IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

Performance Analysis of S-Box using GF Arithmetic for AES

Akanksha Shukla
akanksha_engg2005@yahoo.co.in

Satyendra Sharma
satyendracommn@gmail.com

Abstract — **Sub-Byte transform or Substitution Box, better known as S-BOX is the most important part of Advance Encryption Standard (AES) algorithm. Many S-BOX architectures have been proposed in past based on Area consumption, Power Consumption, Speed etc. In this paper we present an optimized S-BOX architecture based on the Galois Field (GF) Operations. Proposed architecture is implemented in VHDL Using Xilinx ISE 9.2i on device XQ6VLX130T of Spartan family.**

Keywords — **S-Box, AES, Galois Field, VHDL, Spartan.**

I. INTRODUCTION

Advanced Encryption Standard (AES) is one of the secret key algorithms used in Cryptography. It is applied in a variety of applications including smart cards, internet web servers, automated teller machines (ATMs), etc. Both hardware and software implementations are taken into consideration while addressing AES algorithms. In addition to achieving standard requirements, hardware realization provides better security than the software realization while selecting the AES algorithm.

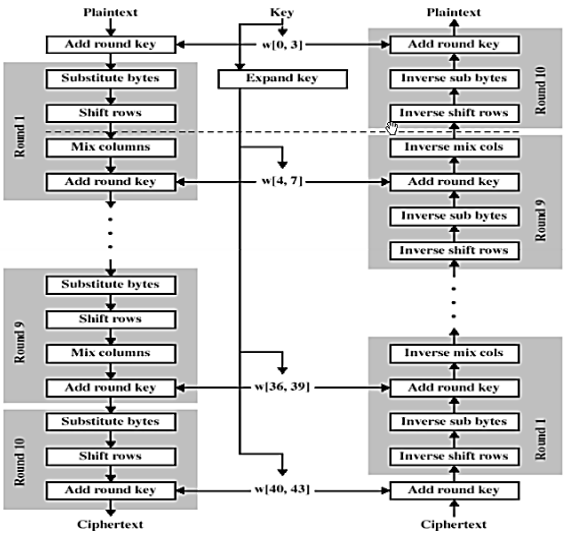
Since its announcement in 2001, the Advanced Encryption Standard (AES) has become a widely known and relied upon block cipher. It has been used for countless different applications ranging in size and scale from internet banking operations performed on large web servers to private communications between a wireless smart card and its reader.

On January 2, 1997, The National Institute of Standards and Technology (NIST) published a request for comments for the "Development of a Federal Information Processing Standard for Advanced Encryption Standard." NIST sought to "consider alternatives that offer a higher level of security" than that offered by the Data Encryption Standard (DES), which grew vulnerable to brute-force attacks due to its 56-bit effective key length [1]. AES candidates were required to support a symmetric block cipher that supported multiple key lengths.

II. AES ALGORITHM

The Advanced Encryption Standard (AES) also called the Rijndael algorithm, specifies a FIPS (Federal Information Processing Standards Publications) approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called cipher-text; decrypting the cipher-text converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.

A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key. This description of the AES algorithm therefore describes this particular implementation.



(a) Encryption (b) Decryption

Figure 1: Overflow of the AES algorithm

The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

counterpart in the encryption algorithm. The four stages are as follows:

1. *Substitute bytes*
2. *Shift rows*
3. *Mix Columns*
4. *Add Round Key*

The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

1. *Inverse Shift rows*
2. *Inverse Substitute bytes*
3. *Inverse Add Round Key*
4. *Inverse Mix Columns*

Again, the tenth round simply leaves out the Inverse Mix Columns stage. Each of these stages will now be considered in more detail.

The input is a single 128 bit block both for decryption and encryption and is known as the in matrix. This block is copied into a state array which is modified at each stage of the algorithm and then copied to an output matrix (see figure 1). Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words (the w matrix). It must be noted that the ordering of bytes within the in matrix is by column. The same applies to the w matrix.

Among the four transformations of AES, the implementation of SubBytes transformation for encryption and InvSubBytes for decryption decide the speed of operation and the area requirement.

