APPLICATION OF UT MULTIPLIER IN AES ALGORITHM AND ANALYSIS OF ITS PERFORMANCE

Bindu Swetha Pasuluri [1], V.J.K.Kishor Sonti [2],
[1] *Research Scholar, Department of Electronics and Communication Engineering*
Sathyabama Institute of Science & Technology , Chennai, India
binduswetha.ece@gmail.com
[2] *Assoc .Professo, Department of Electronics and Communication Engineering*
Sathyabama Institute of Science & Technology , Chennai, India
kishoresonti.ece@sathyabama.ac.

Abstract: From the previous few years numerous cryptographic algorithms have been implemented, giving scrupulous significance to high safety applications, i.e. for ATMs, smart cards, WWW servers & many others. Amid the specified cryptographic algorithms, the Advanced Encryption Standard (AES) algorithm is preferred algorithm. The algorithm is implemented in various bit sizes. An AES algorithm recognizes a 128-bit plain data text and generates a 128-bit cipher text under the secret key control of 128, 192 or 256-bits. Moreover in this brief, an AES algorithm with 128/192/256 bits is implemented by using Vedic Mathematics. The conventional pipelined based algorithm is compared with proposed Vedic mathematics based AES algorithm in turns of area. With the help of Xilinx the AES algorithm is simulated and synthesized. Also it can be seen that proposed algorithm occupies approximately 40% less area when compared with conventional algorithm.

***Keywords*: AES algorithm, Vedic Mathematics, chipper text, area.**

1. Introduction

In a fast-paced globe, communication was produced accessible through the use of computers and internet at everybody's finger tips. With technology advances, all transactions such as economic transactions, bill payments, exchange of reliable data via mails and emails can be carried out with ease, via pcs, handheld devices such as mobile phones, tablets, etc. This creates the need for an area-efficient, high-speed cryptographic algorithm that offers the information being exchanged with encryption and decryption.

Several cryptographic algorithms have been found and studied over the previous few years, giving particular significance to the algorithm vulnerability issue in apps that require high safety, i.e. for WWW servers , ATMs, smart cards, etc. In these algorithms, one of the most perfect algorithms is the Advanced Encryption Standard (AES) algorithm because it is resistant to assaults.

Advanced Encryption Standard (AES) is definitely the default option for the purpose of data encryption in networked apps, the recent encryption standard authorized by NIST. Implementing the algorithm's hardware provides better efficiency but provides less suppleness and it is also hard and takes time to execute when compared with the implementing software. A call for suggestions for a fresh symmetrical algorithm called the The National Institute of Standards and Technology (NIST), USA released Advanced Encryption Standard (AES) in 1997.Since 1976, the DES algorithm has been the standard for symmetric algorithms.. Fifteen candidate algorithms were accepted in 1998 and five of these candidates were announced as finalists after one year of research[1,2].

Extensive study has been performed on all these algorithms to discover attacks or weaknesses. All 5 finalists seem to give sufficient safety, according to NIST. A lot of studies has also been performed to evaluate the performance in both software and hardware of these 5 algorithms. In 2000, NIST announced the selection of Rijndael to achieve AES cryptographic algorithm. The blend of safety, effectiveness, flexibility and execution have made Rijndael proposed algorithm as an appropriate choice for the AES.

Some design requirements had to be met by the applicants for the AES algorithm are listed. First of all, the algorithm must be a balanced algorithm and unaffected to all the attacks known. Further, the AES algorithm must be very well-organized in efficiency and memory for separate devices. It is necessary to handle the design with simple and different significant lengths of 128, 192 and 256 bits each. The block cipher's length should be 128 bits[4].

In this brief , sections II deliberates the importance of AES algorithm , Section III discusses the Importance of Vedic Mathematics and implementation of AES algorithm ,Simulation Results are deliberated in Section IV and followed by Conclusion in Section V.

2. Significance of AES Algorithm:

The AES recognizes a plain 128-bit text and generates a chipper text of length 128-bit with the help of a secret key

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

control using 128,192 or 256-bits. It is a network design of substitution-permutation with a single set of steps named as a round. A number of runs depends on the key length of the AES algorithm during implementation of the algorithm.

