

A MODIFIED ARCHITECTURE DESIGN FOR ADVANCE ENCRYPTION STANDARD WITH ENHANCEMENT IN MIX COLUMNS FOR SECURE DATA USAGE

Gadagottu Jyosthna¹, L.Srinivas Reddy²

¹ M.Tech (VLSI) Scholar, Nalanda Institute of Engg. and Tech.(NIET),
Siddharth Nagar, Guntur, A.P. (India)

² Asst. Professor (ECE) from Nalanda Institute of Engg and Tech.(NIET),
Siddharth Nagar, Guntur, A.P. (India)

ABSTRACT

In this paper we are presenting an encryption algorithm called Advance Encryption Standard .We have designed AES algorithm using Verilog HDL and in this design we have used look up table substitution for byte in state matrix , also for low complexity and low latency hardware for efficient performance. This design was simulated in Xilins ISE 13.2, compared results with previous design performances.

Keywords: AES, Cipher, Shiftrows, Mix Columns,Lookup Table, ROM

I. INTRODUCTION

In earlier days Data encryption standard (DES) was considered as encryption standard with symmetrical key encryption with the keysize of 56 bits. After certain days 56 bit key was considered to be small and for high data bit systems require key and data size to be large. In the year 1990 the National institute of standards call for papers on new encryption methods. So many researchers sent their papers to NIST, out of allthose few were selected for testing. Cryptographic researchers after performing test on them only five are best among them ,those are Mars, RC6, Rijndael, Serpent and Twofish. These five went onto further testing afterperforming these tests they have declared that Rijndael algorithm was the winner. According this AES algorithm data and key size may be any size i.e muiltiple of 32 bits,with minimunm of 128bits and maximum of 256 bits. This algorithm also called Rijndael algorithm of AES. AES can be implemented in software and hardware. Software implementation require less resources and cost and its implicablity also limited ,having low seed. Nowadays we require large volume data and high speed requirements made it to implement ain hardware. Hard ware nothing but we have application specific IC and FPGA. FPGA is reconfigurable device which supports wide range of functionality than ASIC. So we prefer FPGAS to implement AES.

II. AES ALGORITHM

AES algorithm is using for encrypting the data and for decrypting the data. Encryption is nothing but converting data into unknown format called cipher text. Decryption iis converting the cipher into normal text called plain text. In AES we are using 128 bit block and data and keys.we will perform M_r number of iterations each

iteration is called as loop. M_r denotes number of loops. this may be depend upon the key size that may be 10, 12 or 14 with respective of size of key is 128,192,256. First we will discuss about Encryption later then decryption

2.1 Aes Encryption

As we discussed earlier it will operate on the 128bit block size of data and key. This is having following stages addroundkey, substitution byte, shiftrows, mix columns, are repeated in loop format depending upon of our key size. They key can be used for every loop is different and is derived from the original key, for that we have a key expansion algorithm. This algorithm will generate different keys for all rounds.

The proposed AES is similar to the conventional AES but the difference is in construction of S-box which was made by combinational gates. AES algorithm works on 4*4 matrix element called states. It works on states which is of 8bit length.

The state will undergo following stages namely sub bytes and inverse sub bytes, shift rows, and mix columns, transformations. The AES algorithm represented pictorially as below

2.1.1 Subbytes

This is a nonlinear transformation each byte in the state matrix was replaced by the precalculated data called substitution box S-box. This s-box was precalculated and stored in the rom called look up table. This method of implementation will reduce the latency and more prominently can be implemented in single clock cycle

2.1.2 Shiftrows

This stage having shifting of state rows cyclically leftside. That means each row is shifted by some offset, first row is unchanged, second row shifted by one bit, third row shifted by two bits, last row shifted by three bits.

