(IJIRSE) International Journal of Innovative Research in Science & Engineering
ISSN (Online) 2347-3207

FPGA implementation of AES using Vedic Mathematics

Ambika R[1] , C S Mala[2] , S K Pushpa[3]
Dept. of ECE[1], Dept. of TCE[2] , Dept. of ISE[3]
BMSIT Bangalore, INDIA
ambika2810@gmail.com, csmala63@gmail.com, skpushpamurtheppa@yahoo.com

***Abstract*-In recent years, the importance of security in the information technology has increased significantly. This paper presents a new efficient architecture for high speed advanced encryption standard algorithm using Vedic Mathematics. The proposed architecture is implemented using Field Programmable Gate Array.**

Keywords- AES, Cryptography, Decryption, Encryption, FPGA, LUT, Vedic Mathematics.

I. INTRODUCTION

In cryptography, encryption is the process of translation of data into a secret code. It is unreadable by anyone except those possessing the key. Encryption is important to secure data such that it is received with full integrity at the receiver end. There are two types of encryption called asymmetric encryption (public-key encryption) and symmetric encryption. Decryption is the process of converting the data back into its original form. Advanced Encryption Standard (AES) was developed by J. Daemen and V. Rijmen [1] and set as the standard by The National Institute of Standards and Technology (NIST) in the year 2001.The AES algorithm is a symmetric block cipher. The advantages of using AES algorithm are low power consumption, low cost implementation and resistance to brute force attack.[1] and [2]include comparison of Candidate Algorithms on FPGA, implementation of Sub-bytes and its inverse using combinational logic replacing the look-up table approach, decomposition of inverse Mix Column. [3]deals with high speed encryption using merging technique.[2] and [4] explain resource sharing in terms of functional unit and expressing decryption matrix using encryption matrix.

AES was defined with the following requirements:

• 128-bit blocks (solves issues with CBC)

• Accepts keys of size 128, 192 and 256 bits (128 bits are enough to resist exhaustive key search;
• Has no academic weakness worse than exhaustive key search
• should be as fast as 3DES (AES turned out to be much faster than 3DES in software, typically
5 to 10 times faster)
The resistance of AES towards differential and linear cryptanalysis comes from a better "avalanche effect" (i.e. a bit flip at some point quickly propagates to the complete internal state) and contains, bigger "S-boxes" (a S-box is a small lookup table used within the algorithm, and is an easy way to add non-linearity; in DES, S-boxes have 6-bit inputs and 4-bit outputs; in AES, S-boxes have 8-bit inputs and 8-bit outputs).

APPLICATIONS:

Encryption plays an important role in authentication of identity in various institutions and companies, credentialing Systems, electronic Signatures, electronic cash transfers, Email, secureelectronic transactions, generating password, digital mobile phones and Confidential documents in the office. Vedic Mathematics can be suitably used to enhance the speed.

II. AES ALGORITHM:

The AES algorithmic program was developed by NIST for secret writing and decoding of information. It works on a 128 bit datablock that is taken into account as a 4x4 matrix referred to as the state matrix. Every byte in the state matrix is a component within the mathematician field (2^8). Mathematician field may be a finite field consisting of a finite range of parts. All arithmetic operations on the bytes square measure allotted in mathematician field (2^8). The AES algorithmic program uses 128, 192 or 256 bits key. Depending on the size of the key the numbers of rounds are 10, 12, and 14 respectively In every spherical, four transformations are performed. In the last around the combine

column operation isn't performed just in case of secret writing and inverse combine column in case of decoding operation.

