HIGH SPEED BLOW FISH ALGORITHM USING S-BOX

Mohammad Azharuddin[1], Mani Poornima

MTech Student, Department of ECE, Vemu Institute of Technology, chittoor, India, Email: azaruddin.mohammad@gmail.com
Associate Professor, Department of ECE, Vemu Institute of Technology, chittoor, India, Email: poornimakpk20@gmail.com

ABSTRACT: Advanced Encryption Standard (AES) is the most secure symmetric encryption technique that has gained worldwide acceptance. The AES based on the Rijndael Algorithm is an efficient cryptographic technique that includes generation of ciphers for encryption and inverse ciphers for decryption. For Higher security and speed of encryption/decryption We are designing the Blow fish algorithm that the proposed architecture out performs the existing techniques in terms of speed. Implementation of S-Box was synthesized and implemented using Xilinx ISE v14.3 and Xilinx Spartan-3E.

Keywords: S-Box, FPGA, AES, Spartan-3E.

I. INTRODUCTION

The purpose is to provide a standard algorithm for encryption, strong enough to keep U.S. government documents secure for at least the next 20 years. The earlier Data Encryption Standard (DES) had been rendered insecure by advances in computing power, and was effectively replaced by triple-DES. Now AES will largely replace triple-DES for government use, and will likely become widely adopted for variety of encryption needs, such as secure Transactions via the Internet. 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 S box 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 A wide variety of approaches to implementing AES have appeared, to satisfy the varying criteria of different applications. Some approaches seek to maximize throughput, others minimize power consumption, and yet others minimize circuitry, and . For the latter goal, Rijmen suggested using subfield arithmetic in the crucial step of computing an inverse in the Galois Field of 256 elements reducing an 8-bit calculation to several 4-bit ones. Satoh et al. further extended this idea, using the "tower field" approach of Paar, breaking up the 4-bit calculations into 2-bit ones, which resulted in the smallest AES circuit to date.The AES algorithm have Three different key sizes are allowed: 128 bits, 192 bits, or 256 bits, and the corresponding number of rounds for each is 10 rounds, 12 rounds, or 14 rounds, respectively. From the original key, a different "round key" is computed for each of round.

A. ENCRYPTION PROCESS

AES algorithm uses a round function that is composed of four different byte oriented transformations: SubBytes, ShiftRows, MixColumns and Add Round Key.

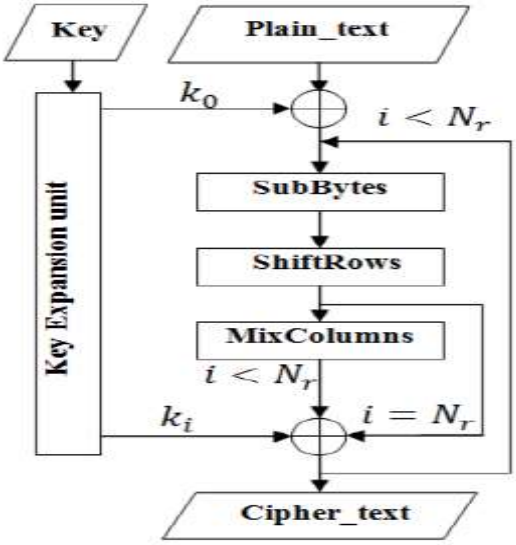


Fig 1 AES encryption algorithm Structur

Pseudo code for generating the expanded key from the actual key. The key is copied into the first four words of the expanded key. The remainder of the expanded key is filled in four words at a time.

B. BYTE SUBSTITUTION (SUBBYTES)

This is simply a table lookup using a 16×16 matrix of byte values called an s-box. This matrix consists of all the possible combinations of an 8 bit sequence
(28 = 16 × 16 = 256).

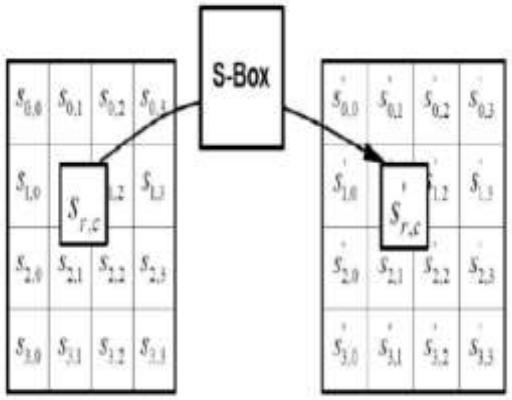


Fig 2 SubBytes function operates on state

The Inverse substitute byte transformation makes use of an inverse s-box.

C. SHIFT ROWS

This is a simple permutation an nothing more.

- The first row of state is not altered.
- The second row is shifted 1 bytes to the left in a circular manner.
- The third row is shifted 2 bytes to the left in a circular manner.

