International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4| ISSN: 2456-3307

High Security S-Box Architecture for Triple AES Byte Substitution

C.Reddy Usha[1], Mr.C.Manoj Kumar[2]
[1]M.Tech Scholar, Department of ECE, Sir Visveshwaraiah Institute Of Science & Technology, Madanapalli, Chittoor, Andhra Pradesh, India
[2]Assistant Professor, Department of ECE, Sir Visveshwaraiah Institute Of Science & Technology, Madanapalli, Chittoor, Andhra Pradesh, India

ABSTRACT

In this paper, presents an optimized combinational logic based Rijndael S-Box implementation for the SubByte transformation(S-box) in the Triple Advanced Encryption Standard (AES) algorithm on FPGA. An optimum number of pipeline registers based on Spartan-3E FPGA. The design is fully synthesizable using Verilog HDL. We also conduct a performance analysis and comparison of the proposed architecture with those achieved by existing techniques. The comparison shows that the proposed architecture outperforms the existing techniques in terms of speed and area, efficient implementation of pipelined S-Box was synthesized and implemented using Xilinx ISE v14.3 and Xilinx Spartan-3E .

Keywords : Pipelined S-Box, FPGA, AES, Spartan-3E, Triple AES

I. INTRODUCTION

Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. It uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key. Byte substitution and Inverse Byte Substitution are the most complex steps in the encryption and decryption processes. In these steps each byte of the state array will be replaced with its equivalent byte in the S-box or the Inverse S-box. As AES algorithm use elements within the GF (2 8), each element in the state array represents a byte with a value that varies between 00H-FFH. The S-box has a fixed size of 256 bytes represented as (16*16) bytes matrix. In this paper propose an Optimized and pipelined architecture for Sbox block in AES based on combinational logic.

We used minimum number of logic gate in proposed design. In recent years, a number of researches have been proposed for Implementation of S-box by using the FPGA by. In continue we present some researches, in , a software method of producing the multiplicative inverse values, which is the generator of S-box values and the possibilities of implementing the methods in hardware applications will be discussed. The method is using the log and antilog values. The method is modified to create a memory-less value generator in AES hardware-based implementation. In, they propose an improved masked AND gate, in which the relationship between inputs masked values and masks, is nonlinear. Usually, when converting S-box operations from GF (2 8) to GF (((22) 2) 2), all the necessary computations become XOR and AND operations. Therefore, to fully mask AES S-box is to substitute the unmasked XOR and AND operations with the proposed masked AND gate and protected XOR gate. In, a general method for sharing common sub expressions derived from the algebraic finite fields is proposed.

The Advanced Encryption Standard is the standard symmetric key block cipher certified by the National Institute of Standards and Technology (NIST) in Federal Information Processing Standard (FIPS) 197. A full AES implementation can be broken down into two main components: the cipher and the key expander. The cipher is the component which is responsible for performing encryption or decryption on blocks of input data, while the key expander is responsible for preparing the input key for use by the cipher in each round. The function SubBytes is the only non-linear function in AES, operating on each of the state bytes independently as shown in Fig It substitutes all bytes of the State using a look-up table called S-Box as in Table I. The hardware complexity of the AES cryptographic module is dominated by the S-box because it is the most essential part of AES.

II. ADVANCED ENCRYPTION STANDARD (AES)

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.
The features of AES are as follows :