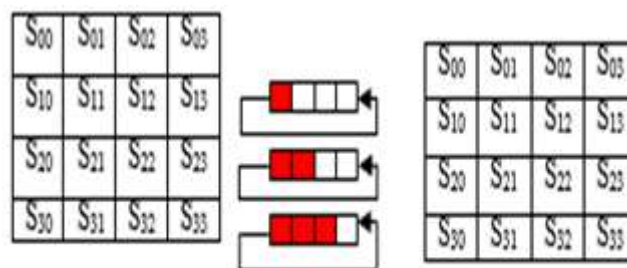


Figure 2.1 Shiftrows

2.1.3 Mix Columns

In this stage columns in the state matrix was considered as polynomial over galois field. This polynomial is multiplied by the modulo X^4+1 with a fixed polynomial $d(x)$.

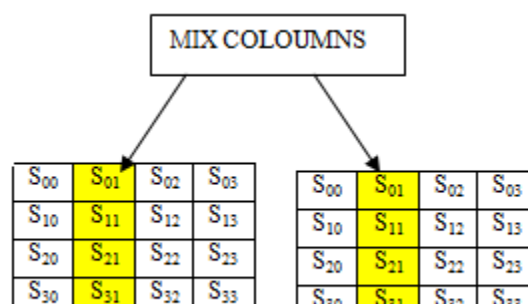


Figure 2.2 Mix Columns

2.1.4 Addroundkey

In this round state byte is added to the key which was derived from main key. This addition is modulo2 addition or xor ing the key with byte. This may be happen in 10 or 12 rounds , for each round new key is used.

2.2 AES Decryption

This is simply reverse process to the encryption algorithm which converts cipher text to plain text i.e to readable format. It uses exactly inverse algorithm used for encryption.

2.2.1 Add Round Key

In this stage we are performing modulo 2 addition or xor operation with the sub byte. Xor operation is self inverse so we will again perform same operation but keys are selected in reverse order.

2.2.2 Inv Shift Rows

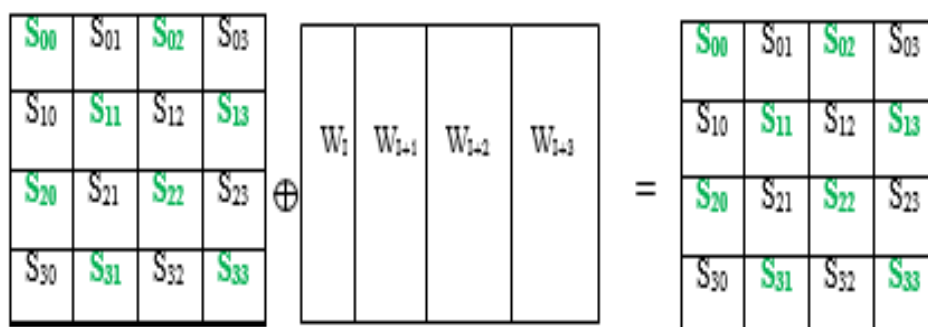


Figure 2.3 Inv Shift Rows

In this round exactly reverse process for the shift rows were performed i.e first row will not be altered, second and third and final rows are shift by one ,two ,and three positions respectively.

2.2.3 Inv Subbyte

In this round the byte which was result of previous operations was replaced according to the precalculated inv sub byte table called inv s-box table. This table has all the values from 0 to 255 and their respective replacement data. According to that data it will be transformed.

2.2.4 Inv Mix Columns

In this round polynomial of state matrix over galois field , which are having degree less than 4 are coefficients of state are multiplied with modulo X^4+1 and with a fixed polynomial $e(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$. enhances the performance and throughput of encryption at faster rate.

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	x	63	7c	77	7b	F2	6b	6f	C5	30	1	67	2b	fe	D7	ab	76
1		Ca	82	C9	7d	Fa	59	47	F0	ad	D4	A2	af	9c	A4	72	C0
2		87	Fd	93	26	36	3f	F7	Cc	34	A5	E5	F1	71	D8	31	15
3		4	C7	23	C3	18	96	5	9a	7	12	80	E2	eb	27	B2	75
4		9	83	2c	1a	1b	6e	5a	A0	52	3b	D6	B3	29	E3	2f	84
5		53	D1	0	Ed	20	Fc	B1	5b	6a	Cb	8e	39	4a	4c	58	cf
6		D0	Ef	Aa	Fb	43	4d	33	85	45	F9	2	7f	50	3c	9f	A8
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	29	90	88	46	Ee	88	14	de	5e	0b	db
a		E0	32	3a	0a	49	6	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	8
c		8a	78	25	2e	1c	A6	B4	C6	E8	Dd	74	1f	4b	bd	8b	8a
d		70	3e	B5	66	48	3	F6	0e	61	35	57	ba	86	C1	1d	9e
e		E1	F8	98	11	69	D9	8e	94	9b	1e	87	ca	ce	55	28	df
f		sc	A1	89	0d	bf	E6	42	68	41	99	2d	0f	B0	54	bb	16

