Implementation of AES algorithm

Abhinandan Aggarwal[1], Gagandeep singh[2], Prof. (Dr.) Neelam Sharma[3]

Department of electronics and communication, Maharaja Agrasen Institute of Technology,New delhi ,India.

Abstract— *Data security has become one of the most important concerns in the recent times. This has led to an increase in the importance of cryptography of the electronic data. Cryptography is the process of protecting digital information. Though there are numerous encryption systems used in security systems by various organizations, for the wider use, a particular encryption method is used as a standard. The internationally accepted and acclaimed algorithm is Advanced Encryption Standard (AES). Here in this design we are implementing the Advanced Encryption Standard (AES) with a key length of 128 bits using Verilog hardware description language (HDL).*

Keywords— Advanced Encryption Standard (AES), cryptography , Cipher, Encryption, Field Programmable Gate Array (FPGA), Hardware description language(HDL), National Institute of Standards and Technology (NIST) , Verilog.

I. INTRODUCTION

Each day milllions of people generate enormous amounts of data in various fields such as banking , financial services , telecommunication etc. it is very important to not only keep this data secure during transmission but also during storage. In this regard cryptogrphy provides a method to be able to rely on the data and keep it secure from the attackers.

For a long time Data encryption standard(DES) was the standard for the symmetric key encryption.It has key length of 56 bits. This key length is small and could easily be attacked. The National Institute of Standards and Technology (NIST) thus called for a proposal for a new advanced encryption standard. Selection of AES was an open process. In 2001 NIST declared the block cipher Rijndael as the new AES.

AES takes an input data stream of 128 bits and encrypts it to give the output cipher of 128 bits. It supports 3 different key lengths and with each key different no. of rounds are associated. For a 128 bit key there are 10 rounds, with 192 bit key 12 rounds and with 256 bit key there are 14 rounds.

Wireless Local Area Networks (WLAN), Wireless Personal Area Networks (WPAN), Wireless Sensor Networks (WSN), Smart Cards ,the Wi-Fi encryption standard IEEE 802.11i, the secure shell network protocol SSH (Secure Shell), the Internet phone Skype are examples of a few technologies where AES is used.

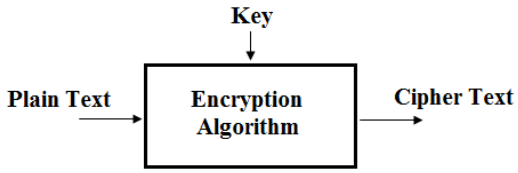

FIGURE 1 AES INPUT/OUTPUT PARAMETERS

II. RELATED WORK

AES was developed in a way so that it could be implemented on both software are hardware. Hardware implementation of AES can be done using the reprogrammable device like FPGAs(Field programmable gate arrays) as they can provide better performance than software methods.

The propsed design uses an efficeint way of implementing the 128 bit key AES encryption by reusing the resources required for encrypting the data in each round. Thus , reducing the resources required for the encryption of data. This helps in reducing number of slices reqiured in the FPGA and also helps in improving the performance. With a key length of 128 bits 10 rounds are required thus each round can be done in one clock cycle and thus a total of 10 clock cycles will be required to get the cipher text.

III. AES ALGORITHM

The complete flow of AES for a 128 bit key is as follows:

First the round key[0] is added to the plain text then the subsequent rounds are performed which use the same algorithm only in the last round there is no mix column layer except for these all other rounds function in the same way.

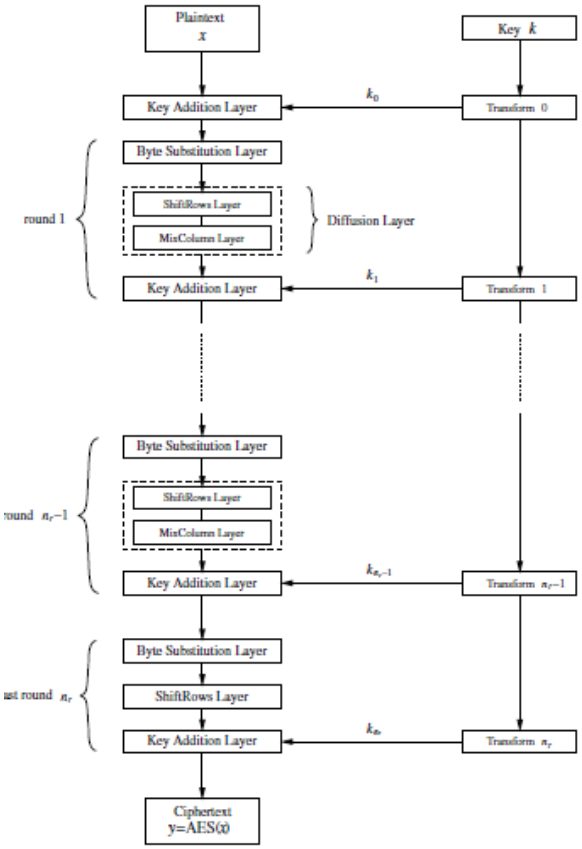

FIGURE 2 AES ENCRYPTION BLOCK DIAGRAM

Each round has the following layers:

4.1 Byte substitution layer:

A simple substitution of each byte using a look up table is done. Each byte of is replaced by byte indexed by row (left 4-bits) & column (right 4-bits) ex. byte {95} is replaced by byte in row 9 column 5 which has value {2a}

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

FIGURE 3 SUBSTITUTION BOX

4.2 Shift row layer : A circular byte shift in each row as :

a. 1st row is unchanged
b. 2nd row does 1 byte circular shift to left
c. 3rd row does 2 byte circular shift to left
d. 4th row does 3 byte circular shift to left

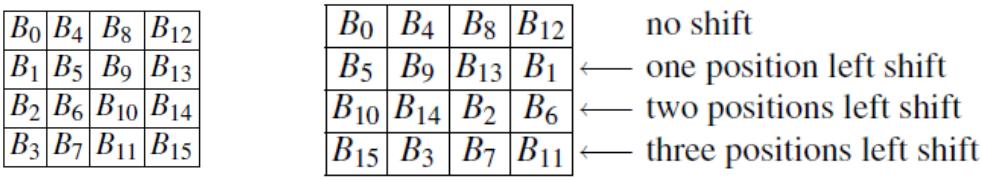

FIGURE 4 SHIFT ROW LAYER

4.3 Mix column layer

The *MixColumn* step is a linear transformation which mixes each column of the state matrix. Each byte is replaced by a value dependent on all 4 bytes in the column and is performed by the following multiplication. Multiplication and addition of the coefficients is done in $GF(2^8)$. Where multiplication by 2 is done by performing a left shift and xor with 1B if msb before shift is 1 and multiplication by 3 is multiplication with (01 xor 10).

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}.$$

FIGURE 5 MIX COLUMN LAYER

4.4 Add round key layer

XOR current state with 128-bits of the round key..

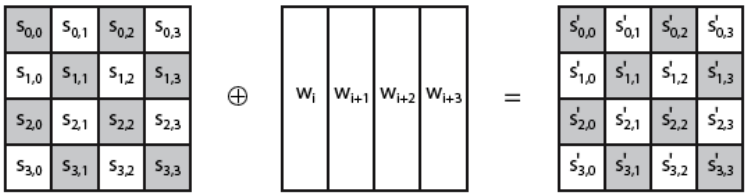

FIGURE 6 ADD ROUND KEY

Key transformation/schedule

The 128 bit key is divided into four 32 bit words. These words are then further processed to produce key in each round.

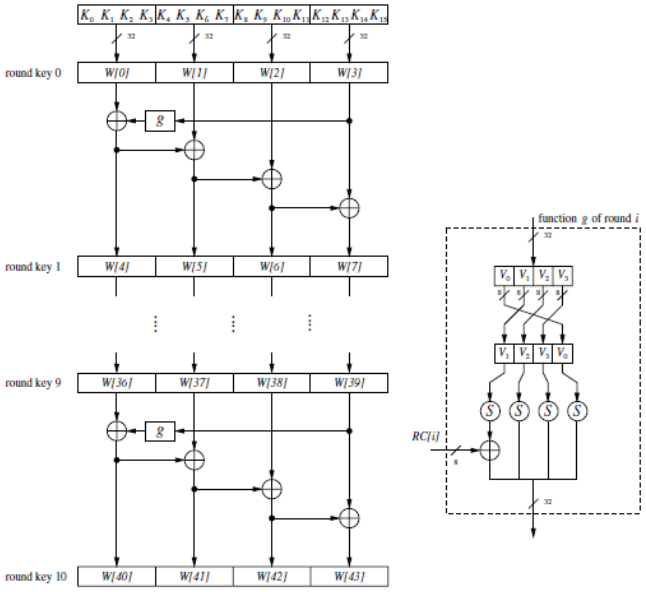

FIGURE 7 KEY TRANSFORMATION

Verilog HDL was used to implement this design.The simulation and verrification of the design was done on modelsim 6.4a . The synthesis of the design was done on Xilinx ISE.To minimze the resource requirment and the no. of slices needed the resources for each round were reused.

