ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A NOVEL ARCHITECTURE FOR INVERSE MIX COLUMNS OPERATION IN AES USING VEDIC MATHEMATICS

Shrita G, Basavaraj S M
Department of Telecommunication, Dayananda Sagar College of Engineering, Bangalore, India

ABSTRACT

In the fast growing world, with the advent of new technologies, people communicate via internet on a day to day basis. Secure transactions such as banking, bill payments, mail delivery, etc. are being carried out easily via smart phones, tablets and computers. Thus, providing information security through encryption and decryption plays a very important roll. Many algorithms have been implemented so far to provide data encryption, of which Advanced Encryption Standard (AES) is one such efficient algorithm. In this paper, a novel method has been proposed for the mix columns and inverse mix columns operation in AES cryptography, which is a major operation that provides diffusion of data i.e. the plain text. A software implementation is done using VERILOG Hardware Description Language, using the three methods: Look-up table method, Splitting method over Galois field and the proposed, Vedic mathematics technique. It can be found that the Vedic mathematics approach provides an area efficient and high speed algorithm when compared to the other two methods

Keywords: Vedic mathematics, mix column, inverse mix column, look-up table, Galois field, AES, encryption, Verilog

INTRODUCTION

In a fast paced world, communication has been made available at everyone's finger tips by the use of computers and internet. With the advances in technology, all transactions such as financial, bill payments, exchange of credible information via mails and messages can be performed with ease, via computers, handheld devices such as mobile phones, tablets, etc. This gives rise to the need for an area efficient and high speed cryptographic algorithm, which provides encryption and decryption of the data being exchanged.

When critical or secret information is being exchanged between two parties, there is always the possibility of an opponent/enemy who is trying to hack the information to misuse it. In order to prevent this, various encryption and decryption algorithms have been developed. The Advanced Encryption Standard (AES) is one such efficient algorithm. AES performs four major operations, namely: Substitute bytes, Shift rows, Mix columns and Add-around key, of which, the mix columns operation plays a major role in inducing diffusion to the message being exchanged.

In this paper, a novel architecture has been proposed for the mix columns and inverse mix columns operations in AES cryptography. The inverse mix columns operation has been implemented using the two known methods: Look-up table method, Splitting method over Galois field, and the proposed: Vedic mathematics technique; and the results have been compared.

BASIC PRINCIPLE OF MIX COLUMNS AND INVERSE MIX COLUMNS OPERATION

The mix columns transformation is a major operation in AES cryptography, which is used to induce Diffusion, which refers to dissipating the structure of plaintext over bulk of cipher text, thus making it difficult to hack. The inverse mix columns operation is the inverse transformation performed on the cipher text, to obtain back the plaintext during decryption. In both the operations, a predetermined matrix is used to perform the transformation.

Mix columns transformation

The mix columns transformation is obtained by performing matrix multiplication of the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

predetermined matrix with the data matrix that needs to be transformed. The data matrix for AES consists of two digits (i.e. 8 bits) hexadecimal numbers.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Matrix multiplication is performed column vice on the data, i.e. the predetermined matrix is multiplied with each column of the data matrix separately, to obtain the corresponding columns of the transformed matrix, as shown in Fig. 1.

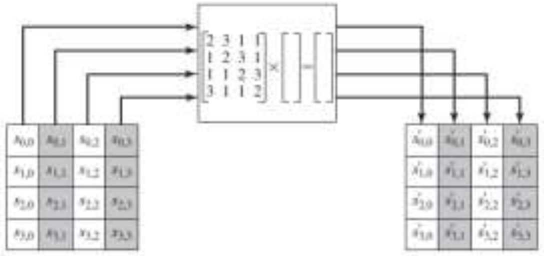

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2
×
=
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig. 1: Mix Columns Transformation

Inverse Mix Columns Transformation

The inverse mix columns transformation is obtained by performing matrix multiplication of the predetermined matrix with the cipher text matrix, given by:

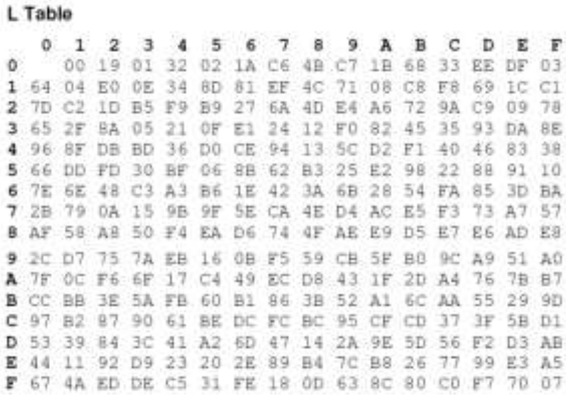
$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

IMPLEMENTATION OF MIX COLUMNS AND INVERSE MIX COLUMNS OPERATIONS OF AES

Encryption and decryption of data are implemented using the Look-up table method, Splitting method over Galois field and the Vedic mathematics technique.

Look – Up Table method

In this method, we make use of two tables: L-table and E-table respectively, as shown in Fig. 2 and Fig. 3. The product of the two hex numbers is obtained by: first performing an L table look-up of both the numbers, adding the resultant values, followed by an E table look up of the resultant.

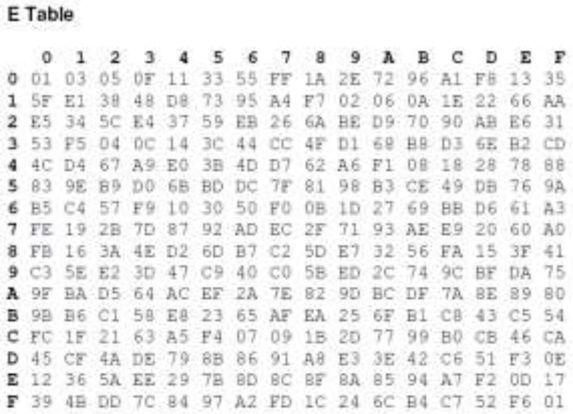


L Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

Fig. 2: L Table

While performing look-up, the tenth's place digit is looked up on the vertical index and the unit's place digit is looked up on the horizontal index. For example: the L-table look-up for A1 would be 0C and the E-table look-up would be BA, in reference with Fig. 2 & 3.



E Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	BF	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

Fig. 3: E Table

Splitting method over Galois Field

In this method, the mix columns operation is performed using GF (2^8) operations. Each element of GF (2^8) is represented as polynomial of degree 7. The coefficients of each term of the polynomial can take the value of either 0 or 1. For example, 10010110 can be represented as $x^7 + x^4 + x^2 + x$.

Addition of two elements in GF (2^8) is performed using XOR gates, to add corresponding bits. Multiplication in GF (2^8) is performed by multiplying each term of one polynomial with all of the terms of the second polynomial. Each of these products should be added together. If the degree of the resultant polynomial is greater than 7, then it must be reduced to an irreducible polynomial. In the case of AES, the irreducible polynomial is $x^8+ x^4+ x^3+x+1$. Reducing a higher degree polynomial to an irreducible form is accomplished by multiplying the irreducible polynomial by x^{i-8}, where i is the degree of the polynomial that is to be reduced. Then, adding the multiplied irreducible polynomial to polynomial to be reduced. The process is continued, till we obtain a polynomial with a degree lesser than or equal to 7. An example is shown in Fig. 4 & Fig. 5.

$$\begin{array}{lr} & x^4 + 1 \\ \times & x^5 + x^4 \\ \hline & x^8 + x^4 \\ + & x^9 + x^5 \\ \hline & x^9 + x^8 + x^5 + x^4 \end{array}$$

Fig. 4: Multiplication of 2 polynomials

$$\begin{array}{lr} & x^9+ x^8+ x^5+ x^4 \\ + & x^9+ x^5+ x^4+ x^2+ x \\ \hline & x^8+ x^2+ x \\ + & x^8+ x^4+ x^3+x+1 \\ \hline & x^4+ x^3+ x^2 + 1 \end{array}$$

Fig. 5: Reduction of resultant polynomial.

The multiplication of two elements of GF (2^8) can also be expressed as a linear combination of products of the first element and a single-termed polynomial in the Galois Field, as multiplication is distributive over addition. For example, we know that 0E = 08 + 04 + 02, therefore, we can express (a * 0E) as (a * 08) + (a * 04) + (a * 02), for any a ∈ GF (2^8).

