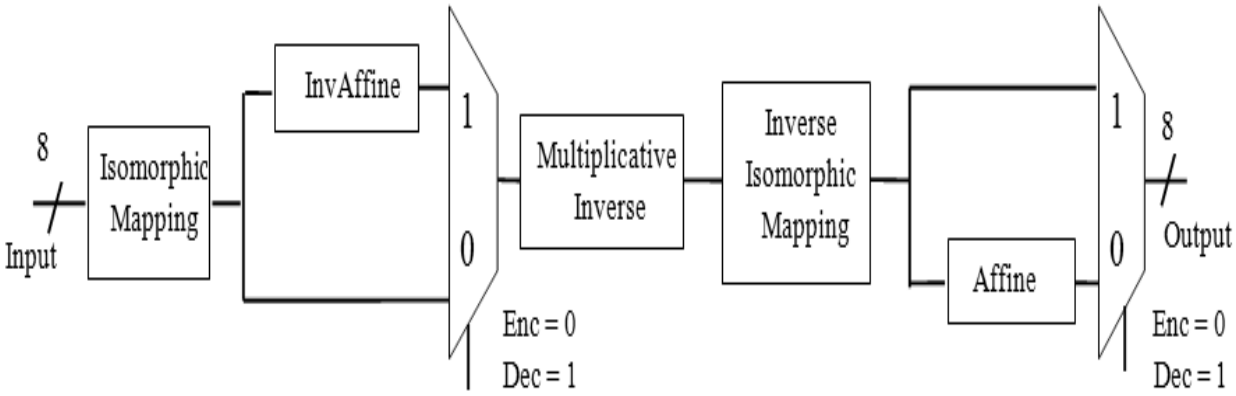


Figure.2 Block diagram of Shared architecture for S-Box & Inverse S-Box.