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

A. Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below

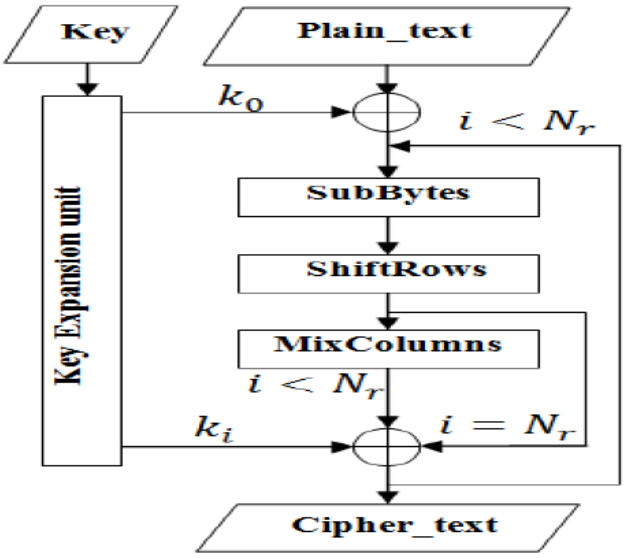

Figure 1. AES encryption algorithm Structure

1) Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

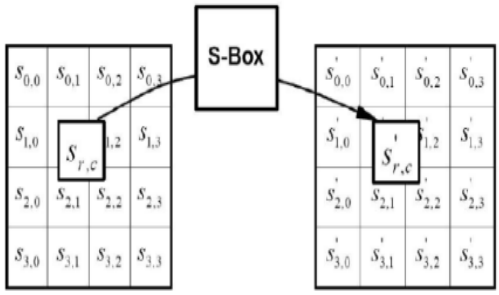

Figure 2. SubBytes function operates on state

2) SHIFT ROWS

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

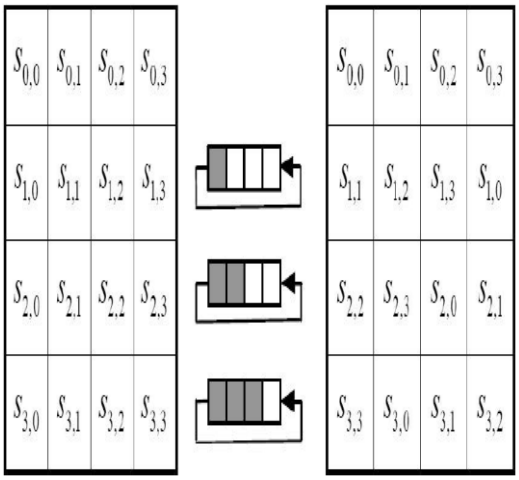

Figure 3. Shift rows

3) MIX COLUMNS

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Figure 4. Mix Columns

4) ADD ROUNDKEY

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

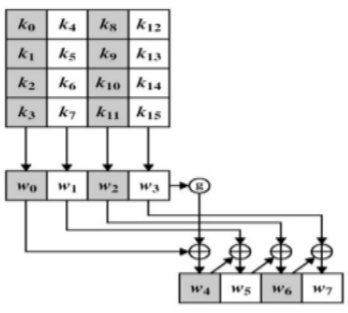

Figure 5. AES key expansion

III. THE S-BOX CONSTRUCTION

The content of the S-Box can be computed based on performing two transformations;

1.Multiplicative inverse

2. Affine Transformation

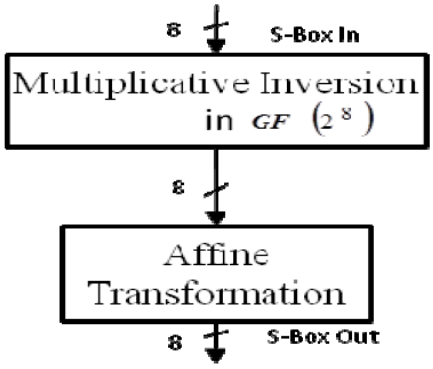

Figure 6. The data flow diagram for the S-Box

A. Affine Transformation

The affine transformation f is defined in a matrix form as in which can also be describing as a polynomial multiplication, followed by the XOR with a constant as outlined in.

$$\begin{pmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{aligned} b_7 &= a_7 \oplus a_6 \oplus a_5 \oplus a_4 \oplus a_3 \\ b_6 &= \overline{a_6 \oplus a_5 \oplus a_4 \oplus a_3 \oplus a_2} \\ b_5 &= \overline{a_5 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1} \\ b_4 &= a_4 \oplus a_3 \oplus a_2 \oplus a_1 \oplus a_0 \\ b_3 &= a_7 \oplus a_3 \oplus a_2 \oplus a_1 \oplus a_0 \\ b_2 &= a_7 \oplus a_6 \oplus a_2 \oplus a_1 \oplus a_0 \\ b_1 &= \overline{a_7 \oplus a_6 \oplus a_5 \oplus a_1 \oplus a_0} \\ b_0 &= \overline{a_7 \oplus a_6 \oplus a_5 \oplus a_4 \oplus a_0} \end{aligned}$$

B. Multiplicative Inversion

The composite field mechanism used for calculating Multiplicative inverses in AES S-Boxes is an efficient method which was proposed early.

C. Mapping And Inverse Isomorphic Mapping

Both isomorphic mapping (δ) and inverse isomorphic mapping (δ−1) can be represented as 8 × 8 matrix. then the isomorphic mappings and its inverse can be written as δ×q and δ−1×q consecutively, which are a case of matrix multiplication as shown below.