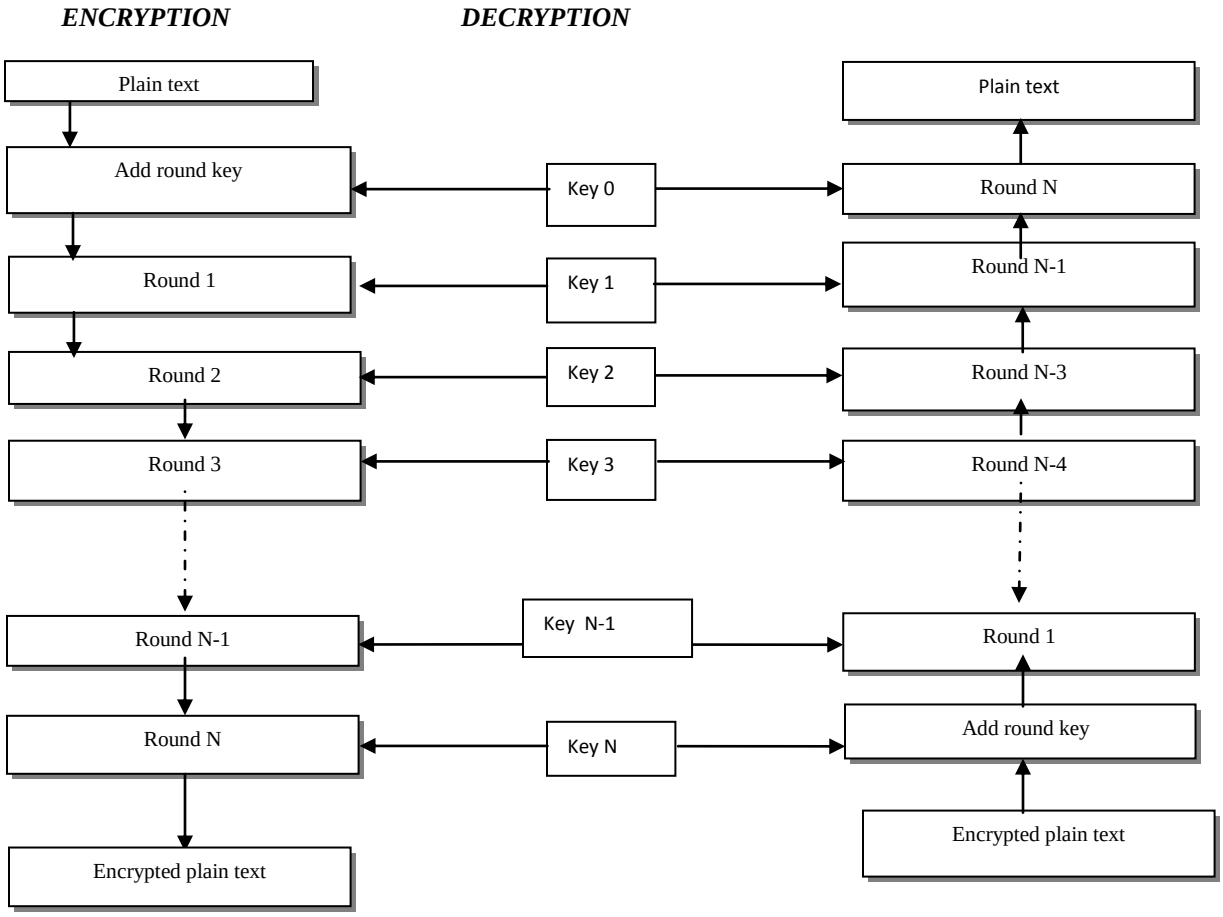


Figure 1.AES algorithm flow chart.

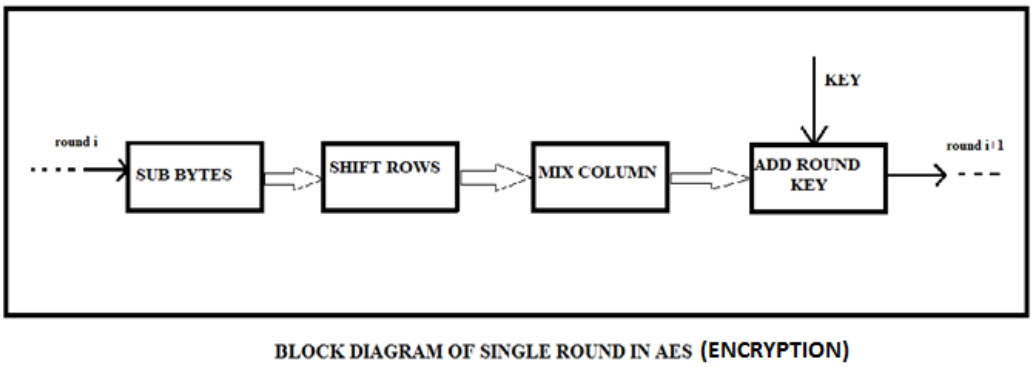
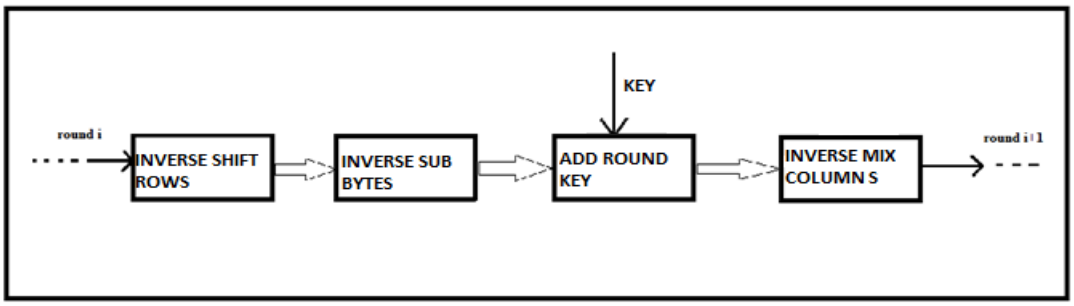