Fig 2.4 S-Box Look Up Table

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	x	52	9	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		8	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	0	8c	Bc	D3	0a	F7	E4	58	5	B8	B3	45	6
7		D0	2c	1e	8f	Ca	3f	0f	2	C1	Af	bd	3	1	13	8a	6b
8		3a	91	11	41	4f	67	Dc	ea	97	F2	cf	ce	F0	84	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	1a
b		Fc	56	3e	4b	C6	D2	79	20	9a	Db	C0	fe	78	cd	5a	F4
c		1f	0d	A8	33	88	7	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	4	e	ba	77	D6	26	E1	69	3c	63	55	21	0c	7d

Fig 2.5 Inverse S-Box Look Up Table

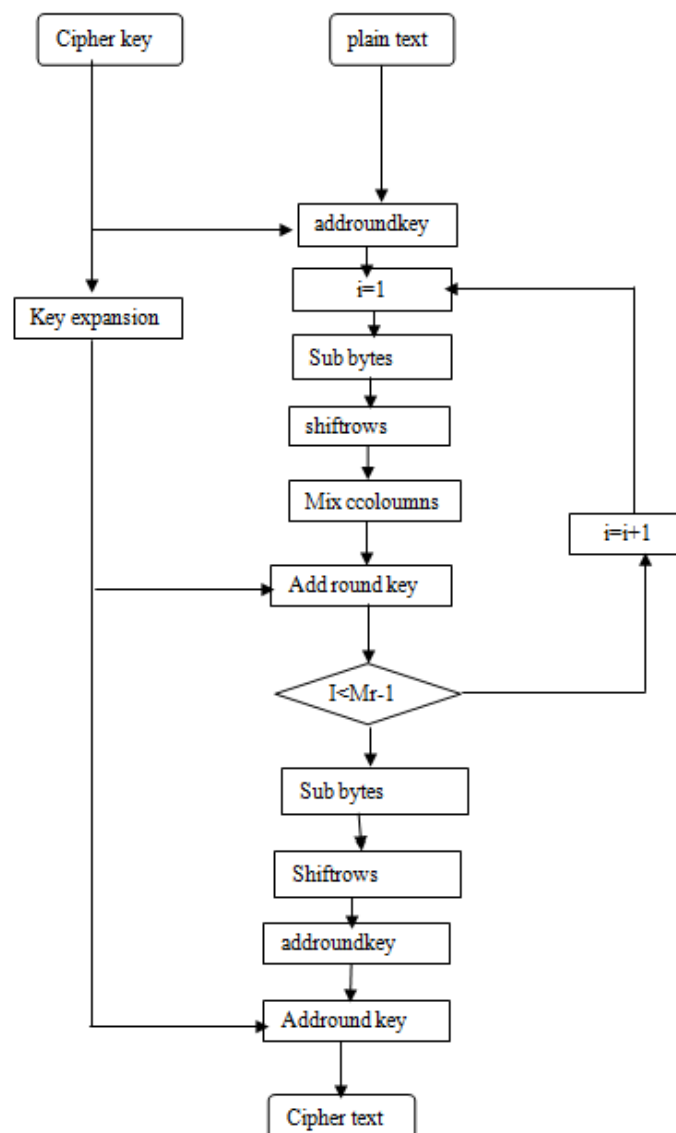


Fig 2.6 AES Flow Chart

III. SIMULATION RESULTS

Here we have designed AES using Verilog HDL , we have given byte of data as input to the subbyte transformation stage. We designed sub byte as a precalculated table is lookup table and is stored in ROM . From that look up table we will select the input data and replace it with the respective byte in such way every round was designed and synthesised using Xilinx ISE 13.2. The synthesis reports will giving us this type of AES design has less delay compared to the combinational design of s-box.