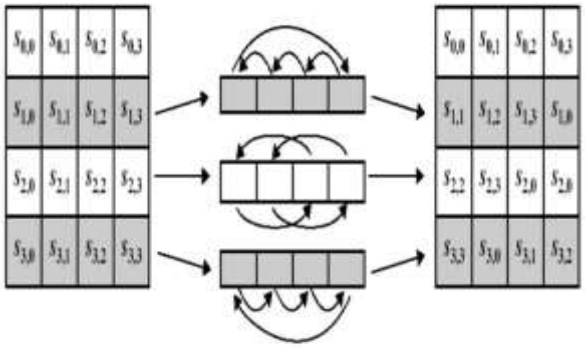


Fig 3 Shift rows

A one byte shift is therefore a linear distance of four bytes. The transformation also ensures that the four bytes of one column are spread out to four different columns.

D. MIX COLUMNS

This stage is basically a substitution but it makes use of arithmetic of GF(28). Each column is operated on individually.

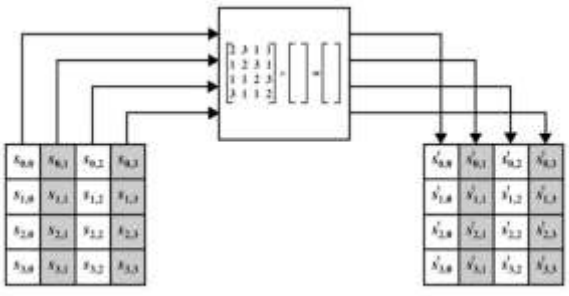


Fig 4: mix columns

The transformation can be determined by the following matrix multiplication on state

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

E. ADD ROUNDKEY

The key is copied into the first four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each added word depends on the immediately preceding word, and the word four positions back . In three out of four cases, a simple XOR is used.

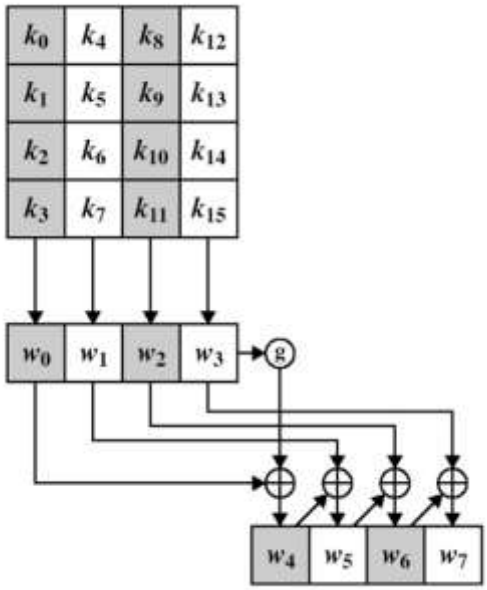


Fig 5 AES key expansion

The round constant is a word in which the three rightmost bytes are always 0. Thus the effect of an XOR of a word with Rcon is to only perform an XOR on the leftmost byte of the word.

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

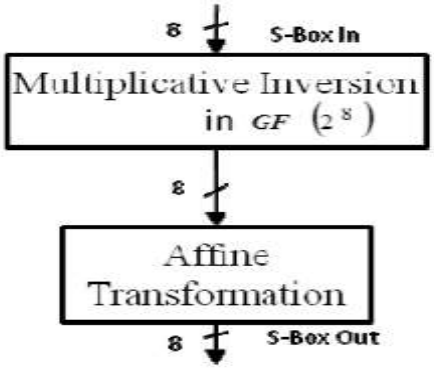


Fig 6 The data flow diagram for the S-Box

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.

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

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

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.

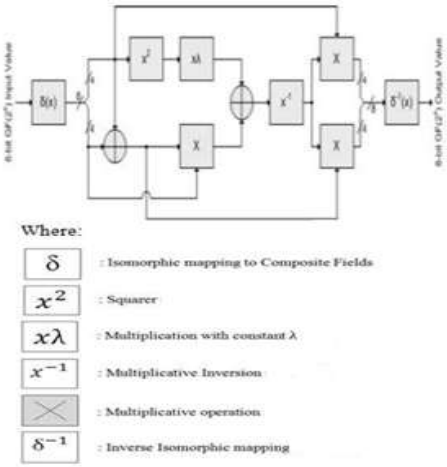


Fig 7 Multiplicative inversion module for the S-Box

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

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

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 IN GF(24)

the multiplicative inversion. Let k be an element of GF(24) so the multiplicative inversion of k-1={(k3)-1 (k2)-1 (k1)-1 (k0)-1}.