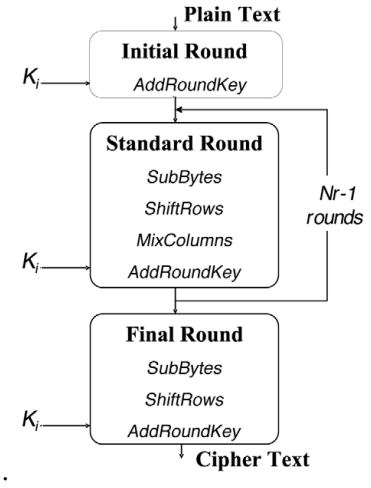

Figure 1: Process flow diagram of AES algorithm[2]

The procedure of AES algorithm is first started with an initial round of taking plain text input and then there are several ordinary rounds, and the final round ends to give the text.To calculate these rounds and a main timetable, only four distinct activities are required. In Rijndael, distinct keylengths can be used depending on the level of safety needed for the implementation. Rijndael is described as a 128, 192 or 256-bit block cipher with key lengths. The possible block lengths for the Rijndael algorithm are 128, 192 or 256. While the AES algorithm is exactly the same as Rijndael's, only one 128-bit block length is described. Although the AES algorithm is the same as the Rijndael, it offers only approximately 128 bit of detailed information [7],[8][11].

The Rijndael algorithm is such that all bits from 2 rounds are dependent on each bit, for example complete diffusion. The duration of the key relies on the number of running rounds.

	Length of the key in words	No. of rounds required-*Nr*
128 bit AES algorithm	4	10
192 bit AES algorithm	6	12
256 bit AES algorithm	8	14

Table 1: Key lengths of AES algorithm

The process of encryption is projected as:

- Firstly. Using S-box Substitution of Bytes should be done
- Next, with the help of different offset, Row shifting operation must be done
- Next, in each column of state array, mixing of data should be carried out
- Finally, with the help of state add a round key.

2.1 AddRoundKey

The AddRoundKey procedure is a easy state-to-RoundKey EXOR operation. The RoundKey is obtained with the help of the key schedule from the Cipherkey. RoundKey and state are identical in size and an EXOR procedure per component is performed to acquire the next State:

$S(i, j) = s(i, j) \oplus W(i, j)$.

Whereas s is the present state, S is denoted as the next state and finally w is designated as round key.

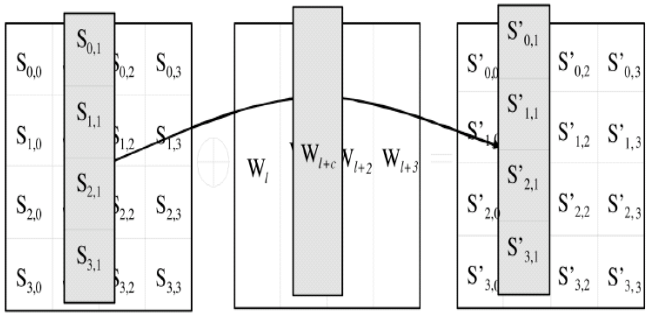

Figure 2: Process of Adding Round Key[2]

2.2 Procedure of Sub Bytes Data

The SubBytes strategy is similar to the DES algorithm's S-boxes. In the Rijindael algorithm, only one S-box has the key.In differential, linear cryptanalysis, and an assault on the algebraic handling of the S-box design criteria are extremely resistant.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
x	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 3: The architecture of AES S-Box architecture[2]

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

2.3 Shifting Rows

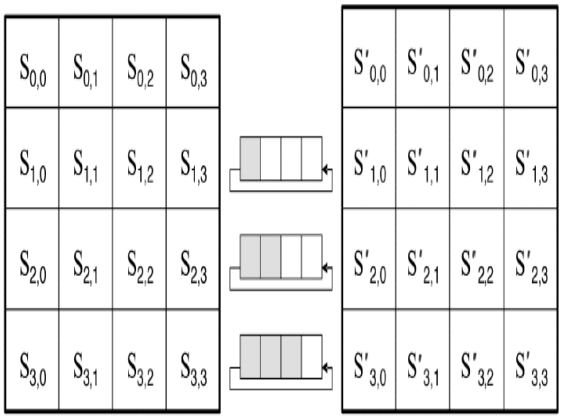

Figure 4: Process of Shifting Rows[2]

In ShiftRows, with distinct offsets, the state rows are cyclically changed. Here 1st is moved to c1 bytes , then bytes of 2nd row to c2 bytes, then the rows of 3rd bytes to c3 bytes. The block duration Nb depends on the values of c1, c2, and c3:

Nb	c1	c2	c3
4	1	2	3
6	1	2	3
8	1	3	4

Figure 5: block duration Nb[2]