IV. RESULTS

The top module of the AES in Xilinx ISE 14.1 is as shown below

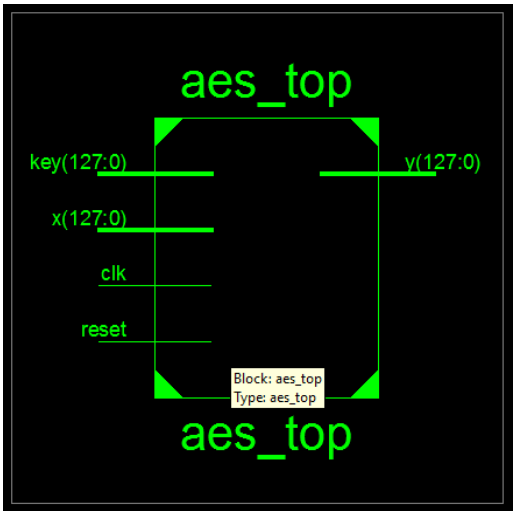

FIGURE 8 AES TOP BLOCK DIAGRAM

5.1 AES encryption simulation

This simulation is done in modelsim and shows the 128 bit output cipher text as output for 128 bit input. 'y' is the output as the cipher text of 128 bits in the simulation for input 'x' and for the key of 128 bits. Active low reset was used.

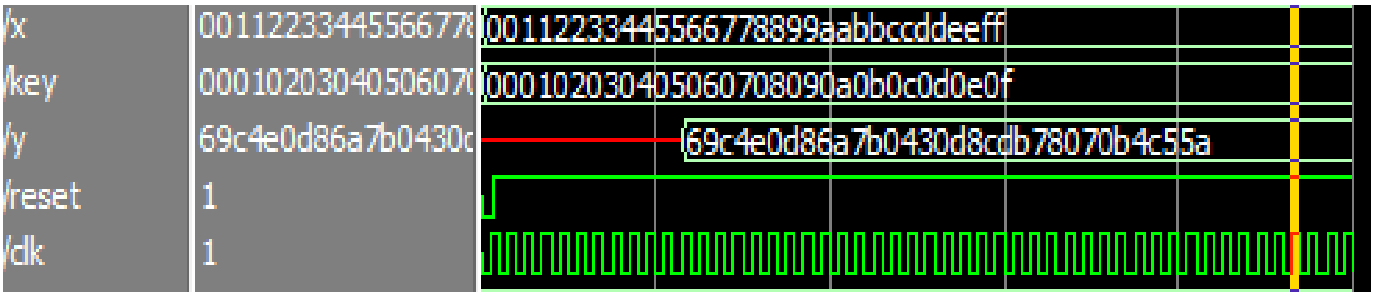

FIGURE 9 AES SIMULATION WINDOW

5.2 AES synthesis

Synthesis was done using Xilinx ISE 14.1 and Spartan 3E XCs500E – FG320 FPGA kit was used. Synthesis results were:

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	1464	4656	31%
Number of Slice Flip Flops	541	9312	5%
Number of 4 input LUTs	2764	9312	29%
Number of bonded IOBs	386	232	166%
Number of BRAMs	2	20	10%
Number of GCLKs	1	24	4%

FIGURE 10 AES SYNTHESIS RESULT

V. CONCLUSION

Although software implementations leads to smaller requirement of resources but high performance and speed can be achieved by hardware implementations. An efficient implementation of AES was done which resulted in lower no. of slices required for implementation. The efficiency and performance was made to increase. Thus reusability of resources can lead to better results. Simulation of AES algorithm was done on ModelSim software and implemented on Xilinx XC3S500E Spartan-3E FPGA kit.

REFERENCES

[1] Hoang Trang and Nguyen Van (2012), An efficient FPGA implementation of the Advanced Encryption Standard algorithm

[2] Xinmiao Zhang and Keshab K. Parhi,"Implementation approaches for the advanced encryption standard algorithm".

[3] Ritu Pahal Vikas kumar,"Efficient implementation of aes" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013 ISSN: 2277 128X.

[4] Understanding AES Mix-Columns Transformation Calculation , Kit Choy Xintong

[5] Yang Jun Ding Jun Li Na Guo Yixiong School of Information Science and Engineering, Yunnan University Kunming, China - "FPGA based design and implementation of reduced AES algorithm"(IEEE 2010).

[6] Deshpande, Hrushikesh S; Karande, Kailash J; Mulani, Altaaf O [IEEE 2014 International Conference on Communications and Signal Processing (ICCSP) efficient implementation of aes algorithm on FPGA.]

[7] Verilog HDL: A Guide to Digital Design and Synthesis, By Samir Palnitkar.

[8] Understanding Cryptography by Christof Paar, Jan Pelzl.