Design and Implementation of Area Optimized 256 bit Advanced Encryption Standard on FPGA

Pallavi A. Vaidya
Student
Electronics Department
JDCOEM
Nagpur, India
e-mail: pallavi.vaidya10@gmail.com

Prof. Nilesh A.Mohota
Professor
Electronics Department
JDCOEM
Nagpur, India
email:nileshmohota@gmail.com

***Abstract*—** This paper presents architecture of the Advanced Encryption Standard (AES-Rijndael) cryptosystem. The reconfigurable architecture is capable of handling all possible combinations of standard bit lengths (128,192,256) of data and key. The two main parts of AES algorithm, namely encryption and key expansion, are considered for optimization. The major optimization criteria considered are maximization of hardware reduction and path delay reduction. The fully rolled inner-pipelined architecture ensures lesser hardware complexity. A new AES algorithm with 256-bit keys (AES-256) was described in this paper, which is to be realized in Verilog Hardware Description Language on FPGA board. In this novel work, substantial improvement in performance in terms of area, power and dynamic speed will obtained. This will give low complexity architecture and will easily achieve low latency as well as high throughput.

Keywords-*AES,DES,FPGA,Cryptography, pipeline, security, communications.*

*****

I. INTRODUCTION

Cryptographic algorithms are utilized for security services in various environments in which low cost and low power consumption are key requirements. Wireless Local Area Networks (WLAN), Wireless Personal Area Networks (WPAN), Wireless Sensor Networks (WSN), and Smart Cards are examples of such technologies[1]. For a long time, the Data Encryption Standard (DES) was considered as a standard for the symmetric key encryption. DES has a key length of 56 bits. However, this key length is now considered small and can easily be broken. For this reason, the National Institute of Standards and Technology (NIST) opened a formal call for algorithms in September 1997[2]. A group of fifteen AES candidate algorithms were announced in August 1998. In August 2000, NIST selected five algorithms: Mars, RC6, Rijndael, Serpent and Twofish as the final competitors. These algorithms were subject to further analysis prior to the selection of the best algorithm for the AES. Finally, on October 2, 2000, NIST announced that the Rijndael algorithm was the winner. Rijndael can be specified with key and block sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits. Therefore, the problem of breaking the key becomes more difficult[3]. The AES algorithm can be efficiently implemented by hardware and software. According to growing requirements for high speed, high volume secure communications combined with physical security, hardware implementation of cryptography takes place[2].

The Advanced Encryption Standard (AES) specifies cryptographic algorithm that can be used to protect electronic data. The AES algorithm is capable of using cryptographic keys of 128,192 and 256 bits to encrypt and decrypt data in blocks of 128,192 and 256 bits . With the rapid development and wide application of computer and communication networks, the information security has aroused high attention. Information security is applied to the political, military , diplomatic fields and common fields of people's daily lives. AES can resist various currently known attacks. Hardware security solution based on highly optimized programmable FPGA provides the required encryption performance.

II. DESCRIPTION OF ADVANCED ENCRYPTION STANDRAD ALGORITHM

The AES algorithm is a symmetric block cipher that can encrypt and decrypt the information. Encryption converts data to an unreadable form which is called as cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called as the plain-text [2].

III. AES ENCRYPTION

.

AES-Rijandael can support a variable block and key lengths of 128, 192 or 256 bits . The round transformations consists of following four different transformation : Bytesub, Shiftrow, Mixcolumn and Add round key. The first and last rounds differ from other rounds as there is an additional

ISSN: 2321 - 8169
772 - 775

AddRoundKey transformation at the beginning of the first round and no Mix Coulmns transformation is present in the last round.

A. SUB BYTES TRANSFORMATION:

The Sub Bytes transformation is a non-linear byte substitution method and operates on each of the state bytes independently. The Sub Bytes transformation is done using a pre calculated substitution table called as S-box. That S-box table is having 256 numbers (from 0 to 255) and their corresponding resulting values. In this design, we are using u a look-up table as shown in Table I. This is a more efficient method than directly implements the multiplicative inverse operation followed by affine transformation of the polynomial. This approach is used to avoid complexity of hardware implementation. This process has the advantage of performing the S-box computation in a single clock cycle and thus latency is reduced.