2.4 MixColumns

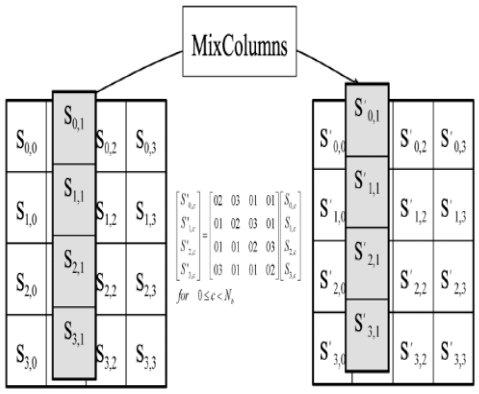

Figure 6: Process Showing MixColums Procedure[2]

The transformation of the MixColumn is an operation on various columns. The columns in the current state are seen as polynomials above GF[3][9] for the purposes of calculating the transformation of the Mix Column.

2.5 Process of Key schedule

In AES algorithm, RoundKeys are obtained through a key schedule from the CipherKey.

The quantity of RoundKeys needed for encrypting a database varies according to the block size and the duration of a key as the number of rounds is determined. 11 round keys (1 in the initial round, 9 in standard rounds, 1 in the final round,) must be set for a 128-bit block length.

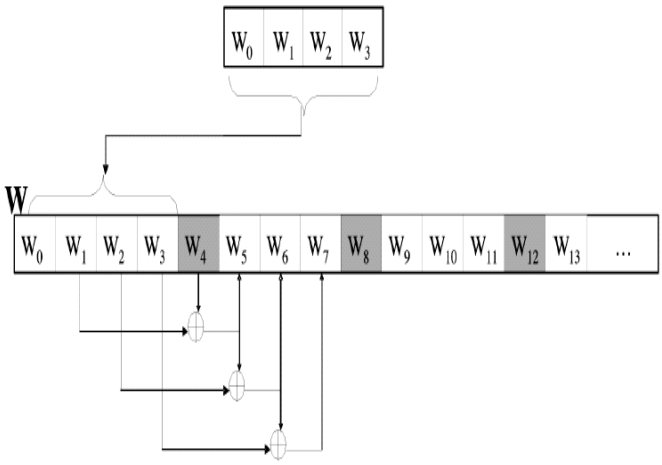

Figure 7 : Procedure of Key Scheduling[2]

2.6 Process for Decryption

2.6.1 The Inverse Cipher Prcoess:

The reversal of the following code is easy and is just the other way around. This section describes the entire inverse state cipher in detail.

A number of decryption steps are taken by the State[7][8]:

a. Firstly, Inversing of shift row has to be taken place

b. Subsequently, S-box conversion is carried out with Inverse Sub Bytes.

c. Next, mix column has to be inversed

d. Finally the sub key has to be inversed.

2.6.2 process of Inverse Shift row

The reverse motion of the shift line is the reverse of the encryption technique. The first line has remained the same, and the second row has a spot on the right, the 3rd row has two spots and the 4th row is three spots. Figure 11 for Nc= Nb= 4 explains the operation of the inverse correct change.

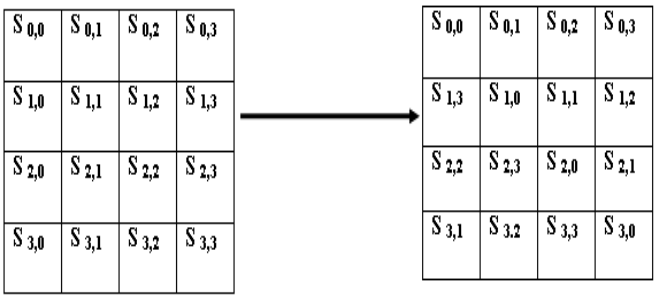

Figure 8 : Process of Inverse shift row[2]

2.6.3 Transformation of Inverse Sub bytes using S-Box

The reverse sub-byte conversion utilizes the reverse S-Box table given in Figure 9.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

x

Figure 9: Inverse S-Box table[2]

2.6.4 Inversion of Mix column

The reverse column is transformed on each column separately.

$$\begin{bmatrix} s'_{0c} \\ s'_{1c} \\ s'_{2c} \\ s'_{3c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0c} \\ s_{1c} \\ s_{2c} \\ s_{3c} \end{bmatrix} \quad \text{for } 0 \le c < Nb$$

Figure 10: Inversing mix column[2]

2.6.5 Inversion of Round Key

The reverse round key conversion is its own reverse. Every round key is produced for each round.

