IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
ISSN: 2278-2834, ISBN: 2278-8735, PP: 63-67
www.iosrjournals.org

VLSI Based Implementation of Single Round AES Algorithm

[1]Mr.Shelke R.B., [2]Mrs.Patil A.P.' [3]Dr.(Mrs)Patil S.B.
[1]*E&TC Engg, Dr.J.J.M.C.O.E, Shivaji University, Kolhapur,India*
[2]*Electronics Engg, Dr.J.J.M.C.O.E, Shivaji University, Kolhapur,India*
[3]*Electronics Engg, Dr.J.J.M.C.O.E, Shivaji University, Kolhapur,India*

ABSTRACT : *This paper presents VLSI based implementation of single round AES algorithm for encryption purpose and it is one of the most popular algorithm used in symmetric key cryptography The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt and decrypt 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. This paper illustrates number of different transformations applied consecutively over the data block bits, in a single iteration, called round. The number of rounds depends on the length of the Key used for the encryption process. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits, this paper implements the 128 bit key length with single round.*
VHDL is used as the hardware description language because of the flexibility to exchange among environments. The software used for this work is Xilinx ISE Design Suite 13.2 the tools primarily used is the Xilinx ISE and ISim for simulation, Synthesis and Implementation. Some results verified using test bench for the input plaint text of 128 bit and output Cipher Text of 128 bit.

Keywords *– Cryptography, Decryption, Encryption, Key schedule, Mix Columns, Shift Rows, Simulation, Sub Bytes, Synthesis.*

I. INTRODUCTION

The Advanced Encryption Standard (AES) began in 1997 with an announcement from NIST seeking a replacement for the aging and insecure Data Encryption Standard (DES).At that point, DES has been repeatedly shown to be insecure, and to inspire confidence in the new standard they asked the public to once again submit new designs. DES used a 56-bit secret key, which meant that brute force search was possible and practical. Along with a larger key space, AES had to be a 128-bit block cipher; that is, process 128-bit blocks of plaintext input at a time. AES also had to support 128, 192, and 256-bit key settings and be more efficient than DES. Today it may seem rather redundant to force these limitations on a block cipher. We take AES for granted in almost every cryptographic situation [1]. However, in the 1990s, most block ciphers such as IDEA and Blowfish were still 64-bit block ciphers. Even though they supported larger keys than DES, NIST was forward looking toward designs that had to be efficient and practical in the future.

As a result of the call, 15 designs were submitted, of which only five (MARS, Twofish, RC6, Serpent, and Rijndael) made it to the second round of voting. The other 10 were rejected for security or efficiency reasons. In late 2000, NIST announced that the Rijndael block cipher would be chosen for the AES.The decision was based in part on the third round voting where Rijndael received the most votes (by a fair margin) and the endorsement of the NSA. Technically, the NSA stated that all five candidates would be secure choices as AES[1], not just Rijndael. Rijndael is the design of two Belgian cryptographers Joan Daemen and Vincent Rijmen It was proven to resist both linear and differential cryptanalysis (attacks that broke DES) and has very good statistical properties in other regards. In fact, Rijndael was the only one of the five finalists to be able to prove such claims. The other security favorite, Serpent, was conjectured to also resist the same attacks but was less favored because it is much slower.

The structure of this paper is as follows : Section I introduces about AES algorithm used in cryptography. In Section II, we briefly described the four transformations required for the encryption of AES algorithm. Section III discusses result and implementation of Single Round AES algorithm. Section IV describes conclusion of the paper.

II. AES ALGORITHM OVERVIEW

2.1The AES Algorithm

The AES [5] accepts a 128-bit plain text, and produces a 128-bit cipher text under the control of a 128 ,192, or 256-bit secret key. It is a Substitution-Permutation Network design with a single collection of steps called a round For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key length [4]

Algorithm	Key Length (Nk words)	Block Size (Nb words)	Number of Rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Figure 1. Block- Round Combinations.

that are repeated 10, 12, or 14 times (depending on the key length) to map the plain text to cipher text each round uses its own 128-bit round key, which is derived from the supplied secret key through a process known as a key schedule. It distributes the entropy of the key across each of the round keys. If that entropy is not spread properly, it causes all kinds of trouble such as equivalent keys, related keys, and other similar distinguishing attacks. AES treats the 128-bit input as a vector of 16 bytes organized in a column major (big endian) 4 x 4 matrix called the state. That is, the first byte maps to S0,0, the third byte to S2,0the fourth byte to S3,0, and the 16th byte maps to S3,3shown in Fig(2)

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Figure 2 State Diagram

A single round of AES consists of four Transformations namely Sub Bytes, Shift Rows, Mix Columns and Add Round Key shown in Fig (3).