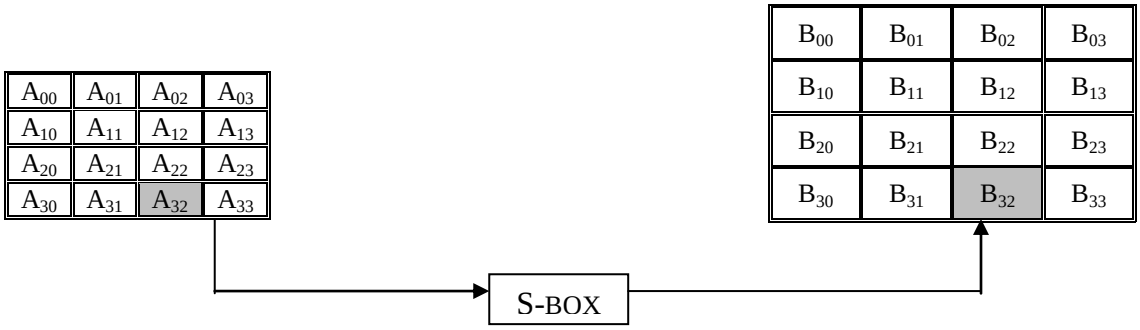
Figure 2. Block diagram of Single round in AES



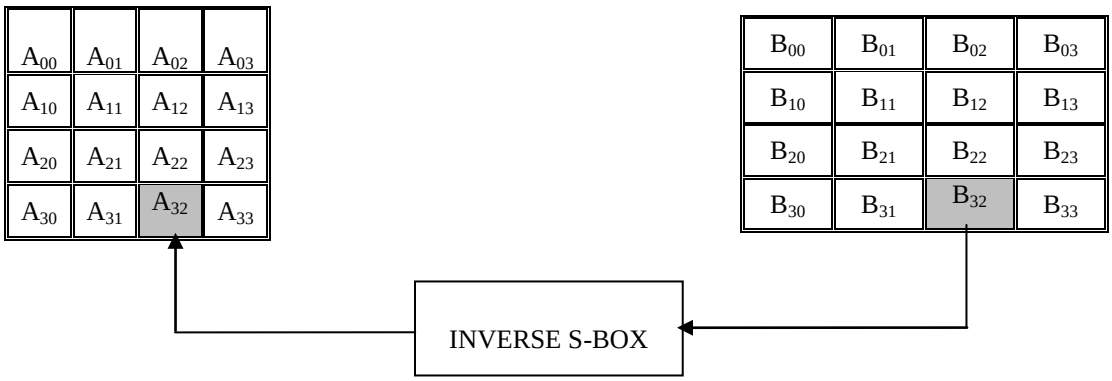
BLOCK DIAGRAM OF SINGLE ROUND IN AES (DECRYPTION)

Each round of AES consists of four transformations:-

1. ***Sub bytes****:* This block uses the S-box to perform byte by byte substitution of each byte of the state matrix in case of encryption and inverse S-box in case of decryption.

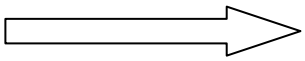


Example: if A_{32}= EC, then using the s-box transformation we get B_{32} = CE.
Using the inverse s-box and B_{32} as input we get A_{32} = EC.