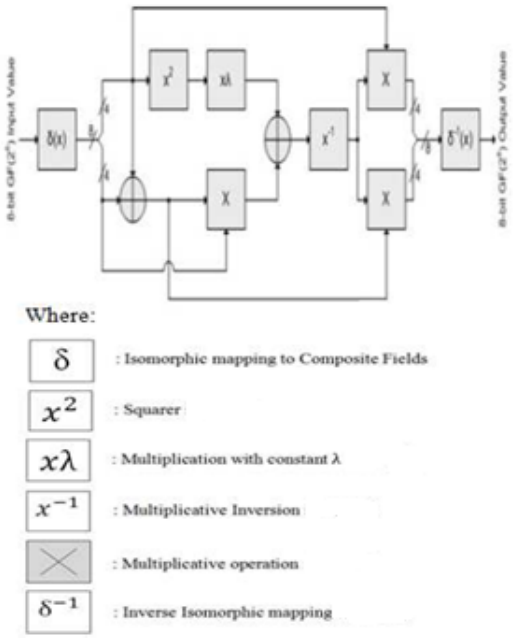

Figure 7. Multiplicative inversion module for the S-Box

$$\delta \times q = \begin{pmatrix} q_7 \oplus q_5 \\ q_7 \oplus q_6 \oplus q_4 \oplus q_3 \oplus q_2 \oplus q_1 \\ q_7 \oplus q_5 \oplus q_3 \oplus q_2 \\ q_7 \oplus q_5 \oplus q_3 \oplus q_2 \oplus q_1 \\ q_7 \oplus q_6 \oplus q_2 \oplus q_1 \\ q_7 \oplus q_4 \oplus q_3 \oplus q_2 \oplus q_1 \\ q_6 \oplus q_4 \oplus q_1 \\ q_6 \oplus q_1 \oplus q_0 \end{pmatrix}$$

$$\delta^{-1} \times q = \begin{pmatrix} q_7 \oplus q_6 \oplus q_5 \oplus q_1 \\ q_6 \oplus q_2 \\ q_6 \oplus q_5 \oplus q_1 \\ q_6 \oplus q_5 \oplus q_4 \oplus q_2 \oplus q_1 \\ q_5 \oplus q_4 \oplus q_3 \oplus q_2 \oplus q_1 \\ q_7 \oplus q_4 \oplus q_3 \oplus q_2 \oplus q_1 \\ q_5 \oplus q_4 \\ q_6 \oplus q_5 \oplus q_4 \oplus q_2 \oplus q_0 \end{pmatrix}$$

D. Squaring

The formula for computing the squaring operation is acquired as shown in.

$$\begin{pmatrix} k_3 \\ k_2 \\ k_1 \\ k_0 \end{pmatrix} = \begin{pmatrix} r_3 \\ r_3 \oplus r_2 \\ r_2 \oplus r_1 \\ r_2 \oplus r_1 \oplus r_0 \end{pmatrix}$$

E. Multiplication with constant ʎ

As has been done to drive the formula for computing the squaring operation, Let k = sʎ, where k = {k3 k2 k1 k0}, s = {s3 s2 s1 s0} and ʎ = {1100} are elements. The k_H and k_L terms can be further broken down and the result of the decomposition is shown in

$$\begin{aligned} k_3 &= s_2 \oplus s_0 \\ k_2 &= s_3 \oplus s_2 \oplus s_1 \oplus s_0 \\ k_1 &= s_3 \\ k_0 &= s_2 \end{aligned}$$

F. Multiplicative Inversion

All have derived the formula for computing the multiplicative inversion. Let k be an element so the multiplicative inversion of k-1={(k3)-1 (k2)-1 (k1)-1 (k0)-1} can be computed from .

$$k_3^{-1} = k_3 \oplus k_3k_2k_1 \oplus k_3k_0 \oplus k_2$$
$$k_2^{-1} = k_3k_2k_1 \oplus k_3k_2k_0 \oplus k_3k_0 \oplus k_2 \oplus k_2k_1$$
$$k_1^{-1} = k_3 \oplus k_3k_2k_1 \oplus k_3k_2k_0 \oplus k_2 \oplus k_2k_0 \oplus k_1$$
$$k_0^{-1} = k_3k_2k_1 \oplus k_3k_2k_0 \oplus k_3k_1 \oplus k_3k_1k_0 \oplus k_3k_0 \oplus k_2 \oplus k_2k_1 \oplus k_2k_1k_0 \oplus k_1 \oplus k_0$$

G. Multiplication

The multiplication is defined as in Fig. It can be observed that there are addition, multiplication operations and multiplication with constant. The multiplication requires decomposition to be implemented in hardware.