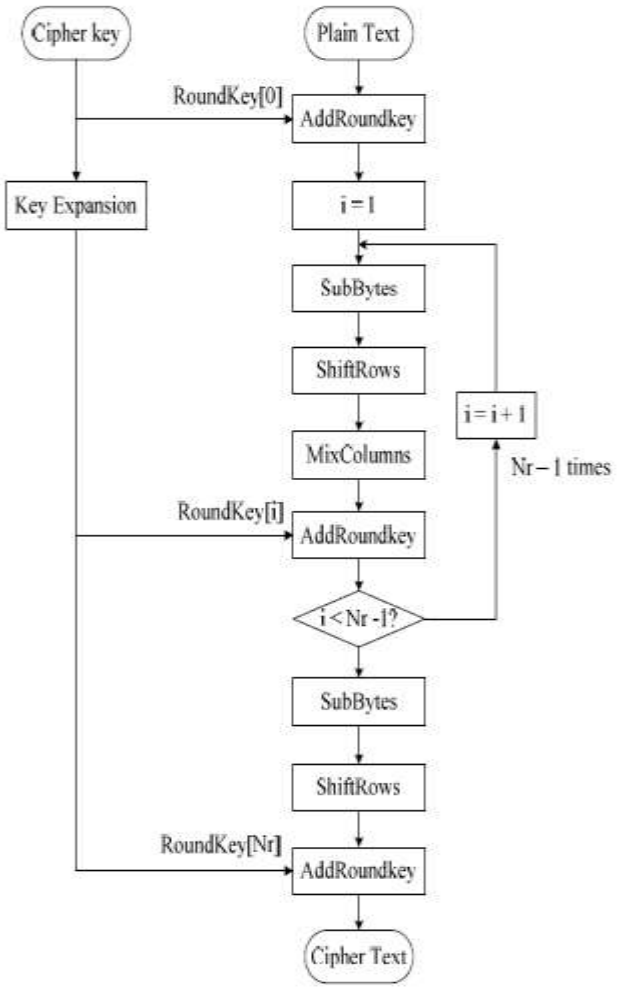

Figure 1. AES encryption structure

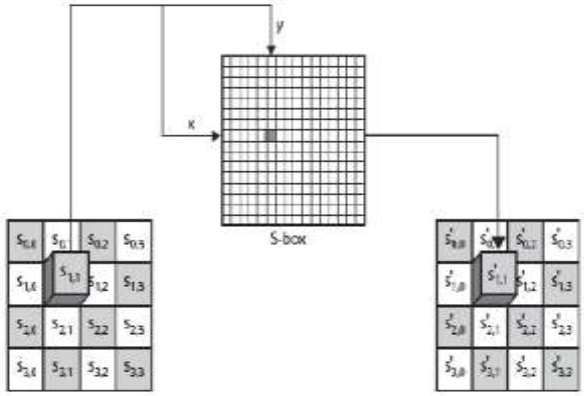

Figure 2.Sub bytes transformation process

TABLE I. S-BOX TABLE

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	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	b7	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	be	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	2a	90	88	46	ee	b8	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	ba	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	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

B. SHIFTROWS TRANSFORMATION:

In Shift Rows transformation method, the rows of the state matrix are cyclically left shifted over different offsets. Row 0 is not shifted by any value; row 1 is shifted by one byte to the left; row 2 is shifted by two bytes to the left and row 3 is shifted by three bytes to the left as shown in following figure.

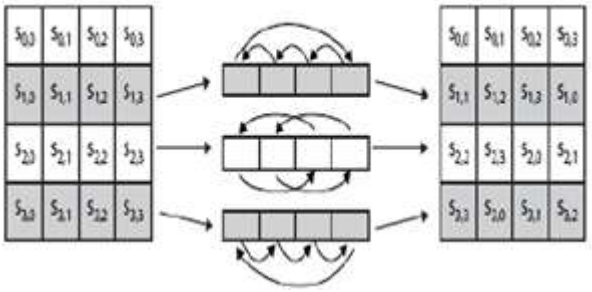

Figure 3.Shift row transformation process

 ISSN: 2321 - 8169
772 - 775

C. MIXCOLUMNS TRANSFORMATION:

In MixColumns transformation process, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial given by c(x),

c(x)={03}x3 + {01}x2 + {01}x + {02}.