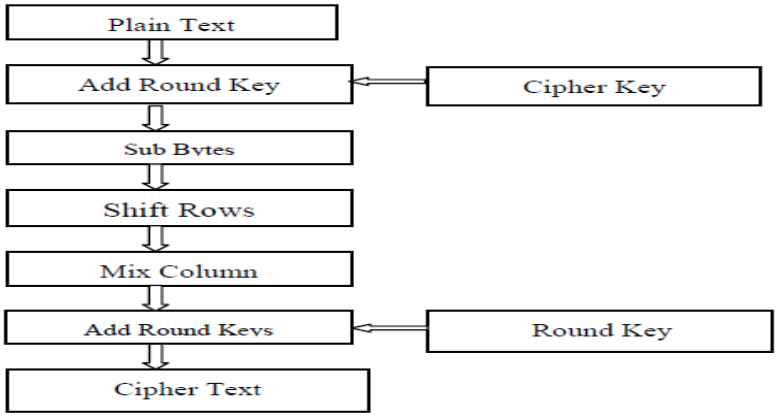

Figure 3 Single Round AES Algorithm

2.2 Add Round Key

This step of the round function adds in GF(28)[9]. the round key to the state. It performs 16 parallel additions of key material to state material. The addition is performed with the XOR operation Fig (4).

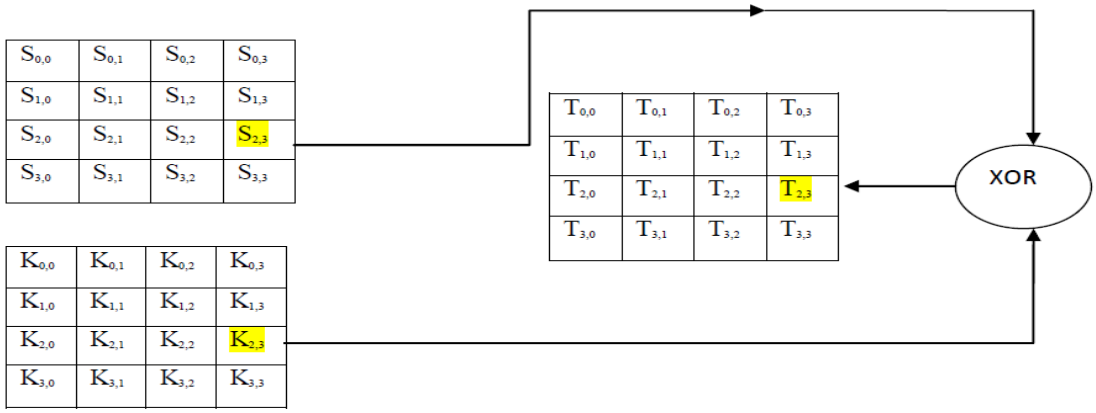

Figure 4 Add Round Key Function

The k matrix is a round key and there is a unique key for each round. Since the key addition is a simple XOR, it is often implemented as a 32-bit XOR across rows in 32-bit keys.

2.3 Sub Bytes

It maps each of the 16 bytes in parallel to a new byte by performing two-step substitutions Fig (5). The substitution is composed of a multiplicative inversion in GF(28)[9]followed by an affine transformation (Fig 6) in GF(28)[9].The multiplicative inverse of a unit S is another unit T, such that ST modulo the AES polynomial is congruent .

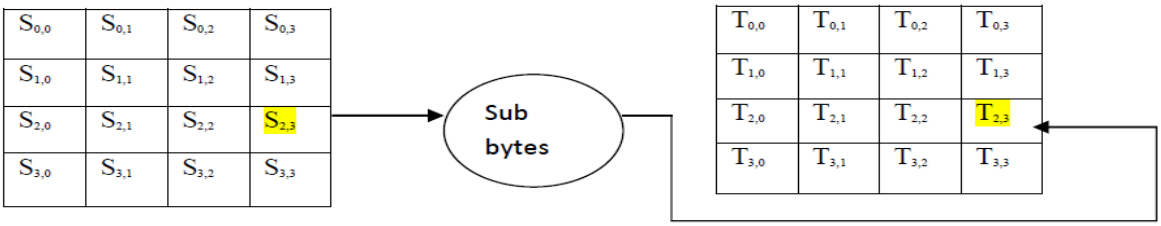

Figure 5 SubBytes Function

$$\begin{pmatrix} 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} x_0\\ x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7 \end{pmatrix} + \begin{pmatrix} 1\\ 1\\ 0\\ 0\\ 0\\ 1\\ 1\\ 0 \end{pmatrix}$$