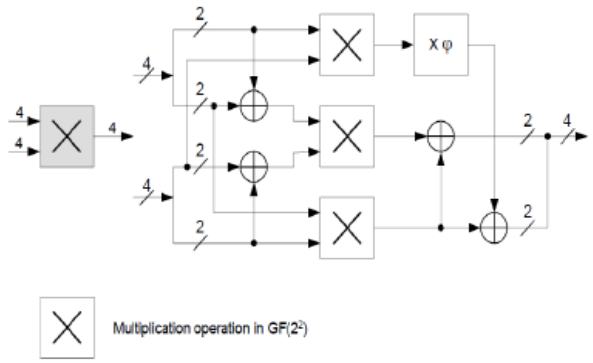

Figure 8. Multiplication operation

H. Triple AES

Triple AES is a process to reuse implementations of AES with serial installation of three instances of AES to improve the security of the data. In the process of encryption, use of three keys leads to the formation of triple AES. The AES operation is performed three times with three different keys. K1, K2 and K3 are the three different keys used on a plain text P to convert it into cipher text C. By using the Rijndael algorithm, the encryption E1 is done with the help of key K1 and the result of this is fed to encryption E2 having key K2 and third encryption E3 is performed with key K3. Multi encryption of the data increases its security and makes it difficult to decode or crack the data. Figure 2 shows the model for implementing triple AES.

The decrypting procedure is same as encryption procedure but executed in reverse. Some keys can make the encryption weak i.e. if second or first key or the third or second key is same. This encryption procedure will almost be same as the encryption procedure for standard AES.

The Triple AES encryption is done as:
Encryption along with K1
Encryption along with K2
Encryption along with K3