III. Proposed S-box Design Method

This section illustrates the steps involved in constructing the multiplicative inverse module for the S-Box using composite field arithmetic. Since both the SubByte and InvSubByte transformation are similar other than their operations which involve the Affine Transformation and its inverse, therefore only the implementation of the SubByte operation will be discussed in this paper. The multiplicative inverse computation will first be covered and the affine transformation will then follow to complete the methodology involved for constructing the S-Box for the SubByte operation [2]. For the InvSubByte operation, we can reuse multiplicative inversion module and combine it with the Inverse Affine Transformation, as shown in Figure 2.

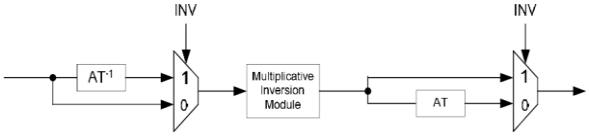


Figure 2: Combined SubByte and InvSubByte sharing a common multiplicative inversion module

The individual bits in a byte representing a GF (2^8) element can be viewed as coefficients to each power term in the GF (2^8) polynomial. For instance, $\{10001011\}_2$ is representing the polynomial $q^7 + q^3 + q + 1$ in GF (2^8). From [3], it is stated that any arbitrary polynomial can be represented as $bx + c$, given an irreducible polynomial of $x^2 + Ax + B$. Thus, element in GF (2^8) may be represented as $bx + c$ where b is the most significant nibble while c is the least significant nibble. From here, the multiplicative inverse can be computed using the equation below [3].

$$(bx + c)^{-1} = b(b^2B + bcA + c^2)^{-1}x + (c + bA)(b^2B + bcA + c^2)^{-1}$$

From [4], the irreducible polynomial that was selected was $x^2 + x + \lambda$. Since $A = 1$ and $B = \lambda$, then the equation could be simplified to the form as shown below.

$$(bx + c)^{-1} = b(b^2\lambda + c(b + c)^{-1}x + (c + b)(b^2\lambda + c(b + c)^{-1}$$

The above equation indicates that there are multiply, addition, squaring and multiplication inversion in GF (2^4) operations in Galois Field. Each of these operators can be transformed into individual blocks when constructing the circuit for computing the multiplicative inverse. From this simplified equation, the multiplicative inverse circuit GF (2^8) can be produced as shown in Figure 3.

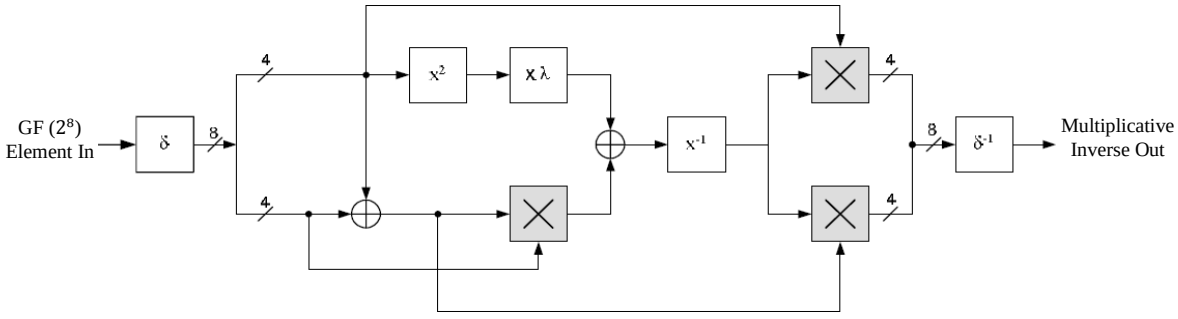


Figure 3: S-Box Architecture

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 1, Issue 6, January 2013)

IV. Simulation and Results

Here we show the FPGA implementation and results of given architectures for S-box implemented on Xilinx XQ6VLX130T device.

