International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 7 41 - 44

Realization of Advanced Encryption Standard for Power and Area optimization

Mohini Mohurle
M.Tech , Electronics and communication Engineering
Priyadarshini College of Engineering
Nagpur, India
mohinimohurle@gmail.com

Prof. Vishal V. Panchbhai
Dept .of Electronics and Telecommunication Engineering
Priyadarshini College of Engineering
Nagpur, India
Vishal_panchbhai@rediffmail.com

***Abstract*—**An AES algorithm can be implemented in software or hardware but hardware implementation is more suitable for high speed applications.AES is most secure security algorithm to maintain safety and reliability of date transmission for this key size is important. And here used AES-256 bit.The main goal of paper is AES hardware implementation to achieve less area and low power consumptions also to achieve high speed data processing and reduce time for key generation. This paper presents AES-256 bit algorithm design consist of 128 bit symmetric key. Xilinx ISE.14.7(64-bit) is used for simulation by using VHDL and hardware implementation on FPGA(Xilinx Spartan 6 or Altera Cyclone 2 FPGA device).

Keywords- *Cryptography, Cipher, FPGA, Advanced Encryption Standard(AES),VHDL*

*****

I. INTRODUCTION

AES is a symmetric encryption algorithm processing data in block of 128 bits. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. Here AES 256 bit used for high efficiency. The only secret necessary to keep for security is the key.New AES algorithm with encryption and decryption was realized in Verilog Hardware Description Language. The 128-bit plaintext and 256-bit key, as well as the 128-bit output data were all divided into four 32-bit consecutive units respectively controlled by the clock. The pipelining technology was utilized in the intermediate nine round transformations so that the new algorithm achieved a balance between speed and chip area.

AES may configured to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. In this project AES-256 encryption and decryption with 256-bit key is considered. Hardware security solution based on highly optimized programmable FPGA provides the parallel processing capabilities and can achieve the required performance benchmarks. The current Area Optimized algorithm of AES are mainly based on the realization of S-box mode and the minimizing of the internal registers which could save the area of IP core significantly.

II. NETWORK SECURITY

Cryptographic algorithm are utilized for security services in various environments in which low cost and low power consumption are key requirements.The Encryption algorithm must provide facility of implementing security protocols, even for those low network speeds, increases the requirements for computational power. AES provides combination of security, performance and efficiency. For any security, here key size is important because of this it determines the strength of security, area optimization and power consumption[5].These parameters are important in case of designing lower area and less power consumption designs are used in real time applications[5].

III. AES ALGORITHM

The AES algorithm is a symmetric block cipher that processes data blocks of 128 bits using a cipher key of length 128, 192, or 256 bits. Each data block consists of a 4 × 4 array of bytes called the state, on which the basic operations of the AES algorithm are performed. The AES encryption and decryption procedure as shown in below figure.

After an initial round key addition, a round function consist of four different transformation subbyte(), shiftrows(), mixcolume() and addroundkey() is applied to the data block(i.e the state array). The round function is performed iteratively 10, 12, or 14 times, depending on the key length. Check that in the last round MixColumn() is not applied.

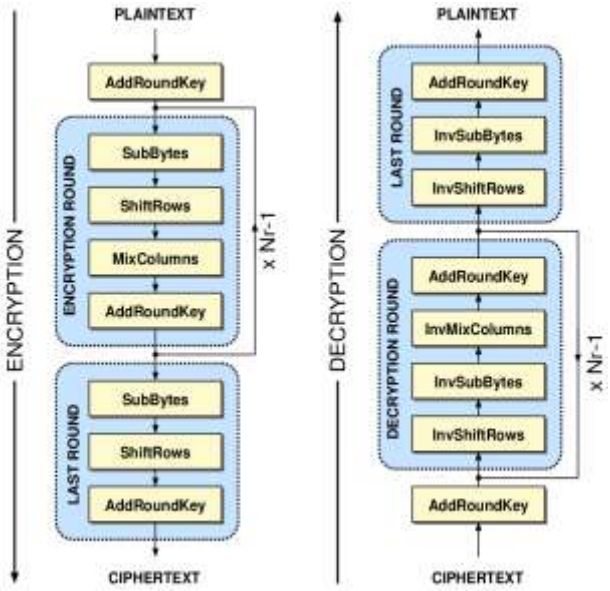

Fig1: AES Operational Flow

1. SubBytes( ) Transformation:

- Non-linear substitution table used in several byte substitution transformations and in the Key Expansion routine to perform a one-for-one substitution of a byte value.
- The S-box used in the Subbytes transformation is presented in hexadecimal form.