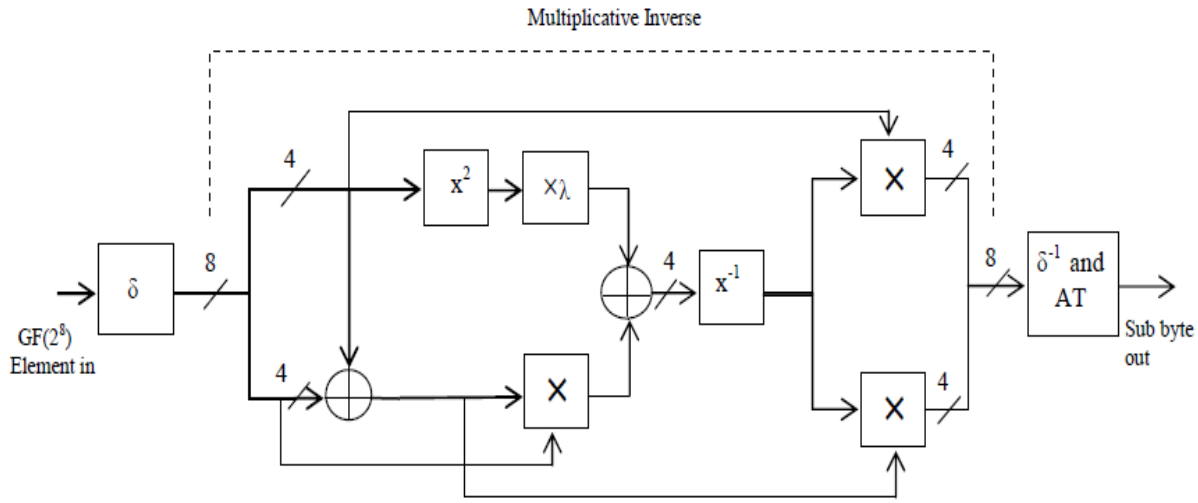


Figure.3 The conventional S-box architecture using composite field arithmetic

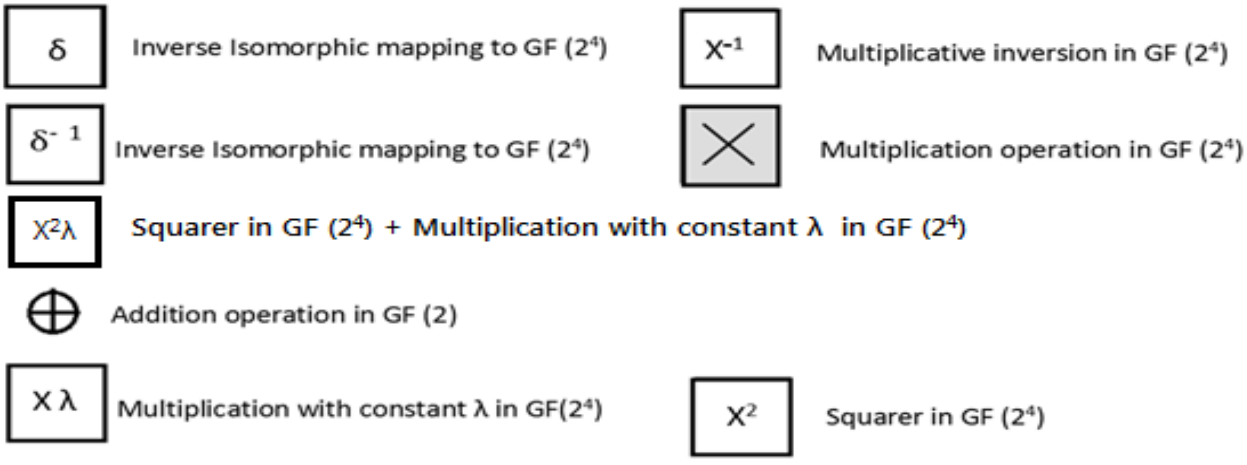


Figure.4 Meaning of the symbol used in Fig.2

2.2 Squaring Operation in GF (2^4)

The squaring operation of 4 bits. The final output bits h in the form of input bits q can be calculated using irreducible polynomial, which represented in equation 6. There are four XOR gate [6] is required to implement the Squaring operation in GF (2^4).

$$h_3 = q_3$$
$$h_2 = q_3 \oplus q_2$$
$$h_1 = q_1 \oplus q_2$$
$$h_0 = q_0 \oplus q_1 \oplus q_3 \quad (6)$$

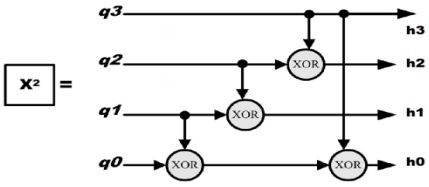


Figure.5 Squarer for GF (2^4)

2.3 Multiplication with Constant, λ

The final output bits k in the form of input bits h can be calculated using irreducible polynomial, which represented in equation 7. There are total four XOR gate [6] is required to implement the multiplication with λ.

$$k_3 = h_3 \oplus h_1 \oplus h_2 \oplus h_3$$
$$k_2 = h_1 \oplus h_3$$
$$k_1 = h_2$$
$$k_0 = h_2 \oplus h_3 \quad (7)$$

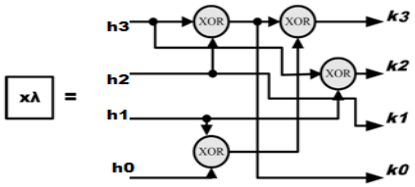


Figure.6 Multiplication with constant, λ

2.4 Galois Field GF (2^2) Multiplication

The Galois field (2^2) multiplier is the major component in GF (2^4) multiplication. It also implemented using combinational logic which presents in equation 8.

$$k_1 = q_1w_1 \oplus q_0w_1 \oplus q_1w_0$$
$$k_0 = q_1w_1 \oplus q_0w_1 \quad (8)$$

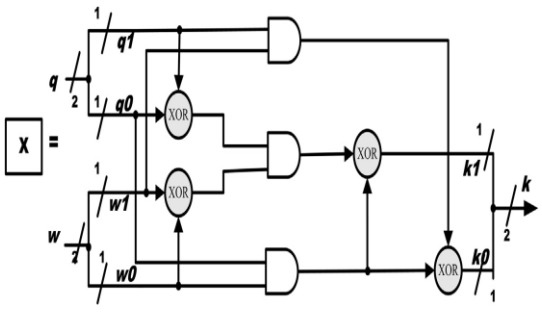


Figure.7 GF (2^2) Multiplication

2.5 Existing logic gates and MUX

The CMOS logic is used to design the XOR and AND gate.12T is needed to design the XOR gate.6T is needed to design the AND gate.