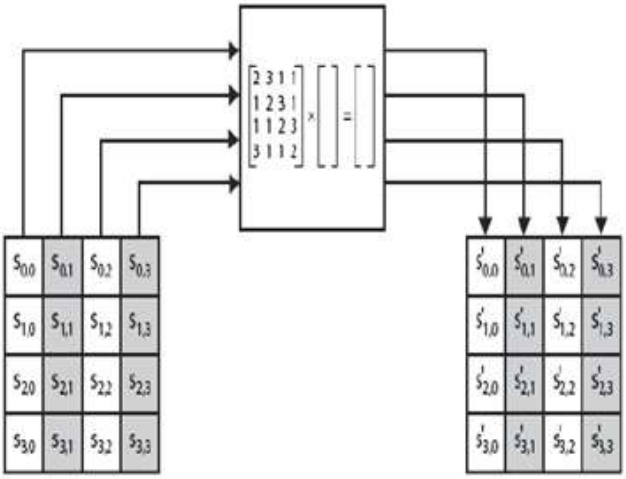

Figure 4.Mix columns transformation

D. Add Round key operation:

The XOR operation is performed between the state and the round key that it is generated from the main key by the Key Generation method. The matrix of keys is represented by w columns. Add Round Key is used both in the encryption and decryption algorithms. The XOR operation is conducted on byte basis, where the new output byte $s'_{x,y}$ is given by $s_{x,y}$ □□$k_{x,y}$.

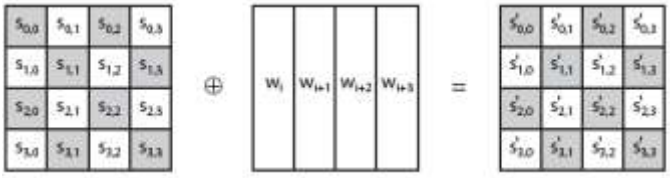

Figure 5.Add round key transformation process

IV . AES DECRYPTION

Decryption process is a reverse of encryption process which is inverse round transformations for find out the original plaintext of an encrypted cipher-text in reverse order. The round transformation of decryption uses the following four transformation:

Add Round Key, Inv Mix Columns, Inv Shift Rows, and Inv Sub Bytes .

E. ADD ROUND KEY OPERATION:

Add Round Key is its own inverse function as the XOR function is having its own inverse value. The round keys are selected in reverse order. The description of the other transformations is given below.

F. INV SHIFT ROWS TRANSFORMATION:

Inv Shift Rows exactly operates as Shift Rows, only in the opposite direction. The first row is not shifted by any value, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

G. INV SUB BYTES TRANSFORMATION:

The Inv Sub Bytes transformation is done using a once-pre calculated substitution table called as Inv S-box able. Inv S-box table is having 256 numbers (from 0 to 255) and their corresponding values. Inv S-box is presented in following Table II.

TABLE II INVS-BOX TABLE

	Y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	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	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	1a
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	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	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

X

V. PIPELINED ARCHITECTURE OF THE AES

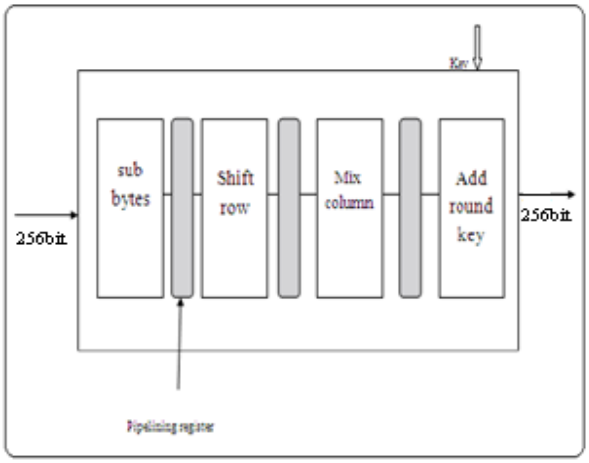

Figure 6. The basic structure of AES core

Figure.1 shows the basic building block of the AES core.The basic structure contains four separated blocks, Sub Bytes, Shift Rows, Mix Columns and Add Round Key. There is a 256-bit pipelining register in between each of these blocks. This full block is repeated ten times with the use of counter of value 10.Transformation of the plaintext for the cipher text, the value of rounds in AES algorithm whose packet length is 256 bits should be 10, 12, or 14 respectively, corresponding to the key length of 256 bits. In this paper, only the (AES-256) encryption methods with 256-bit keys is considered. In order to reduce the area and to enhance the processing of all 32-byte (one block) at a single AES, 256 bit pipelining registers are used in between all the blocks.