3. Vedic Multiplier

The architecture of the Vedic multiplier is based on the Vedic multiplication formula (Sutra). Traditionally, these Vedic sutras are used in the decimal number system to multiply the two digits. In this assignment, we use the same thoughts in order to formulate the suggested algorithm which is well suited for digital hardware. [5,6].

Vedic multiplication is discussed below based on certain algorithms:

3.1 Urdhva Tiryakbhyam Sutra:

The architecture of the multiplier was based on the algorithm of ancient Indian Vedic mathematics, Urdhva-Tiryakbhyam (Vertical and Crosswise). UrdhvaTiryakbhyam Sutra is a widely used multiplication formula in every case. It is based on the idea by which all partial products can be produced simultaneously by adding such partial products. The parallelism in the development and summarization of partial goods is described in Figure 7 using Urdhava Triyakbhyam. For the amount of nx n bit the algorithm can be generalized. The multiplier is independent of the clock frequency of the processor by calculating the partial products together with their amount. The multiplier must therefore simultaneously calculate the object and is thus separate from the clock frequency.

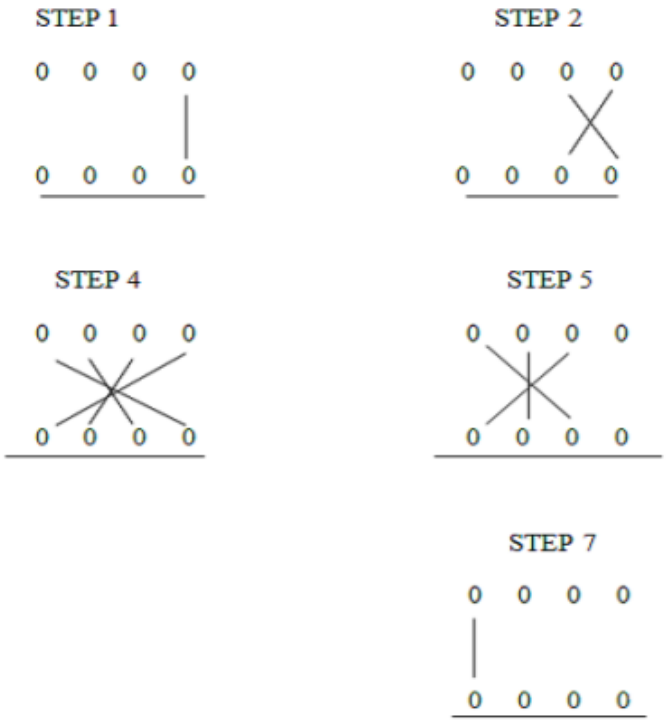

Figure 11: Example showing the multiplication process using Urdhava Triyakbhyam.[5,6]

4. Implementation of Vedic UT Multiplier AES architecture

AES defines a data encryption and decryption algorithm with the same key. The size of the block is 128 bits limited. The size of the key may be 128, 192, or 256 bits.

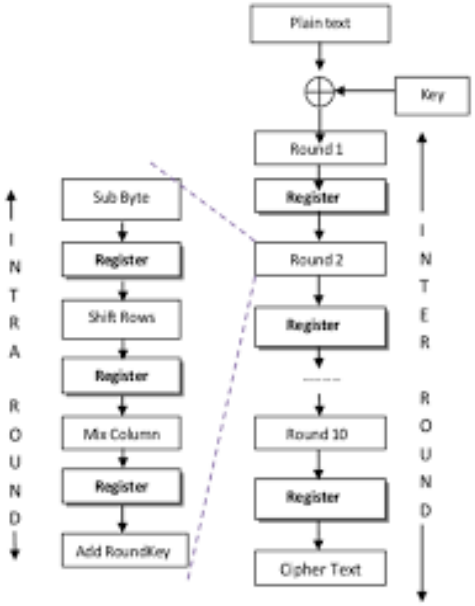

Figure 11: Architecture of Pipelined AES Algorithm[9]

AES works in a 4 byte, state-appointed matrix. The plaintext is the ultimate cipher text in some stages of conversion. Six rounds plus 32 different key sizes are available. In one round, the state is read in four 4 byte y0. y 1, y 2, y 3, and the

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

parameters are converted; the xor is read in a 16 byte round key; and the result is set to z0,z 1,z 2,z 3.