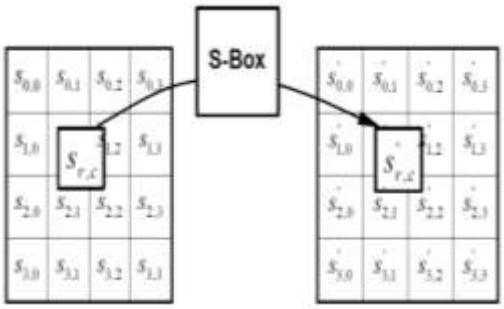

Fig 2:SubBytes( ) applies the s-box to each byte of the state.

- If S_{11}={53}, then the substitution value can be determined by the intersection of the row with index '5' and the column with index '3'. This result in S'_{11}having a value of {ed}.
- AES Encryption S-box:

x \ 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
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

Fig 3:AES encryption s-box

- AES Decryption S-box:

x \ 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

Fig 4:AES Inverse s-box of decryption

2.ShiftRows( ) Transformation:

- A transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

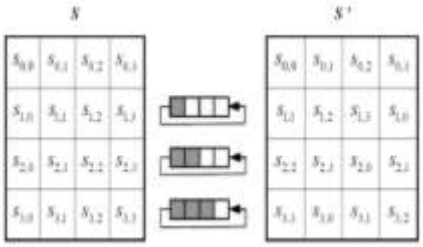

Fig 5:ShiftRows( ) cyclically shifts the last three rows in the state.

3.Mixcolumns( ) Transformation:

- The MixColumn transformation operates on the state column-by-column, treating each column as a four-term polynomial.

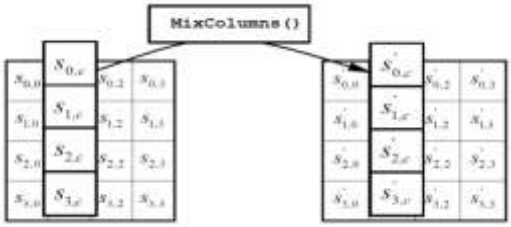

Fig 6: MixColumns( ) Operates on the state column by column

4.AddRoundKey( ) Transformation:

- The Add Round Key operation is a simple EXOR operation between the Data and the Round Key.
- The output of EXOR ing is again in the form of state matrix.

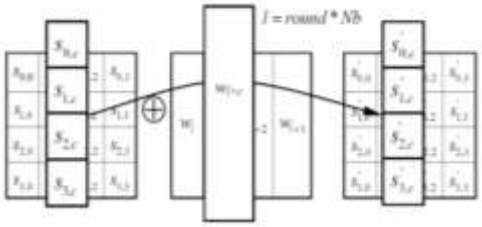

Fig 7:AddRoundKey( ) XORs each column of state with a word from the key schedule.

IV.AES ENCRYPTION PROCESS

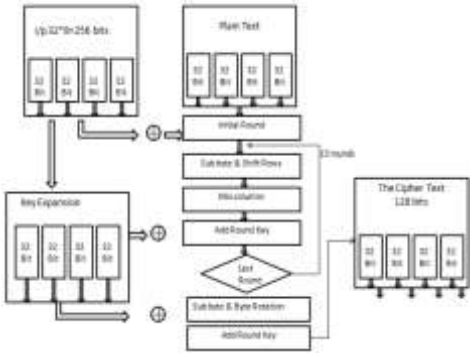

Fig 8: AES-256 Encryption Process

The four packets of consecutive 32-bit plaintext (128 bits)
have been put into the corresponding registers. While another four packets of consecutive 32-bit cipher key (256 bits)have been put into other registers.This module should combine the plaintext and cipher key by using the XOR operators.