Example: 02* 0E = (02 * 08) + (02 * 04) + (02 * 02)

= 10 + 8 + 4

= 1C

Vedic Mathematics Technique

In this approach, we make use of the Urdhva Tiryagbhyam multiplication technique which is a general formula applicable to all cases of multiplication and also division. This technique has been proven to be the most efficient in terms of speed. 'Urdhva Tiryagbhyam' means 'vertically and cross-wise'.

To perform bit-wise multiplication of two 2-digit numbers, the general approach is as shown in Fig. 6.

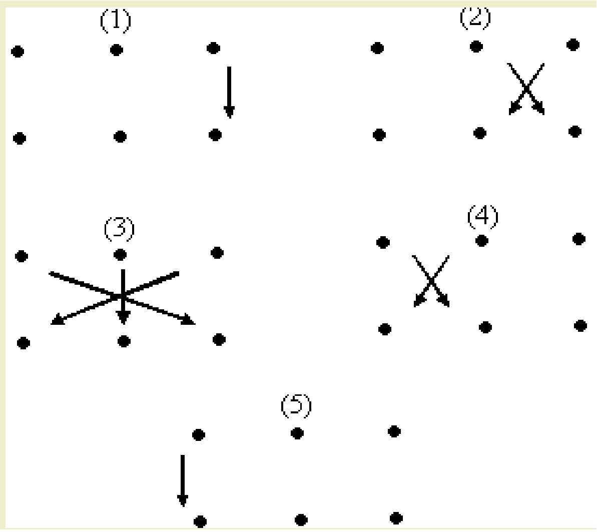


Fig. 6: General concept of Urdhva Tiryagbhyam multiplication

Applying the above concept, the two hex numbers to be multiplied are converted to binary and the following equations are obtained:

Say the two numbers are expressed as:

(a7 a6 a5 a4 a3 a2 a1 a0) and (b7 b6 b5 b4 b3 b2 b1 b0)

y0 = a[0] and b[0];

y1 = (a[1] and b[0]) xor (a[0] and b[1]);

y2 = (a[2] and b[0]) xor (a[0] and b[2]) xor (a[1] and b[1]);

y3 = (a[3] and b[0]) xor (a[0] and b[3]) xor (a[2] and b[1]) xor (a[1] and b[2]);

y4 = (a[4] and b[0]) xor (a[0] and b[4]) xor (a[3] and b[1]) xor (a[1] and b[3]) xor (a[2] and b[2]);

y5 = (a[5] and b[0]) xor (a[0] and b[5]) xor (a[4] and b[1]) xor (a[1] and b[4]) xor (a[3] and b[2]) xor (a[2] and b[3]);

y6 = (a[6] and b[0]) xor (a[0] and b[6]) xor (a[5] and b[1]) xor (a[1] and b[5]) xor (a[4] and b[2]) xor (a[2] and b[4]) xor (a[3] and b[3]); and so on…

Finally, these outputs are concatenated to obtain the product of the two numbers.

SIMULATION AND RESULTS

Simulation has been performed using ModelSim SE 64 simulator for various sets of 16 8-bit hexadecimal values that are given as input. The correctness of the encrypted and decrypted values have been tested and verified.

The simulation results obtained are as follows:

Look – Up Table method

Disk Utilization Details:

Area of LTable = Number of Slices: 128 out of 960 13%

Area of ETable = Number of Slices: 66 out of 960 6%

Area of SubByFF = Number of Slices: 10 out of 960 1%

Since we will be accessing the LUT VERILOG module/code 16 times for each column of input matrix and since there are 4 columns, we will be accessing it 64 times. Thus the total FPGA area required would be:

Total Area = 20% * 64 = 1200%

The total area needed by the LUT method is excessively huge and beyond the FPGA.