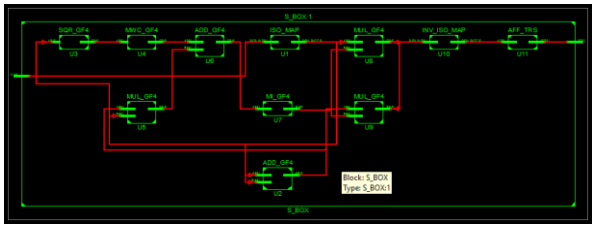


Figure 4: Main Module of S-Box

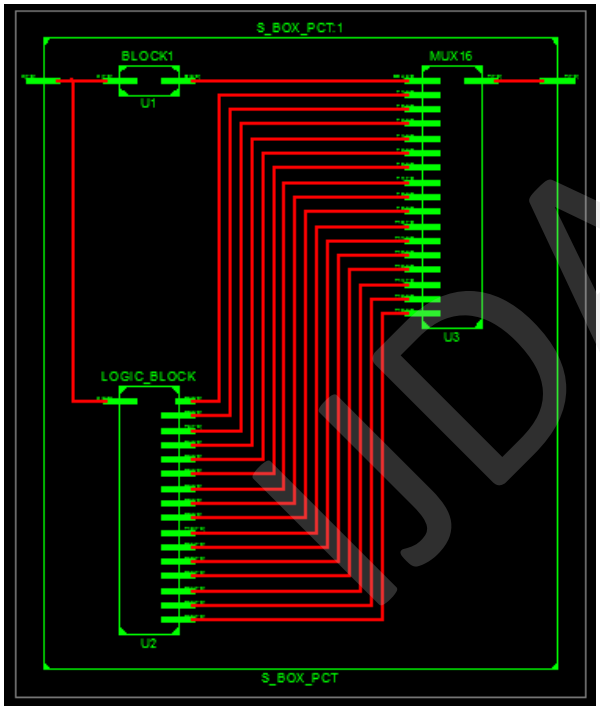


Figure 5: S-Box with PCT (Pre Computation Technique)

Xilinx ISE 9.2 is used to synthesize the design and provide post placement timing results. Table below showing the area consumed by various s-box architectures:

Table 1: Comparison of Various S-box architectures

SBOX	SLICES	GATE COUNT
ROM-based	0	66739
PCT-BASED	179	2700
GF- BASED	55	1489

V. Conclusions

This paper provides a novel approach for area optimized architecture of S-box used in AES encryption. This approach will lead to generate more secure block ciphers, solve the problem of the fixed structure S-boxes, and will increase the security level of the AES block cipher system. This architecture allows the S-Box to be used in both area-limited and demanding throughput AES chips for various applications, which range from small smart cards to high speed servers.

References

[1] Advanced Encryption Standard (AES) – GIAC, available at : www.giac.org/cissp-papers/42.pdf

[2] Minli Dai, "Innovative Computing and Information", International Conference, ICCIC 2011, Wuhan, China, September 17-18, 2011.

[3] Vincent Rijmen, "Efficient Implementation of the Rijndael S-Box.", Katholieke University Leuven, Dept. ESAT, Belgium.

[4] Akashi Satoh, Sumio Morioka, Kohji Takano and Seiji Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization", Springer-Verlag Berlin Heidelberg, 2001.

[5] X. Zhang and K. K. Parhi, "High Speed VLSI architectures for AES algorithm", IEEE Transactions on VLSI Systems, Vol.12, No. 9, pp 957-967, 2004.

[6] R. Liu, K.K.Parhi "Fast composite field architectures for Advanced Encryption standard" Proceedings GLSVLSI'08, Orlando, Florida, USA, pp 65-70, May 4–6, 2008.

[7] T Good, M. Benaissa, "AES on FPGA from the Fastest to the Smallest", LNCS 3659, pp. 427-440, 2004.

[8] J Zambreno, D. Nguyen, Alok Choudhary, "Exploring Area/Delay Trade-offs in an AES FPGA Implementation" Lecture Notes in Computer Science 3203, Proc.FPL, Antwerp, Belgium, pp 575-585, 2004.

[9] M. M. Wong, M.L.D. Wong, "A high throughput Low power compact AES S-box implementation using composite field arithmetic and Algebraic form representation", Proc. IEEE 2nd Asia Symposium on Quality Electronic Design, pp 318-323, 2010.

[1] Rashmi Ramesh Rachh, P.V. Ananda Mohan and B.S. Anami, "Efficient Implementations of AES S box and Inverse S- box", Proc. IEEE TENCON, Singapore, pp 1-6, 2009.

[2] M. Fayed, M. El-Kharashi and F. Watheq Gebali, "A High-Speed, Fully-Pipelined VLSI Architecture for Real-Time AES", 4th International Conference on Information & Communications Technology, 2006.