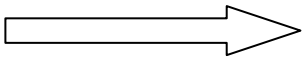
2. ***Shift rows****:*Here we perform shifting operation. The first row is left unchanged, in the second row one byte circular left shift is performed, in the third row two byte circular left shift is performed, and in the fourth row three byte circular left shift is performed as shown.

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}



C_{00}	C_{01}	C_{02}	C_{03}
C_{11}	C_{12}	C_{13}	C_{10}
C_{22}	C_{23}	C_{20}	C_{21}
C_{33}	C_{30}	C_{31}	C_{32}

AF	FE	08	47
EB	56	45	0D
82	BC	34	AE
99	01	0F	EC



AF	FE	08	47
56	45	0D	EB
34	AE	82	BC
EC	99	01	0F

The similar process is repeated for inverse shift rows by performing shifts to the opposite direction (right shifts). The first row is left unchanged whereas in the second, third, fourth rows one, two, three byte circular right shifts are performed respectively.

3. Mix Columns: Here Galois field multiplication (GF 2^8) is performed. It is finite field multiplication. In finite field multiplication the operands belong to a finite range of number and the result of the multiplication operation also results in a value within the finite field.For encryption the state matrix is multiplied with

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

i.e., the following operation is performed:-

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} C_{00} & C_{01} & C_{02} & C_{03} \\ C_{10} & C_{11} & C_{12} & C_{13} \\ C_{20} & C_{21} & C_{22} & C_{23} \\ C_{30} & C_{31} & C_{32} & C_{33} \end{pmatrix} = \begin{pmatrix} D_{00} & D_{01} & D_{02} & D_{03} \\ D_{10} & D_{11} & D_{12} & D_{13} \\ D_{20} & D_{21} & D_{22} & D_{23} \\ D_{30} & D_{31} & D_{32} & D_{33} \end{pmatrix} \dashrightarrow \boxed{1}$$

Similarly for decryption the state matrix is multiplied with

$$\begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \begin{pmatrix} D_{00} & D_{01} & D_{02} & D_{03} \\ D_{10} & D_{11} & D_{12} & D_{13} \\ D_{20} & D_{21} & D_{22} & D_{23} \\ D_{30} & D_{31} & D_{32} & D_{33} \end{pmatrix} = \begin{pmatrix} C_{00} & C_{01} & C_{02} & C_{03} \\ C_{10} & C_{11} & C_{12} & C_{13} \\ C_{20} & C_{21} & C_{22} & C_{23} \\ C_{30} & C_{31} & C_{32} & C_{33} \end{pmatrix} \dashrightarrow \boxed{2}$$

Rules of multiplication in Galois field:

a) Multiplication by 02: Check the MSB of the byte being multiplied. If it is '1' then shift it to the left by '1' bit and XOR with '1B', else just perform shift left by '1' bit.

b) Multiplication by 03: '03' can be split up as '02+01'. Hence we make use of the previous stated rule for '02' and then XOR the answer with the current byte value.
For decryption we make use of look up table method(LUT method).

*4. Add round key:*Each of the 16 bytes of the state matrix is XORed against each of the 16 bytes of a portion of the expanded key for the current round. Similar process is repeated for decryption.

Key expansion algorithm takes a 16 byte key an input and produces an array of 156 bytes. It was designed to be resistant to known cryptanalytic attacks.
In general,this algorithm takes an input key of 'k' bytes where k may be 4, 6 or 8.The output is 16*(Nr+1) bytes. This is the expanded key.Nr refers to the number of rounds.

III LOOK UP TABLE (LUT) Method

The LUT is used in encryption and decryption using AES algorithm. It is used in the mix columns block where Galois field multiplication is performed. Galois field multiplication is a finite field multiplication. Here the finite field is limited to 2^8.