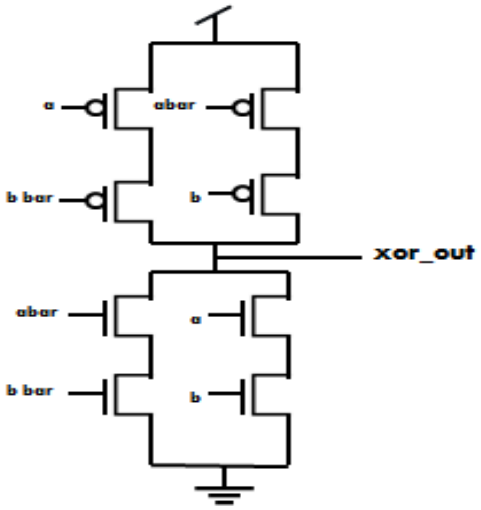


Figure.8 XOR gate

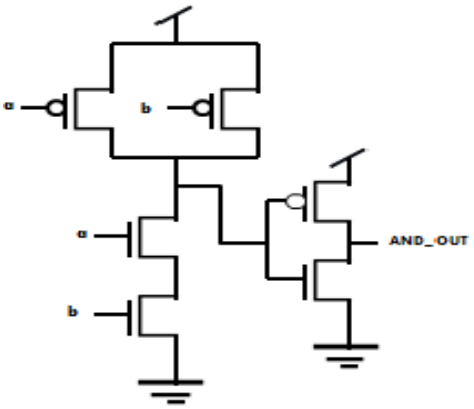


Figure.9 AND gate

The CMOS logic is used to design the mux.6T is needed to design the MUX.

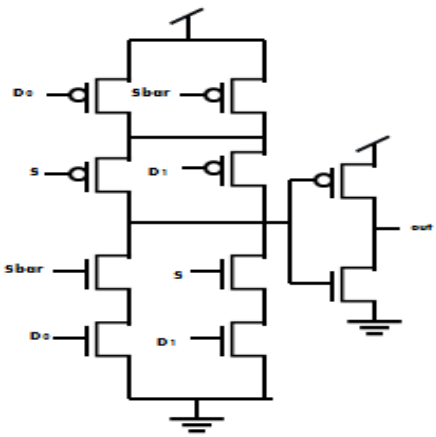


Figure.10 2:1 MUX

Three 2: I MUX is used to design the 4:1 MUX.18T is needed to design the 4:1MUX.

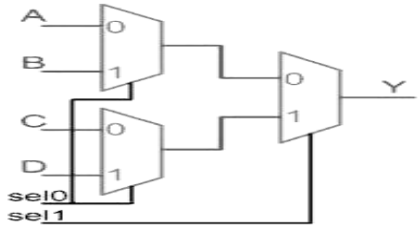


Figure.11 4:1 MUX

III. PROPOSED SYSTEM

In the proposed work, the compact S-Box has been designed by using reduced Multiplicative inverse unit. It will be designed by reducing the number of XOR gates in X^2 and Xλ block and replaces XOR and AND gate in the critical path of Multiplicative in GF (2^2) with MUX. The logic gates and MUX are design by using GDI logic. The TANNER EDA is used to design the proposed S-Box.

3.1. Reduction of XOR Gate

The $X^2\lambda$ block design after merging of blocks like squarer, multiplication with constant λ. There are three XOR gates needed to design the $X^2\lambda$ block.

Substitute equation (6) in equation (7) then the $X^2\lambda$ block output equation can be written as in equation (9)

$$k_3 = q_0 \oplus q_3$$
$$k_2 = q_1 \oplus m$$
$$k_1 = m$$
$$k_0 = q_2$$
$$m = q_3 \oplus q_2 \quad (9)$$

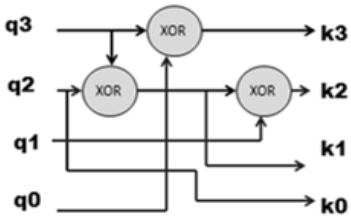


Figure.12 Proposed $X^2\lambda$ block

3.2 Reduced Critical Path of Multiplication in GF (2^2)

The conventional architecture of multiplication in GF (2^2), there is two XOR gates and one AND gate in the critical path. The output equation can be written as in equation (10)

$$k_1 = q_1w_1 \oplus q_0w_1 \oplus q_1w_0$$
$$k_0 = q_1w_1 \oplus q_0w_1 \quad (10)$$

The above equation can be implemented using two 4:1 parallel multiplexers as follows. Suppose y is the select line. Then, for different values of y, multiplication result z, from equation (10), will have values as given in Table.1.