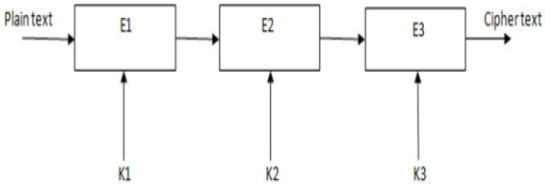

Figure 9. Model of triple AES

IV. RESULTS

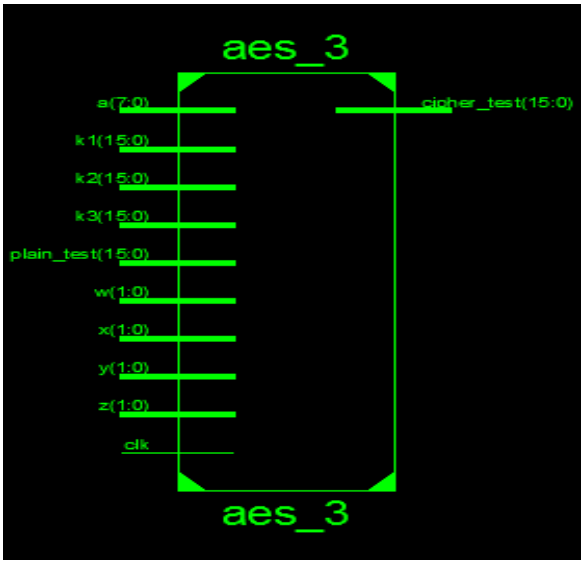

Figure 10. Block Diagram

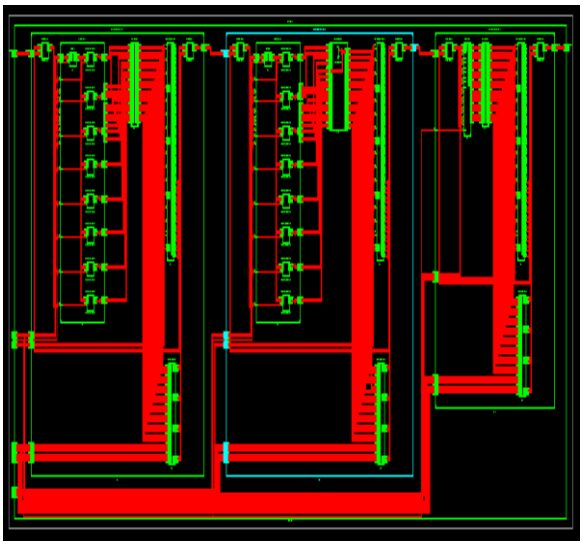

Figure 11. RTL Schematic

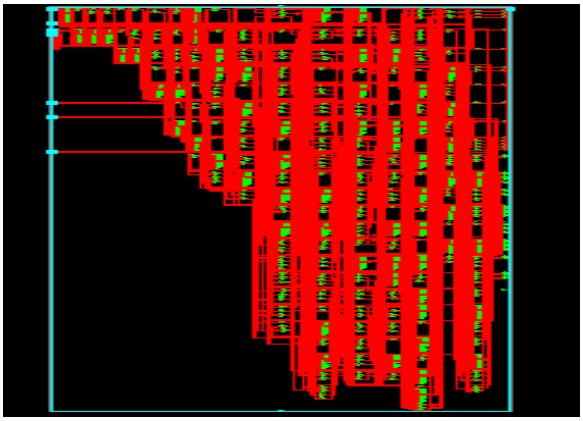

Figure 12. Technology Schematic

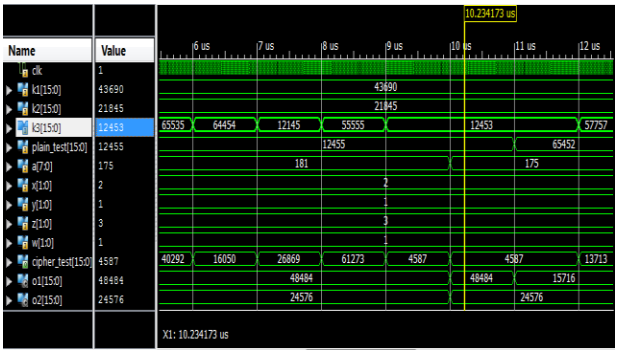

Figure 13. Simulation Results

V. CONCLUSION

The aim of paper is design and implementation of the optimized combinational logic based Rijndael S-Box on FPGA. Proposed method Triple AES is based on combinational logic, thus it is low power and number of logic gates is very low. The approach used for increase performance is pipelining technique we use stage pipelining in S-Box design. This method has more speed and low power than previous methods.

VI. REFERENCES

[1]. Announcing the Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 2001.

[2]. J. Daernen and V.Rijrnen, "Specification of Rijndael," in The Design of Rijndael: AES - The Advanced Encryption Standard, Berlin; New York: Springer-Verlag Berlin Heidel-berg, 2002, pp.31-35

[3]. Satoh, S. Morioka, K. Takano and S. Munetoh, "Acompact rijndael hardware architecture with S-box optimization," Springer- Verlag Berlin Heidelberg, 2001.

[4]. Yibo Fan, Takeshi Ikenaga, YukiyasuTsunoo, and Satoshi Goto,(2008) "A Lowcost Reconfigurable Architecture for AES Algorithm" proceedings of world academy of science, engineering and technology volume 31 july 2008 ISSN 2070-3740

[5]. William Stallings, "Cryptography and Network Security", Third Edition, www.williamstallings.com/Crypto3e.html

[6]. P. Chodowiec, P. Khuon and K. Gaj,(2001) "Fast Implementations of Secret-Key Block Ciphers Using Mixed Inner- and Outer-Round Pipelining," Proc. ACM/SIGDA Int. Symposium on Field Programmable Gate Arrays, FPGA'01, Monterey, CA.

[7]. M. McLoone and J. McCanny,(2001) "Single-chip FPGA Implementation of the Advanced Encryption Standard Algorithm," in Proc. 11th Int. Conf. Field-Programmable Logicand Applications (FPL 2001), LNSC 2147, pp. 152-161.

[8]. N. Sklavos and O. Koufopavlou,(2002) "Architectures and VLSI Implementations of the AES-Proposal Rijndael," IEEE Trans.on Computers, vol. 51, Issue 12, pp. 1454-1459.

[9]. J. H. Shim, D. W. Kim, Y. K. Kang, T.W. Kwon and J. R. Choi,(2002) "A Rijndael Crypto processor Using Shared On-the-fly Key Scheduler,", pp147-150, 2002.

[10]. Refik Sever, A. NeslinI smailoglu, Yusuf C. Tekmen, Murat Askar, BurakOkcan,(2004)"A High speed fpga Implementation of the Rijndael Algorithm" Proceedings of the EUROMICRO Systems on Digital System Design (DSD'04),IEEE, pp.358-362.