For encryption, the state matrix is multiplied with

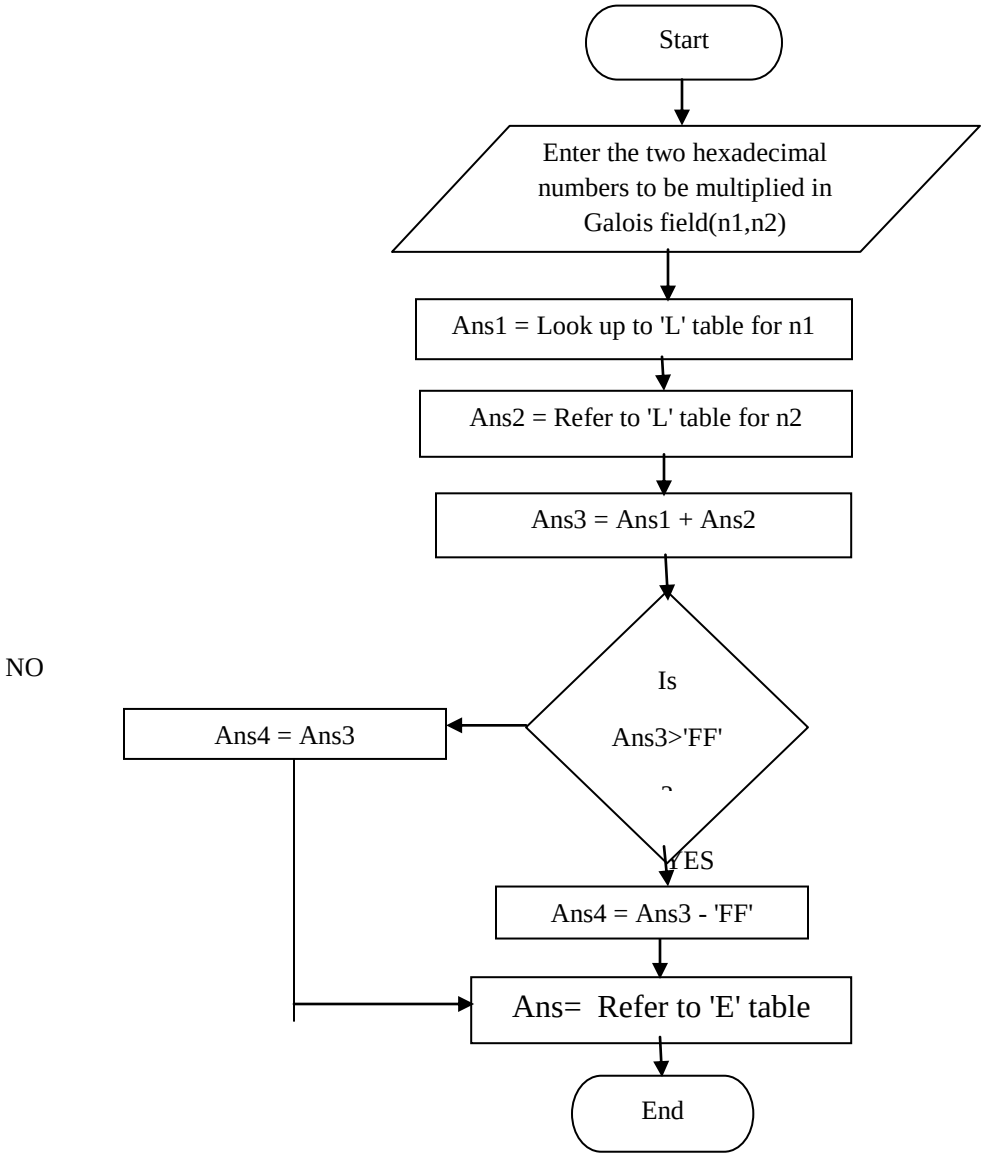
$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

For decryption, the state matrix is multiplied with

$$\begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix}$$

There are two "look up" tables. "Look up" tables are pre-computed tables that are used to compute multiplication in Galois field(2^8) for any input value. The 'L' table and 'E' table. We first refer to the 'L' table and then we refer to the 'E' table as shown in the example.
Example
Consider the hexadecimal numbers '47' and '0E'.

First we refer to the L table and find the L value of '47' = '94' = Ans1.
We once again refer to the L table and find Lvalue of '0E' = 'DF' = Ans2.
Now we do the following computation Ans3 = Ans1 + Ans2 = '94' + 'DF' = '173'.
Since '173' is greater than 'FF' we subtract 'FF' from it i.e., Ans4='173' - 'FF' = '74'.
Now we refer to E table i.e., look up of '74' in E table is '87'= Ans.
Therefore the summarized steps for two hexadecimal numbers are L (n1) + L(n2)=Ans3.
Check for Ans3>'FF', if greater Ans4= Ans3 - 'FF' else Ans4=Ans3.
Ans= E(Ans4).



Disadvantages of LUT method

1. More memory is required to store the two look up tables.
2. Computation time is more since we refer two tables for calculating one product.