The plaintext should first be divided into distinct blocks and then encrypted in some operation mode by using a randomization-based additional initialization vector when aiming for a variable-length .

FIPS 81 specifies the cipher feedback (CFB) mode, output feedback (OFB) mode. NIST specifies the counter (CTR) mode in SP800-38A. The benefit of these methods is to use both encryption and decryption algorithms only. Other primary element besides the Encryption and Decryption module is the Key Expansion Schedule. The AES Encryption / Decryption Standard's safety factor relies primarily on this portion. For better safety, XORed with the initial plain / cipher text is the first round user key in the AES algorithm. And the Expanded Key from the Expanded Key Schedule will be XORed with information next round. The AES extension algorithm is set. To speed up the Key Generation process, choosing pipeline architecture is preferable.

5. Simulation Results:

The above AES algorithm is implemented for AES128 bit using Verilog HDL. All findings are based on Xilinx ISE tools simulations.The simulation results obtained are as follows:

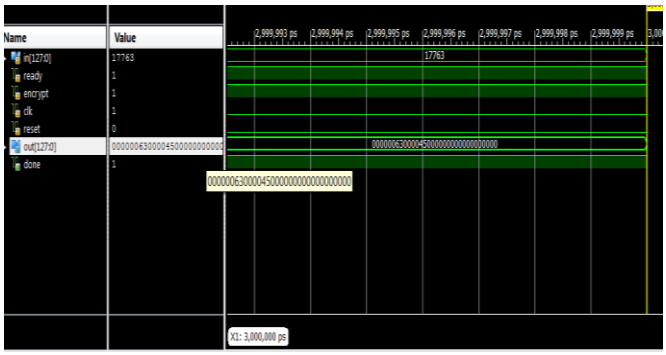

Figure 12: Simulation showing the shifting of rows

The state rows are altered cyclically in ShiftRows with separate offsets. The 1 line is shifted to c1 bytes, the 2-to-c2 bytes rows and the 3-to-c3 bytes rows. The duration of the block Nb depends on c1, c2, and c3 values..

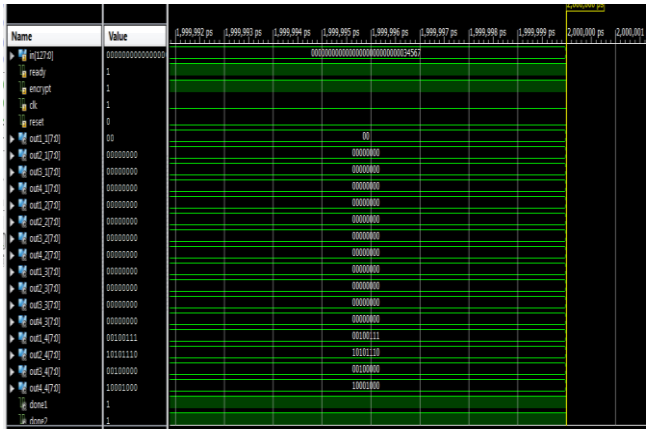

Figure 13 : The findings of the simulation indicating that the present state columns show the mixColumn conversion.

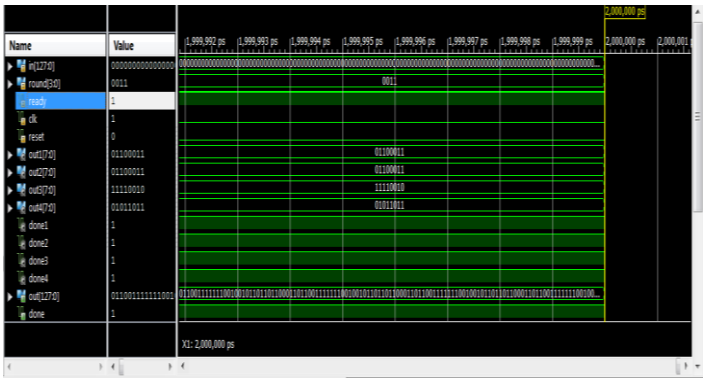

Figure 14: Simulation results showing the KeySchedule.

The RoundKeys required to encrypt one data block depends on the block size and the key length when the round quantity of the data block is determined. For a block length of 128 bits, 11 RoundKeys (1 for the original round, nine for the standard rounds and one for the final round) are required.