Figure 6 Affine Transformation

2.4 Shift Rows

The Shift Rows operation only changes the byte position in the state. It rotates each row with different offsets to obtain a new state as follows:

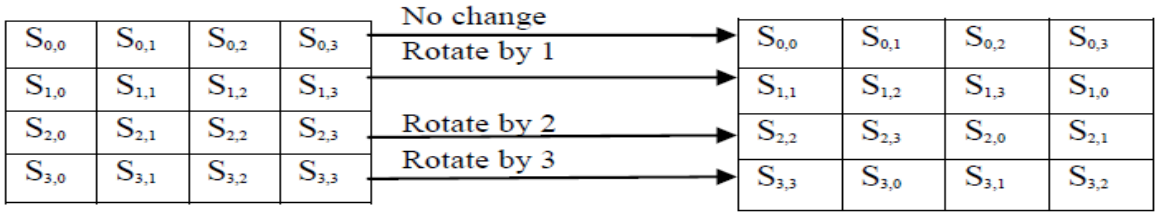

Figure 7 Shift Row Transformations

The first row is unchanged, the second row is left circular shifted by one, the third row is by two, and the last row is by three.

2.5 Mix Columns

The Mix Columns operation [3] mixes every consecutive four bytes of the state to obtain four new bytes as follows:

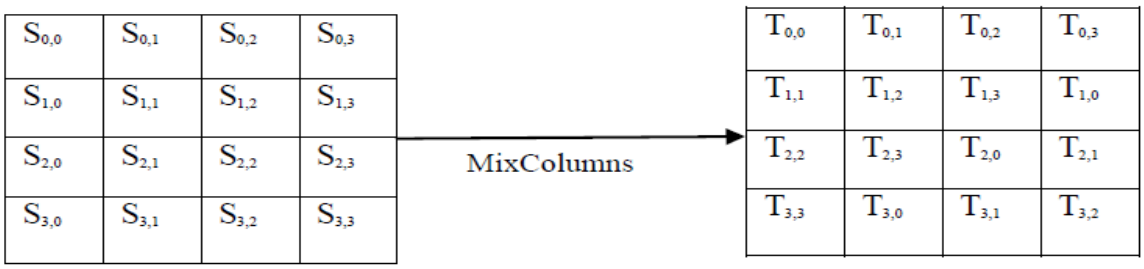

Figure 8 Mix Columns Transformations

Let Si,Si+1, Si+2 and Si+3 represent every consecutive four bytes, where i belongs to {0,1,2,3}Then, the four bytes are transformed by

$$\begin{pmatrix} t_i \\ t_{i+1} \\ t_{i+2} \\ t_{i+3} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 02 & 03 & 01 \end{pmatrix} \begin{pmatrix} s_i \\ s_{i+1} \\ s_{i+2} \\ s_{i+3} \end{pmatrix} \quad (1)$$

Each entry of the constant matrix in (1) belongs to GF(28)[9]hence Equation (1) is a matrix-vector multiplication over GF(28)[9].

2.6 AddRoundKey and Key Expansion

Each round has a 128-bit round key which is segmented into 16 bytes ki Add Round Key is simply addition

$$t_i = s_i + k_i \text{ , Where } 0 \leq i \leq 15 \quad (2)$$

The key expansion expands a unique private key as a key stream of (4r + 4) 32-bit words, where r is 10, 12, or 14. The private key is segmented into Nk words according to the key length, where Nk is 4, 6, or 8 for a 128-bit, 192-bit, or 256-bit cipher key, respectively. then, it generates the ith word (32 bits) by EXORing the (i-Nk)th word with either the (i -1)th word or the conditionally transformed (i- 1)thword, where $Nk \leq i \leq (4r + 3)$. The (i- 1)th word is conditionally transformed by RotWord, SubBytes and EXORing with Rcon[i/ Nk]={02 [I / Nk],00, 00, 00} where the polynomial presentation of 02[i/ Nk] is x[i/ Nk] over GF(28)[9]. Finally, the key stream is segmented into several round keys which are involved in the Add Round Key operation and last we get the output as cipher text.