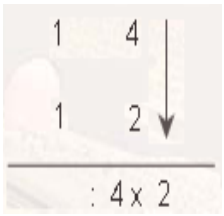
IV PROPOSED APPROACH

Vedic Mathematics[5]is an ancient form of Mathematics which existed in ancient India in 1500 B.C. But was rediscovered by Sri Bharthi Krishna Tirthaji between 1911 and 1918.He bifurcated the entire subject into 16 mathematical Sutras/formulae. These Sutras are easy to understand and fast in terms of computation. Of which, UrdhwaTiryakbhyam Sutra is one of the most popular sutra used to multiplication.

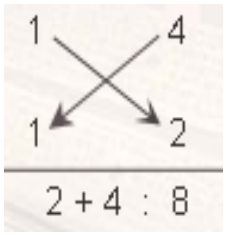
UrdhwaTiryakbhyam technically means “vertically crosswise”. Therefore each step of multiplication involves multiplication of the extreme digits.After which the results of all the stages are concatenated in order to arrive at the final result of the product.

In a regular Urdhwa sutra multiplier, partial products are obtained by ANDing the inputs and addingthe bi-products. In our approach, we replace addition with XORing. The product obtained is limited to 8 bits using conventional methods of Galois field restriction. The addition of XOR gates reduces the complexity of the existing Vedic Mathematics Architecture.

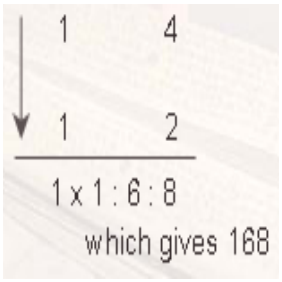
Step1



Step 2



Step 3



V RESULTS & CONCLUSION

The look up table approach & our proposed architecture was implemented on Spartan 3e series of FPGA and the synthesis results for the same is as shown below.It is clearly observed the LUT approach is 3 ns faster than our proposed architecture. Yet, it is not preferable sincemore than 100% of the FPGA resources are utilized.On the other hand, the Vedic Mathematics approach occupies only 6% of area when implemented on a Spartan 3e series of FPGA.

TABLE I RESULTS

Algorithm	**Timing(ns)**	**Area Occupancy(%)**
Look-up Table Approach	12.458	More than 100% of the device resources used
Proposed Architecture	15.367	6.08

REFERENCES:

1. An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists by Elbirt, A.J. ; Dept. of Electr. &Comput. Eng., Worcester Polytech. Inst., MA, USA ; Yip, W. ; Chetwynd, B. ; Paar, C.Very Large Scale Integration (VLSI) Systems, IEEE Transactions on (Volume:9 , Issue: 4) , Aug. 2001,pp:545 - 557

2. Mix/InvMixColumn decomposition and resource sharing in AES byIyer, N.C. Dept. of E&C, BVBCET, Hubli, India Deepa ; Anandmohan, P.V. ; Poornaiah, D.V.International Conference on Industrial and Information Systems (ICIIS), 2010, July 29 2010-Aug. 2010, pp-166 – 171.

3. High-Speed AES EncryptorWith Efficient Merging Techniques by Hammad, I. ; Dept. of Electr. &Comput. Eng., Dalhousie Univ., Halifax, NS, Canada ; El-Sankary, K. ; El-Masry, E.Embedded Systems Letters, IEEE (Volume:2 , Issue: 3), Sept. 2010, pp-67 - 71

4. Realization of a resource sharing fast encryption and decryption AES algorithm, by Jun ShuYiwen Wang ; Wenchang Li ; ZhiyongGanIntelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on 6-8 Dec. 2010, pp-1 – 4.

5. Area and Speed Efficient Arithmetic Logic Unit Design Using Ancient Vedic Mathematics on FPGA, Sushma R. Huddar, SudhirRaoRupanagudi, VenkateshJanardhan, Surabhi Mohan, S. Sandya, Third International Conference on Advances in Computing, Communication, and Control communications in computer and information science ICAC3 2013, Mumbai, India, January 18-19, 2013. Proceedings vol. 361 ,2013,pp-475-483.

6. Jagadguru Swami Sri BharatiKrisnaTirthaji Maharaja: Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda, pp. 5–45. MotilalBanarasidas Publishers, Delhi (2009)