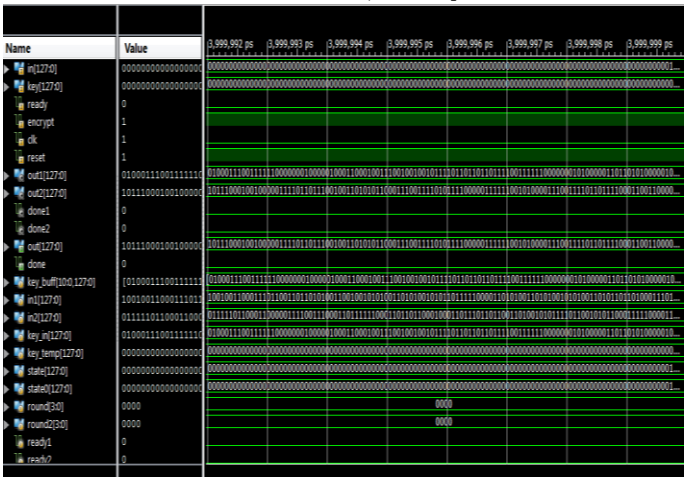

Figure 15: AES output with 128 bit

The conventional and Proposed Vedic Maths based AES algorithm with 128/192/256 bits are implemented on Vertex-6 FPGA board with device name as 6vhx565tff1924-2. After the process of synthesis the following results regarding the LUT are slices are depicted in the table below.

Table 2: Comparison of Area for AES Algorithm

Type /Bit of AES	**Number of Slice Registers**	**Number of Slice LUTs**
Conventional AES -128 bit	3968	3536
Conventional AES-192 bit	5280	4264
Conventional AES-256 bit	6848	6503

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Vedic Mathematics based AES-128 bit	1677	1642
Vedic Mathematics based AES-192 bit	2333	2297
Vedic Mathematics based AES-256 bit	3117	3085

From the above table it is clearly evident that vedic mathematics based AES algorithm uses less number of LUTs and thus occupies less area when compared with conventional type.

6. Conclusion

In the brief, we have simulated and synthesised conventional pipelined AES algorithm and vedic maths based AES algorithm by using Xilinx ISE tool. The AES recognizes a 128-bit plain text and generates a 128-bit cipher text under a 128,192 or 256-bit secret key control. The UT based vedic sutra has been implemented in the algorithm. When compared the proposed model occupies approximately 40% less area than the conventional method.The 128 bit AES algorithm occupied 1642 LUTs using Vedic multiplier where as conventional AES algorithm used 3536 LUTS. Similarly UT based 192-bit and 256- bit AES algorithm uses 2297,3085 LUTs respectively where as conventional method uses 4264 and 6503 LUTs respectively

References

[1] AES page available via http://www.nist.gov/CrypographyToolkit.

[2] Berent, Adam. "Advanced Encryption Standard by Example". Document available at URL http://www.networkdls.com/Articles/AESbyExample.pdf (April l 2007) Accessed: June 2013.

[3] Li, Hua, and Zac Friggstad. "An efficient architecture for the AES MixColumns operation". Circuits and Systems, 2005. ISCAS 2005. IEEE

[4] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," IEEE Trans. Very Large Scale Integrated. (VLSI) Syst., vol. 20, no. 2, pp. 371–375, Feb. 2012.

[5] H. D. Tiwari, G. Gankhuyag, C. M. Kim, and Y. B. Cho, "Multiplier design based on ancient Indian Vedic mathematics", Proc. Int SoC Design Conf., pp.65-68. 2008.

[6] S. Patil "Design of speed and power efficient multipliers using Vedic Mathematics with VLSI implementation"IEEE2014.

[7] Iyer, Nalini C., P. V. Anandmohan, and D. V. Poornaiah. "Mix/InvMixColumn decomposition and resource sharing in AES." International Conference on Industrial and Information Systems (ICIIS), on. IEEE, 2010.

[8] Berent, Adam. "Advanced Encryption Standard by Example.", Document available at URL http://www.networkdls.com/Articles/AESbyExample.pdf (Aprill 2007) Accessed: June 2013.

[9] Li, Hua, and Zac Friggstad. "An efficient architecture for the AES mix columns operation." Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on. IEEE, 2005.

[10] Iyer, Nalini, et al. "Efficient Hardware Architectures for AES on FPGA."Computational Intelligence and Information Technology. Springer Berlin Heidelberg, 249-257, 2011.

[11] Kumar, Saurabh, "VLSI implementation of AES algorithm",ethesis.nitrkl.ac.in,2013.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731