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

G. MULTIPLICATION IN GF(24)

It can be observed that there are addition, multiplication operations in GF(22) and multiplication with constant _. The multiplication in GF(22), requires decomposition to GF(2) to be implemented in hardware.

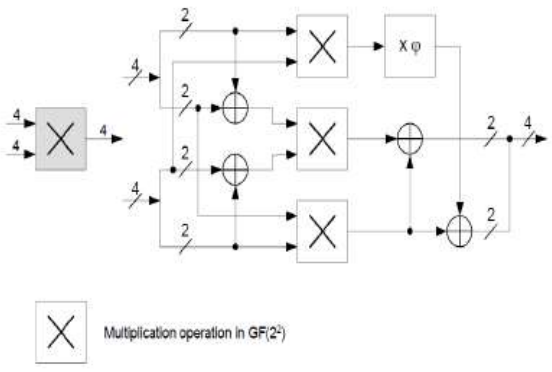


Fig 8 Multiplication operation

IV. BLOWFISH ALGORITHM

Blowfish is a symmetric block cipher algorithm. It had been designed by Bruce Schneier in 1993. It takes 64-bit plaintext

and variable-length key, from 32 bits to 448 bits, as inputs and 64-bit cipher text as an output . Blowfish is a symmetric block cipher. It has a fixed 64-bit data block size and a variable secret key range from 32 bits to 448 bits.

The algorithm consists of two parts: key expansion and data encryption. The key expansion converts a key of at most 448 bits into several sub key arrays totaling 4168 bytes. The data encryption occurs via a 16-round Feistel network.

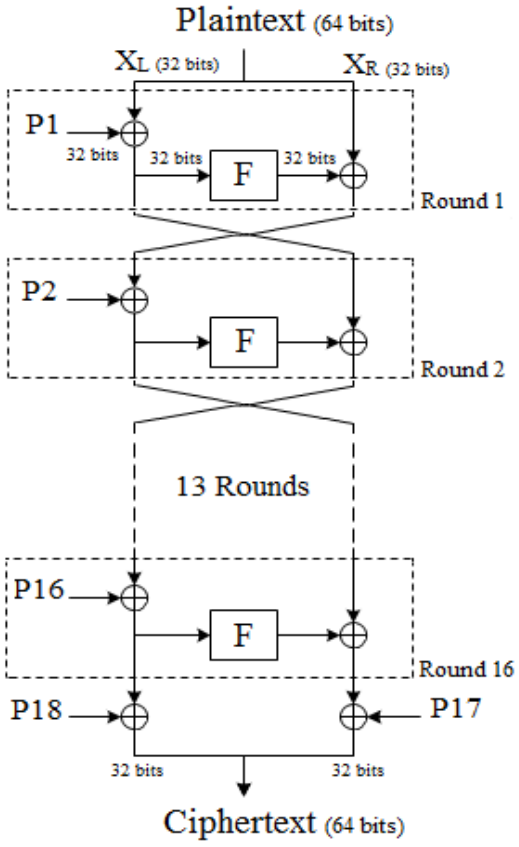


Fig 9 Block diagram of Blowfish algorithm

As mentioned previously, Blowfish is a Feistel network consisting of 16 rounds, The inputs are 64-bit plaintext and 18 Parray syb keys (32 bits). The output is a ciphertext (64 bits).

Function F is calculated by It divides XL into four eight-bit quarters: a, b, c, and d. These quarters are used as input to the S-boxes. The outputs are added (modulo 232) and XORed to produce the final 32-bit output. Decryption is exactly same as encryption, except that P1, P2 ... P18 are used in the reverse order.

F(XL)=((S1,+S2,bmod 232)XORS3,c)+S4,d mod232 (1)

The proposed Blowfish design aimed to increase the throughput. Thus, pipeline technique is adopted. The pipeline strategy modifies the critical path by increasing ,It consists in parallelizing the data inputs and outputs with the processing.