III. IMPLEMENTATION OF ROUND ONE AES ALGORITHM

VHDL is used as the hardware description language because of the flexibility to exchange among environments. The software used for this work is Xilinx ISE Design Suite 13.2 The tools primarily used are the Xilinx ISE and ISim for simulation, Synthesis and implementation. Round one encryption AES algorithm waveform results shown below

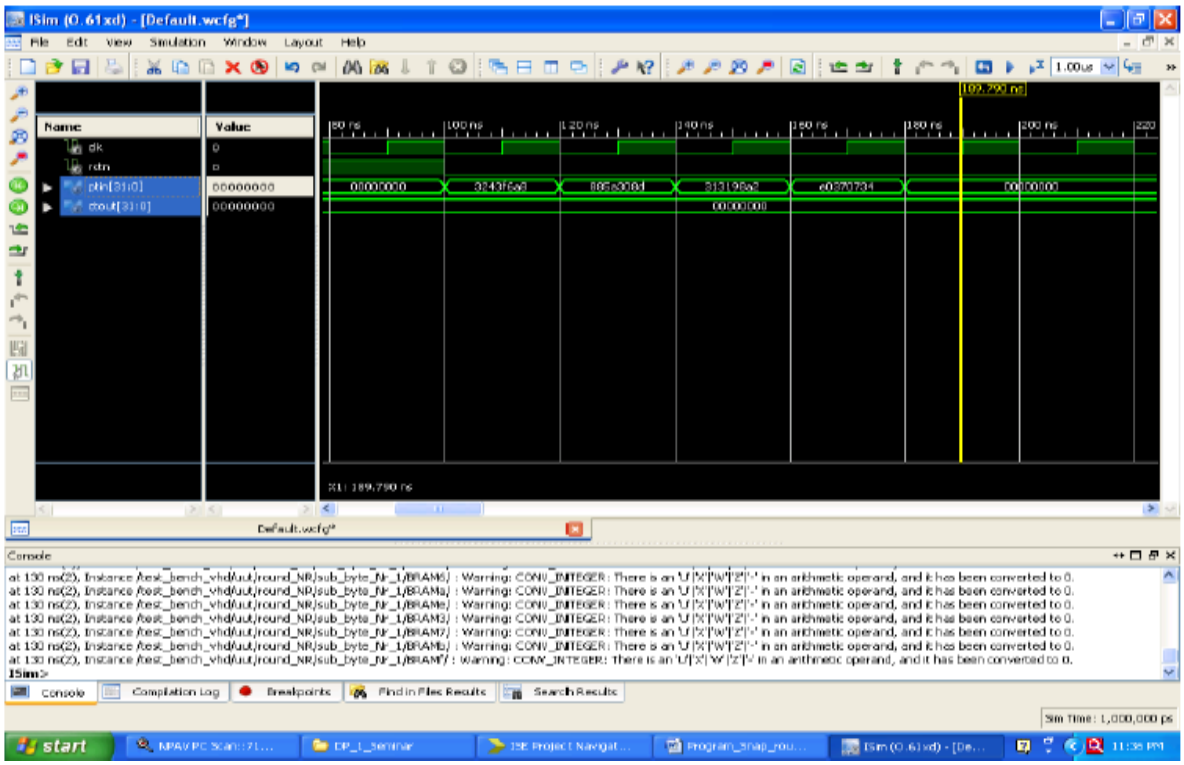

Figure 9. Plain Text Input

Test bench output shown in Fig (9) for the input plaint text of 128 bit in four clock cycle default binary input plain text converted in the hexadecimal form.