After XOR operation round transformation mainly realizes function of SubBytes and MixColumn with 32 bit columns. Four packets of round transformation are processed separately.Then the results of MixColumns and the 32-bit keys from Keyexpansion are combined by using XOR operators. Here, the round transformation is a module with 64 input ports (32-bit plaintext+32-bit key) and 32 output ports formed.

The function of SubByte is realized by Look-Up Table(LUT).The SubByte matrix value replace by S-box table. It means that the operation is completed by the Find and Replace after all bits units are stored in a memory(256×8bit = 1024 bit).

The implementation of MixColumn is mainly based on the mathematical analysis in the Galois field GF(2^8). Only the multiplication module and the 32-bit XOR module of each processing unit(one column) are needed to design. Then the function of MixColumn can be achieved. Further processes will be proceeds up to 14 round and get encrypted output.

Encrypted output is the decryption of input. Decryption is the reverse process of encryption.

V.EXPERIMENTAL RESULTS

The AES simulation outputs is as follows by using Xilinx 14.1 ISE software

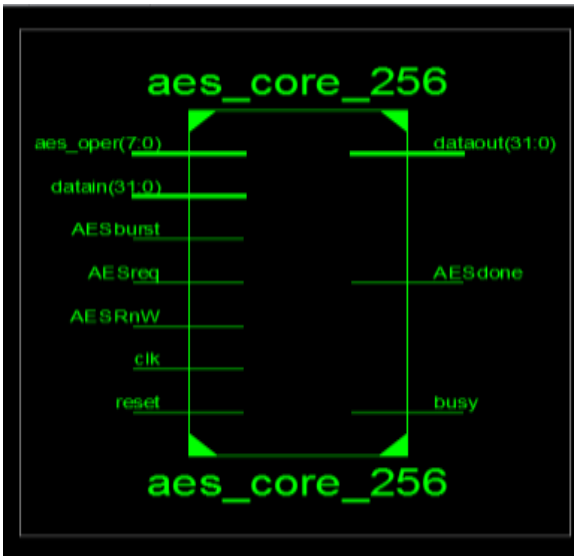

Fig 9:AES RTL Structure

This simulation result shows that RTL box design.RTL view shows all input and output ports. The ace_oper, datain acts as basic input for process along with that AESburst,AESreq to register or capture input. And AESRnW input pin is high for reading were as low for writing.

AES Encryption simulation:-

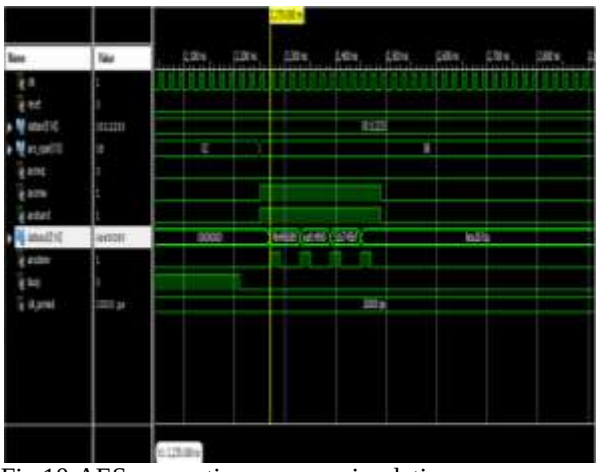

Fig 10:AES encryption process simulation

This is the encryption simulation in Xilinx ISE Project navigator in which Isim is used. The simulation consist of,Input: Data in=256 bit with 32 package,key=128 bit string,acereq=0,aesrnw= aesburst=1.
Output:aesdone=1,Data out=cipher text.