Timing Details:

Timing of one LTable = 8.126ns

Timing of ETable = 8.126ns

Timing of SubByFF = 8.277 ns

Max. Timing = 24.529 ns (Clock frequency: 40.76 MHz)

Splitting Method over GF

Disk Utilization Details:

Area of mulWith0E = Number of Slices: 5 out of 960 0%

Area of mulWith0B = Number of Slices: 5 out of 960 0%

Area of mulWith0D = Number of Slices: 6 out of 960 0%

Area of mulWith09 = Number of Slices: 4 out of 960 0%

Total Area = (30 out of 960) * 16

Total Area = 320 out of 960 = 33 %

Timing Details:

Timing of multiplyWith0E = 6.193ns

Timing of multiplyWith0B = 6.988ns

Timing of multiplyWith0D = 6.819ns

Timing of multiplyWith09 = 5.934ns

Total Timing = 25.934 ns (Clock frequency: 38 MHz).

Vedic Mathematics Approach

Disk Utilization Details:

Area of vedicMath = Number of Slices: 5 out of 960 0%

Total Area = (5*64) = 320 out of 960 = 33%

Timing Details:

Timing of vedicMath = 12.359 ns (Clock frequency: 80 MHz)

On comparison of the results of the three methods, it can be seen clearly that, Vedic Math approach provides 100% area efficiency compared to LUT approach and 2 times increase in speed compared to both LUT and splitting method.

The results and discussion may be combined into a common section or obtainable separately. They may also be broken into subsets with short, revealing captions.

CONCLUSION

A novel and an area efficient architecture for performing mix column and inverse mix column operation in AES, has been proposed. A software implementation has been done using VERILOG Hardware Description Language, of the three methods, namely: Look up table method, Splitting method using Galois field multiplication and the proposed Vedic mathematics technique. A 100% area efficiency and a 2 times increase in speed has been achieved by the proposed novel Vedic math algorithm, in comparison with two other popular implementations of the same.

ACKNOWLEDGEMENTS

I would like to thank my college, Dayananda Sagar College of Engineering for giving me the opportunity to persue my academic interests. I would like to thank Mr Basavaraj SM for his valuable guidance. I would also like to express my deepest gratitude to Mr Sudhir Rao and my family for their immense support at every stage of the project.

REFERENCES

[1] "Advanced Encryption Standard by Example", by Adam Berent, ABI Software Development, Canada, 2009.

[2] "An Efficient Architecture for the AES Mix Columns Operation", by Hua Li and Zac Friggstad, Dept. of Math. & Comput. Sci., Lethbridge Univ., Alta., Canada, May 2005.

[3] "Low power and high speed AES using mix column transformation", by Balamurugan J and Logashanmugam E, St. Peter's Univ., Chennai, India, July 2013.

[4] "Optimized AES algorithm using Galois field multiplication and parallel key scheduling", by Jay Dalal D, Safiya Dalaya S and Nehal Shah, Electronics and Communication Department, Sarvajanik College of Engineering and Technology, Surat, India, December 2012.

[5] "Mix/InvMixColumn decomposition and resource sharing in AES", by NC Iyer, Deepa Anandmohan PV and DV Poornaiah, Dept. of E&C, BVBCET, Hubli, India, 2010.

[6] "FPGA implementation of AES encryption and decryption", by AM Deshpande, MS Deshpande and DN Kayatanavar, Dept. of Electron. & Telecommun. Eng., SRES Coll. of Eng., Kopargaon, India, 2009.

[7] "A Low-cost and High Efficiency Architecture of AES Crypto-engine", by Yan Qing Zhong, Jian Ming Wang, Yu DY and Zang ZF, Vinno Technol. Inc., Beijing, China, 2007.

[8] "High speed and efficient Vedic multiplier", by Kunchigi V, Kulkarni L and Kulkarni S, Jawaharlal Nehru Technol. Univ., Hyderabad, India, 2012.

AUTHOR BIBLOGRAPHY

	Shrita G Pursuing MTech degree in Digital Communication and Networking specialization, from Dayananda Sagar College of Engineering, Bangalore. Completed B.E. in the filed of Electronics & Communication, from SJB Institute of Technology, Bangalore. Interest of research lies in digital communication, wireless communication, cryptography and network security. Has published a paper on Wireless Communication in an International Journal. She has also presented a paper in a National Conference.
	Basavaraj S M Working as an Assistant Professor in the Dept. Of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore. Completed B.E. from NIT, Bhopal in the filed of Electronics & Communication and MTech from PESIT, Bangalore with the specialization in VLSI Design. Interest of research lies in embedded systems, VLSI design and analog CMOS. He has published paper on embedded systems.