Input State: 3243 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

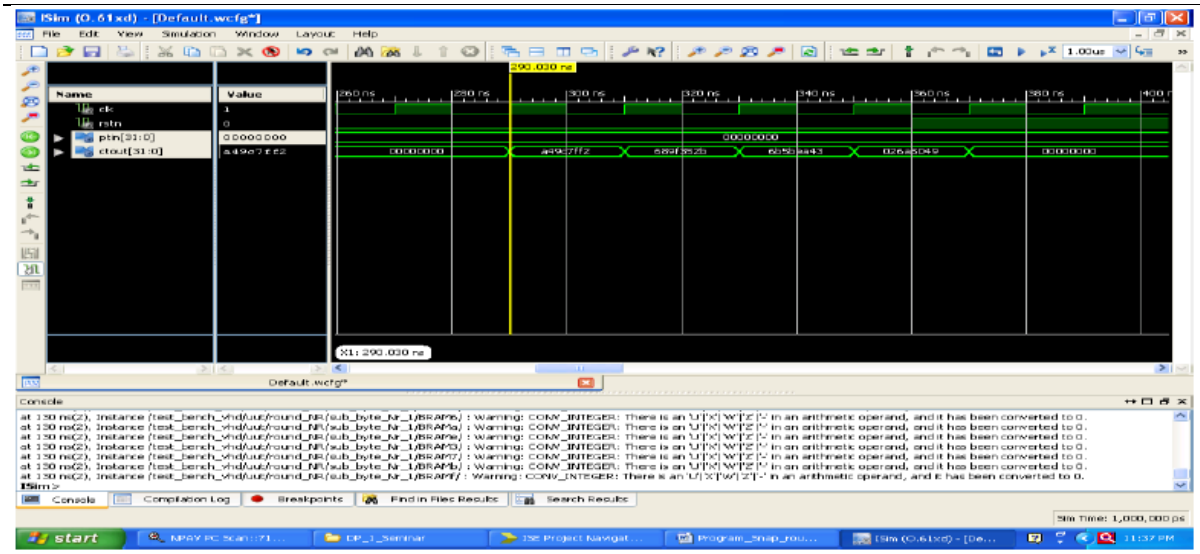

Figure 10. Cipher Text Output

Test bench output shown in Fig(10) for the output Cipher Text of 128 bit in four clock cycle default binary output Cipher text converted in the hexadecimal form.

Round one Cipher Text : a4 9c 7f f2 68 9f 35 2b 6b 5b ea 43 02 6a 50 49

IV. CONCLUSION

In this paper, single round is implemented for the encryption of AES algorithm. In order to get more security further used four transformations namely AddRound keys ,SubBytes, ShiftRows and MixColumns after single round implementation observed logic utilization report as number of slice flip flops available count 9.312 and used count 808 then number of accupied slices available 4,656 and used 1,565 further Total number of four input Look up tables utilization of 26 %.this Device Utilization summary implemented on target Device xc3s500e-4fg320 also used product version as ISE 13.2 this implementation is to have quick encryption but less security to overcome this limitation further we can implement Encryption of Single Round of AES to increased no of 10 Rounds with additional three transformations namely SubBytes ShiftRows and AddRoundKey.

REFERENCES

[1] Xinmiao Zhang and Keshab K. Parhi "High-Speed VLSI Architectures for the AES Algorithm" *IEEE transactions on very large scale integration (VLSI) systems, vol. 12, no.9, September 2004 ,957*

[2] Adam J. Elbirt, W. Yip, B. Chetwynd, and C. Paar "An FPGA-Based Performance and the Evaluation of AES Block Cipher Candidate Algorithm Finalists" *IEEE transactions on very large scale integration (VLSI) systems, vol. 9, no. 4, august 2001, 545*

[3] H. Li, Z. Friggstad, "An Efficient Architecture for the AES Mix Columns Operation," *IEEE International Symposium on Circuits and Systems, Kobe, Japan, May 2005*

[4] X. Zhang and K. K. Parhi, "Implementation approaches for the advanced encryption standard algorithm," *IEEE Circuits Syst. Mag., vol. 2, no. 4,pp. 24–46, 2002.*

[5] National Institute of Standards and Technology (NIST), "Federal Information Processing Standard 197, The Advanced Encryption standard (AES)", Nov. 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[6] William Stallings, "cryptography and network security principles and practice", (4th Edition, Prentice-Hall, Inc. 2006)

[7] A.A.Zaidan, B.B.Zaidan, Anas Majeed, "High Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm", *World Academy of Science Engineering and Technology (WASET), Vol.54, ISSN: 2070-3724, P.P 468-479.*

[8] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani "New Comparative Study Between DES, 3DES and AES within Nine Factors*" journal of computing, volume 2, issue 3, march 2010, issn 2151-9617*

[9] E. M. Popovici and P. Fitzpatrick, "Algorithm and architecture for a Galois field multiplicative arithmetic processor*," IEEE Trans. Inf. Theory, vol. 49, no. 12, pp. 3303–3307, Dec. 2003.*