AES Decryption Simulation:

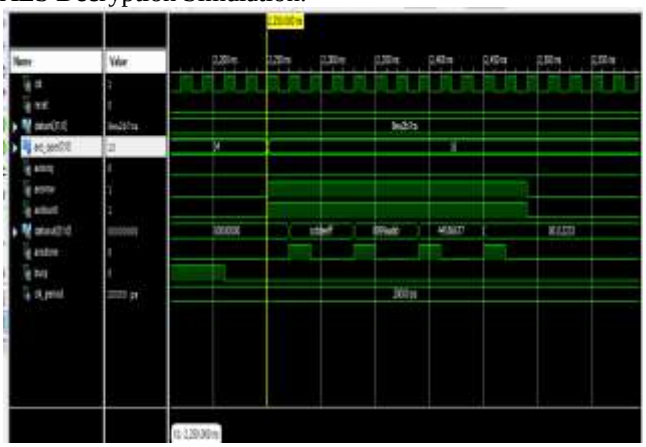

Fig 11:AES Decryption Process simulation

VI.CONCLUSION

AES-256 play an important role in network security.AES hardware high speed data processing using pipeline and parallel processing implementation. It shows 18% utilization of slice registers and 61% utilization of slice LUTs as compare to standard AES hardware implementation. The implementation can be carried out through several trade-offs between area and speed. All transformation of encryption can be simulated using iteration in order to minimize the hardware consumption.

VII.FUTURE SCOPE

Optimization in Area and Power provides better performance for high application used ASIC. The ASIC implementation of AES algorithm for 256 bit key length. In our area efficiency implement pipeling concept are include to get maximum through along with high speed of operation.

REFERENCES

[1] Pritamkumar N. Khose, Prof. Vrushali G. Raut, "Implementation of AES Algorithm on FPGA for Low Area Consumption",2015 International Conference on Pervasive Computing (ICPC).

[2] Wenfeng Zhao, Yajun Ha and Massimo Alioto,"Novel Self-Body-Biasing and Statistical Design for Near-Threshold Circuits With Ultra Energy-Efficient AES as Case Study",IEEE transactions on very large scale integration (VLSI) systems, vol. 23, no. 8, august 2015.

[3] Mostafa Taha, and Patrick Schaumont,"Key Updating for Leakagez Resiliency With Application to AES Modes of Operation",IEEE transactions on information forensics and security, vol. 10, no. 3, march 2015.

[4] Franck Courbon, Jacques J. A. Fournier,Philippe Loubet-Moundi, and Assia Tria, "Combining Image Processing and Laser Fault Injections for Characterizing a Hardware AES",IEEE transactions on computer-aided design of integrated circuits and systems, vol. 34, no. 6, June 2015.

[5] Hrushikesh.S.Deshpande,Kailas.J.Karande Altaaf.O.Mulani,"Efficient implementation of AES algorithm on FPGA",International Conference on Communication and Signal Processing, April 3-5, 2014, India .

[6] Kyungtae Kang, Member, IEEE, Junhee Ryu, and Dong Kun Noh, "Accommodating the Variable Timing of Software AES Decryption on Mobile Receiver",IEEE systems journal, vol. 8, no. 3, September 2014.

[7] Yuwen Zhu, Hongqi Zhang, Yibao Bao, "Study of the AES realization method on the reconfigurable hardware",2013 International conference on Computer Science and application.

[8] An Wang, Man Chen, Zongyue Wang, and Xiaoyun Wang, "Fault Rate Analysis: Breaking Masked AES Hardware Implementations Efficiently",IEEE transactions on circuits and systems—ii: express briefs, vol. 60, no. 8, august 2013.

[9] Dr.R.V.Kshirsagar, M.V.Vyawahare, "FPGA implementation of high speed VLSI Architecture for AES algorithm", 2012 fifth International Conferernce on emerging trends in engineering and technology.

[10] Bin Liu and Bevan M. Baas, "Parallel AES Encryption Engines for Many-Core Processor Arrays",IEEE transactions on computers, vol. 62, no. 3, march 2013.

[11] Chong Hee Kim"Improved Differential Fault Analysis on AES KeySchedule",IEEE transactions on information forensics and security",vol. 7, no. 1, February 2012.

[12] Mehran Mozaffari-Kermani and Arash Reyhani-Masoleh,"Efficient