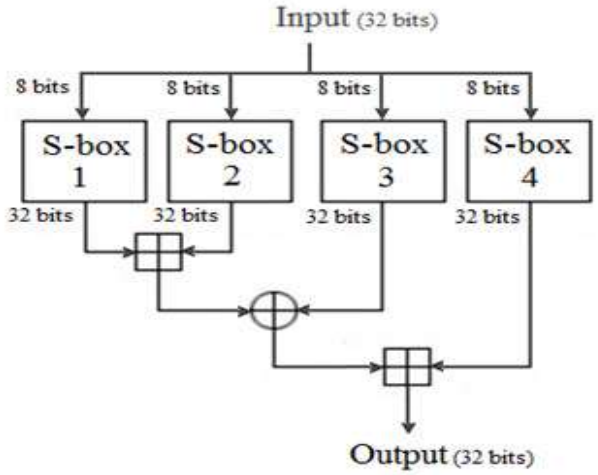
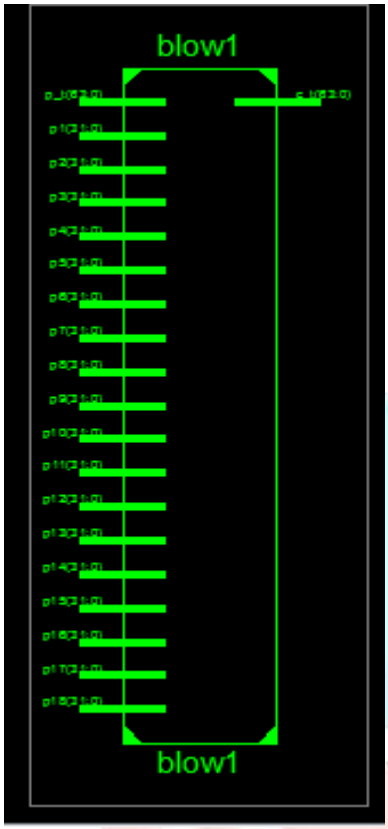


Fig 10 Function F

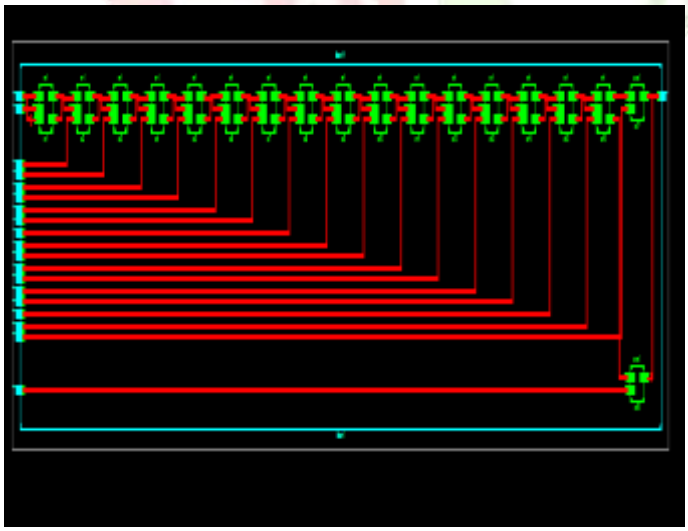
Cryptanalysis refers to the process of illegally attempting to recover the plaintext (or the key) that corresponds to a particular ciphertext. Full-round version of Blowfish algorithm is invulnerable against cryptanalysis, to date. Each one is executed separately and in a parallel manner by inserting registers in appropriate places.our proposed architecture, we used pipelining and parallelism in order to break the critical path delay and to obtain high encryption throughput at the compared with the based non-pipelined Blowfish algorithm.The Blow fish algorithm using S-Box is more efficient then the previous system.

V. RESULTS

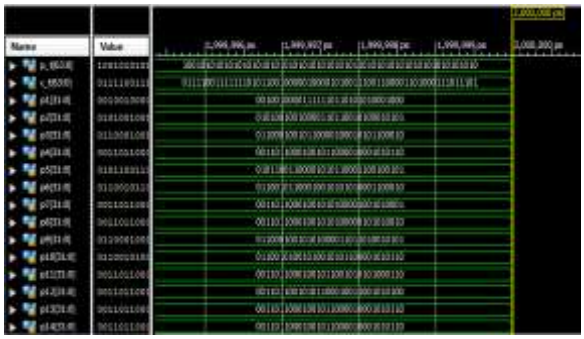
BLOCK DIAGRAM



RTL SCHEMATIC



SIMULATION RESULTS



VI. CONCLUSION

The aim of paper is design and implementation of the optimized combinational logic based Rijndael S-Box on FPGA. Proposed method is based on Algorithm, thus it is low power and number of logic gates is very low. The approach used for increase performance is Blowfish Algorithm we use this design. This method has more speed.

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] A. 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

[1] W. Stalling, "Cryptography and Network Security Principles and Practices", Prentice Hall, 4th ed., 2005.

[2] A. Kahate, "Cryptography and Network Security", Tata McGraw Hill, 2nd ed., 2007.

[3] National Institute of Standards and Technology, Federal Information Processing Standards Publication 46-3: Data Encryption Standard, 1999.

[4] National Institute of Standards and Technology, Federal Information Processing Standards Publication 197: Advanced Encryption Standard, 2001.

[5]D.Joan and R. Vincent, "AES Proposal: Rijndael", National Institute of Standards and Technology, 1999.

[6]B.Schneier,"The Blowfish Encryption Algorithm-One Year Later", Dr. Dobb's Journal, 1995.

[7] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Applied Cryptography", John Wiley and