Two main processes of AES encryption algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. Key schedule consists of two modules: key expansion and round key election. Key expansion means mapping initial key to the so-called expanded key, while the round key selection selects round key from the expanded key module. Round Transformation involves following four transformations as Byte Substitution, Byte rotation, Mix Column and Add Round Key.

a) Byte Substitution operation simply substitutes the element of 256-bit input plaintext with the inverse element corresponding to the Galois field GF (2_8), whose smallest unit of operation is 8 bits/ group.

b) ByteRotation operation takes cyclic shift of the 256-bit state matrix, in which one row (64 bits) is taken as the smallest operand.

Blocks that are 256 bits long are shifted like this:

from

1 5 9 13 17 21 25 29
2 6 10 14 18 22 26 30
3 7 11 15 19 23 27 31
4 8 12 16 20 24 28 32

To

1 5 9 13 17 21 25 29
6 10 14 18 22 26 30 2
15 19 23 27 31 3 7 11
20 24 28 32 4 8 12 16

Note that in the 256 bit case, the rows are shifted 1, 3, and 4 places to the left, instead of 1, 2, and 3 places as for the other two block sizes.

c) Mix Columns operation takes multiplication and addition operations of the results of Byte Rotation with the corresponding irreducible polynomial x8 + x4 + x3 + x + 1 in GF(2_8), whose minimum operating unit is 32 bits.

d) Add round key operation takes a simple XOR operation with 8-bit units.

VI. CONCLUSION:

FPGA implementation of area optimized advanced encryption standard algorithm which meets the actual requirement is described in this paper. The new 256 bit AES algorithm is described in this paper. The design with pipelining technology and s-box implementation with the use of pre calculated table optimize the chip area effectively and meanwhile, design reduces power consumption also up to some extent ,for power consumption directly related to chip area.

REFERENCES

[1] Dr. R. V. Kshirsagar, M. V. Vyawahare:FPGA Implementation of High speed VLSI Architectures for AES Algorithm:2012 Fifth International Conference on Emerging Trends in Engineering and Technology.

[2] Hoang Trang , Nguyen Van Loi: An Efficient FPGA implementation of the Advanced Encryption Standard algorithm:2012IEEE.

[3] D. Mukhopadhyay, D. RoyChowdhury: An Efficient End to End Design of Rijndael Cryptosystem in 0. 18µ CMOS: The 18th International Conference on VLSI Design and The 4th International Conference on Embedded Systems (VLSID'05), pp. 405-410.

[4] V. Fischer and M. Drutarovasky Two Methods of Rijndael Implementation in reconfigurable Hardware: CHES 2001, LNCS Vol. 2162, pp. 77-92, Springer-Verlag.

[5] A. Rudra et al; Efficient Rijndael Encryption Implementation with Composite Field Arithmetic: CHES 2001, LNCS Vol. 2162, pp. 171- 184, Springer-Verlag 2001.

[6] A. Satoh, S. Morioka, K. Takona and S. Munetoh: A Compact Rijndael Hardware Architecture with S-Box optimization: Proceedings of Advances in Cryptography-ASIACRYPT 2001, LNCS Vol. 2248, pp. 239-254, Springer-Verlag.

[7] J. Wolkerstorfer, E. Oswald and M. Lamberger: An ASIC Implementation of the AES S-boxes: in Proc. RSA Conference (CT-RSA) 2002, LNCS Vol. 2271, pp. 67-78, San Jose, CA, Feb. 2002.

[8] C. Paar: Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields: PhD thesis, Institute of Experimental Mathematics, University of Essen, Germany, 1994.

[9] P. Chodowiec and K. Gaj: Very Compact FPGA Implementation of the AES Algorithm: CHES 2003, LNCS Vol. 2779, pp. 319-333, Springer-Verlag