Table 1: Result of multiplication in GF (2^2)

Value of y	z(0)	z(1)
00	'0'	'0'
01	X(0)	X(1)
10	X(1)	X(0) xor X(1)
11	X(0) xor X(1)	X(0)

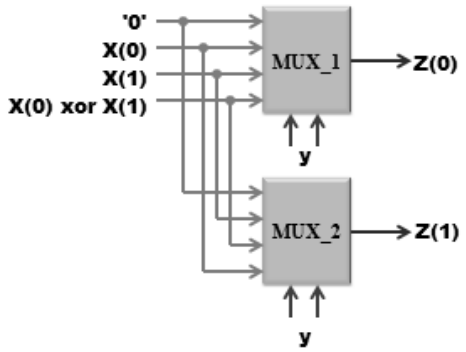


Figure.13 Proposed Multiplication in GF (2^2)

3.3 Proposed logic gates and MUX

The GDI logic is used to design the XOR and AND gate.This logic is used to reduce the transistor [10]. Only 4T is needed to design the XOR gate and 2T is needed to design the AND gate.

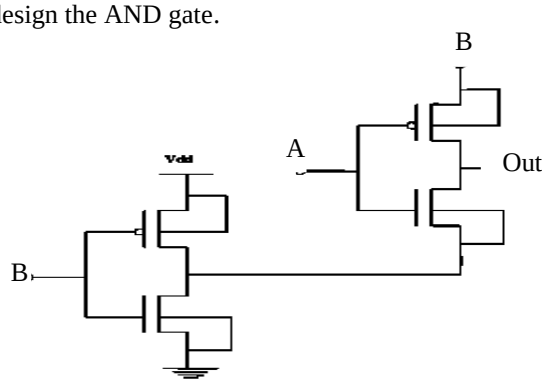


Figure.14 GDI logic XOR gate

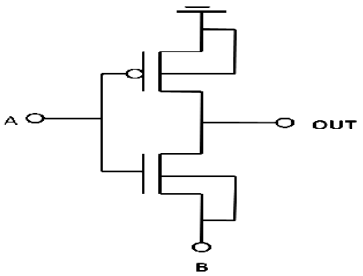


Figure.15 GDI logic AND gate

Two transistors are needed to design the 2:1 MUX. Only 4T is needed to design the 4:1 MUX.

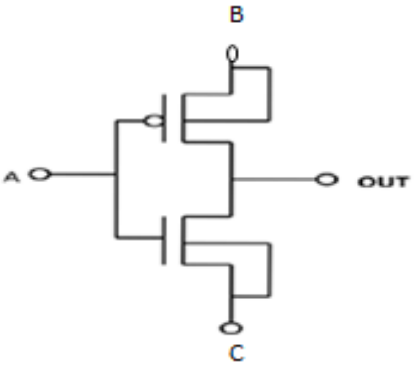


Figure.16 GDI logic 2:1 MUX

IV. RESULT ANALYSIS

The Existing and Proposed architecture of S-Box is implemented by TANNER EDA.

Table 2: Transistor count

	CMOS Logic	GDI Logic
XOR	12T	4T
AND	6T	2T
MUX	12T	2T

Table 3.: Power

	CMOS Logic	GDI Logic
XOR	$3.72e^{-4}$	$1.19e^{-4}$
AND	$1.61e^{-4}$	$0.00265e^{-4}$
MUX	$3.09e^{-3}$	$2.48e^{-11}$

The optimized schematic $X^2\lambda$ block has drawn in the TANNER EDA. The area i.e. number of transistor in the $X^2\lambda$ Block is 12T

Table 4: Power and Transistor count for $X^2\lambda$ Block

	CMOS Logic $X^2\lambda$ Block	GDI Logic $X^2\lambda$ Block
TRANSISTOR COUNT	36T	12T
POWER	$9.62e^{-4}$	$3.37e^{-4}$

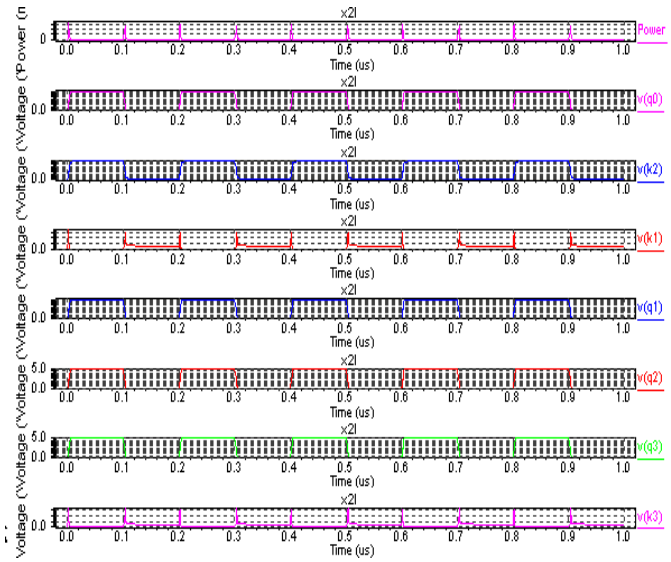


Figure.17 $X^2\lambda$ block output waveform

The optimized schematic X block has drawn in the TANNER EDA. The area i.e. number of transistor in the X block is 16T