Fig 3.1 AES with Key generation

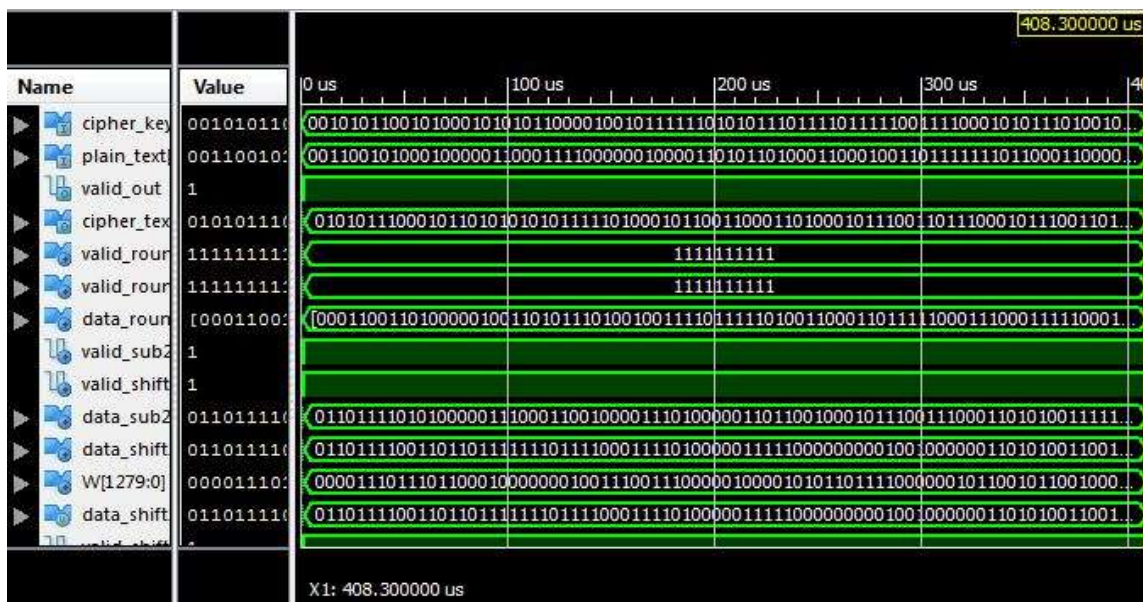


Fig 3.2 AES with output generation

IV.CONCLUSION



We have designed a symmetric block cipher with 128 bit data and key sizes which has encryption and decryption rounds may vary from 10, 12 or 14 rounds depend upon the key size. We have designed an AES algorithm which is having lesser delay and low complexity in hardware structure. This design enhances the

performance and throughput of the AES. This was designed using Verilog HDL and this design was synthesized in the Xilinx ISE13.2 and we have found that it is giving enhanced performance with lesser delay.

V. REFERENCES

- [1] An efficient FPGA implementation of the Advanced Encryption Standard algorithm Hoang Trang, NaNguyen Van Loi .
- [2] Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advanced Encryption Standard", Springer-Verlag, 2002
- [3] Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advancedm Encryption Standard", Springer-Verlag, 2002

AUTHOR DETAILS

	GADAAGOTTU JYOSTHNA pursuing M.tech (VLSI) from the Nalnda institute of Engineering and Technology(NIET), Siddhrath nagar,Kantepudi village, Satenpalli Mandal, Guntur dist, A.P, India. Her area of interest includes cryptographic application of VLSI
	L.SRINIVAS REDDY , He completed his post graduation in DECS. His area of interest includes digital electronics, digital communication, digital system design and VLSI technology and design. His research areas are optimal communication technology. He is currently working as Asst.professor (ECE) from Nalanda institute of Engineering and Technology (NIET), Siddharth Nagar,Kantepudi village, Satenepalli Mandal. Guntur Dist.,A.P,