Table 5: Power and Transistor count for X Block

	CMOS Logic X Block	GDI Logic X Block
TRANSISTOR COUNT	84T	16T
POWER	$17.3e^{-4}$	$1.23e^{-4}$

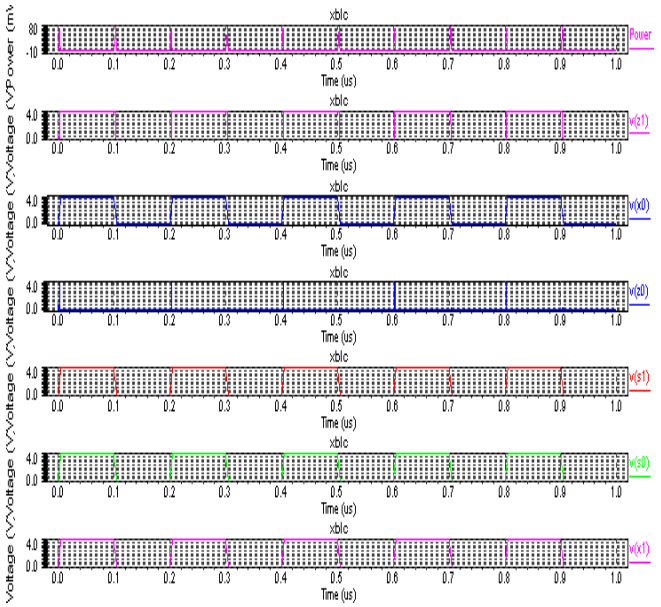


Figure.18 X block output waveform

V. CONCLUSION

An optimized architecture of S-box for AES encryption is proposed in this paper. This novel architecture is implemented in TANNER EDA. The proposed implementation is low power and used fewer transistors compared to existing architecture of AES S-Box. It is suitable for the implementation in small devices like, smart cards, cellular phones.

REFERENCE

[1] B.A. Forouzan and D. Mukhopadhyay, Cryptography and Network Security, 2nd Ed., Tata McGraw Hill, New Delhi, 2012.

[2] Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, and Cheng-Wen Wu, "A High-Throughput Low-Cost AES Processor," *IEEE Communications Magazine*, vol.41 (12), pp.86-91,Dec. 2003

[3] Edwin NC Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic", Custom R&D Engineer TexcoEnterprise Pvt.Ltd

[4] Federal Information Processing Standards Publication197 (FIPS197), available online, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[5] I. Hammad, K. E. Sankary and E. E. Masry, "High-Speed AES Encryptor With Merging Techniques," *IEEE Embedded Systems Letters*, Vol.2 (3), pp.67-71, Sept. 2010

[6] N. Ahmad, S. M. R. Hasan, "Low-power compact composite field AES S-Box/Inv S-Box design in 65 nm CMOS using Novel XOR Gate," *Integration, the VLSI Journal*, Article in Press.

[7] J. V. Dyken, J. G. Delgado-Frias, "FPGA schemes for minimizing the power-throughput trade-off in executing the Advanced Encryption Standard algorithm," *Journal of Systems Architecture*, Vol.56(2–3), pp. 116-123, Mar. 2010.

[8] L. Ali, I. Aris, F. S. Hossain and N. Roy, "Design of an ultra high speed AES processor for next generation IT security," *Computers and Electrical Engineering*, Vol.37 (6), pp.1160-1170, Nov. 2011

[9] M. I. Soliman, G. Y. Abozaid, "FPGA implementation and performance evaluation of a high throughput crypto coprocessor," *Journal of Parallel and Distributed Computing*, Aug. 2011.

[10]HT Bui, AK AL-Sheraidah, Y. Wang, New 4-transistor XOR and XNOR design in *Proc, 2nd IEEE Asia Pacific conf.ASICs*,pp.25-28(2000)

[11] N. Ahmad, R.Hasan, W.M.Jubadi, "Design of AES S-Box using combinational logic optimization," *IEEE Symposium on Industrial Electronics & Applications*, pp.696-699, Oct.2010.

[12]Arkadiy, Morgenshtein, Viacheslay, Yuzhaninov, Alexey, Koyshilovsky, Alexander Fish, "Full-Swing Gate Diffusion Input logic---Case-study of low-power CLA adder design" INTEGRATION, the VLSI journal 2013.