Implementation of Rijindael's Encryption and Decryption Algorithm using FPGA

Vishakha.M.Gajbhiye[1], V.G Puranik[2]
[12]*Department of Electronics& Telecommunication Engineering, Bhivarabai Sawant Institute of Technology &Research, pune, India*

Abstract

Encryption is employed in communications systems to safeguard data being transmitted over a channel from being intercepted and skim by unauthorised parties. This protection is achieved by changing the initial message (plain text) into associate degree encoded kind (cipher text) that seems to be a random stream of symbol 2 architectures and VLSI implementations of the AES Proposal, Rijndael, are bestowed . These various architectures are operated each for encryption and decryption method. They cut back the specified hardware resources and come through high-speed performance. Their style philosophy is totally totally different. the primary uses feedback logic and reaches a output price adequate to 259 Mbit/sec. It performs expeditiously in applications with low lined space resources. The second design is optimized for high-speed performance exploitation pipelined technique. Its output will reach three.65 Gbit/sec. The ensuing VLSI circuits come through information rates considerably high, supporting each operation method (encryption/decryption) of Rijndael algorithmic program. they will be applied to on-line encryption/ decryption wants of high speed networking protocols like Asynchronous Transfer Mode (ATM) or Fiber Distributed information Interface (FDDI).

Keywords -- *Rijindael's algorithm, AES, DES, FPGA, matlab.*

I. INTRODUCTION

In this article, we have a tendency to square measure attending to concentrate on the Rijndael algorithmic program for the AES. This algorithmic program was submitted by two Belgians: Dr. Vincent Rijmen (pronounced Rye'-mun),a postdoctoral researcher within the EE Department (ESAT) of Katholieke Universiteit Leuven and Dr. JoanDaemen (pronounced Yo'-ahn Dah'- mun) of nucleon World International. A combination of things like security, performance, efficiency, ease of implementation and adaptability contributed to the choice of this algorithmic program because the AES. Specifically, Rijndael seems to perform systematically well in each hardware and software system platforms under a good vary of environments. These embrace economical VLSI and firmware implementations within the hardware and easy writing the code for the algorithmic program in numerous programming languages. This algorithmic program has wonderful key setup time and smart key gracefulness. But, a lot of significantly, while not sacrificing performance, it conjointly needs less memory for implementation. This fact makes it well matched for restricted-space environments. moreover, the structure of this algorithmic program seems to possess good potential for taking advantage of instruction-level correspondence. The AES is predicted to switch Triple-DES eventually due to its strong cryptologic options. The AES specifies 3 key sizes: 128, 192 and 256 bits. this suggests that, in decimal terms, there square measure about three.4 x 1038 doable 128-bit keys, 6.2 x 1057 possible 192-bit keys, and 1.1 x 1077possible 256-bit keys. compared, DES keys square measure 56-bits long. This bitlength means there square measure about 7.2 x 1016 doable DES keys.Thus, there square measure on the order of 1021times a lot of AES 128-bit keys than DES 56-bit keys. forward that one might build a machine that would recover a DES key in an exceedingly second (i.e. strive 2^55 keys per second), it\'d then take that machine about 149 thousand billion (149 trillion) years to crack a 128-bit AES key. to place that into perspective, the universe is believed to be less than twenty billion years recent. With AES supporting considerably larger key sizes than what DES supports, federal agency believes that this algorithmic program has the potential of remaining secure well on the far side consecutive few decades.

II. AES ENCRYPTION

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]

A variable block length of 128,192 and 256 bits is supported by Rijandael's AES. The following are four different round transformations: ByteSub, ShiftRow, MixColumn and AddRoundkey. The first and last rounds differ from other rounds as there is an additional 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 a look-up table as shown in Table I. Using S-box, each byte xy (in hexadecimal) in the matrix is substituted with another byte by looking for the entry in the x-row and the y-column of the table. 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.

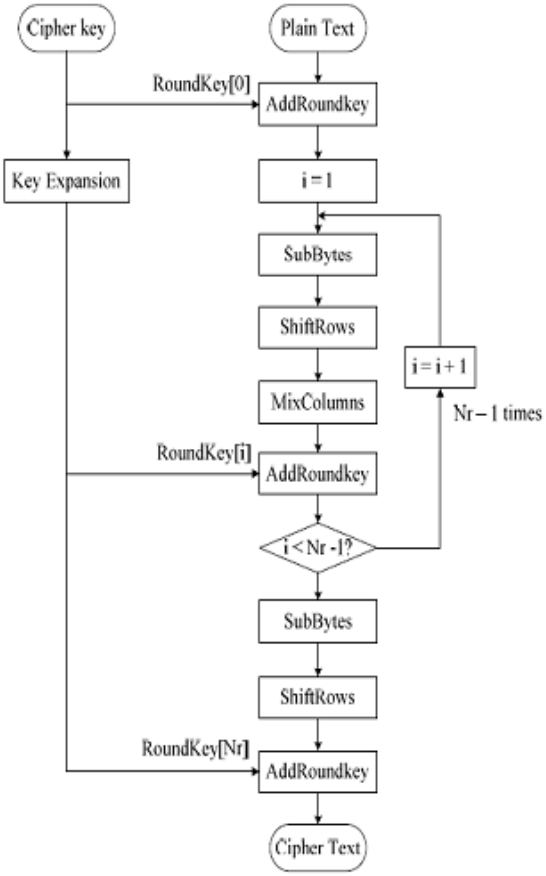


Figure 1. AES encryption structure

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.

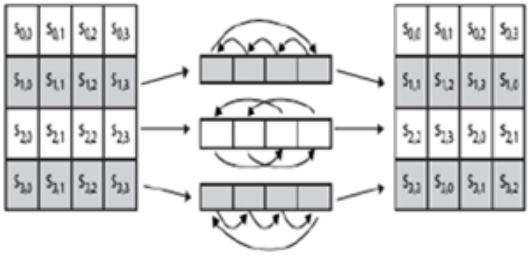


Figure 3. Shift row transformation process

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),

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

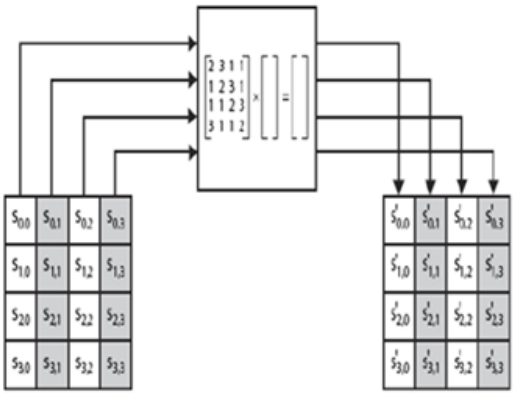


Figure 4. Mix columns transformation

D. *Addround 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 Sx,y □ □ Kx,y.

III. AES DECRYPTION

Decryption method is a reverse of encryption method that is inverse round transformations for determining the initial plaintext of an encrypted cipher-text in reverse order. The round transformation of decryption uses the subsequent four transformations:

Add round Key, Inv mix Columns, Inv Shift Rows, and Inv Sub Bytes.

A. *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 [5]. The description of the other transformations is given below.

B. *Inv Mixcolumn Transformation*

In Inv MixColumn 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 $c(x)^{-1}$,

$$c(x)^{-1} = \{0b\}x3 + \{0d\}x2 + \{09\}x + \{0e\}.$$

C. *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.

D. *INV SUB Bytes Transformation*

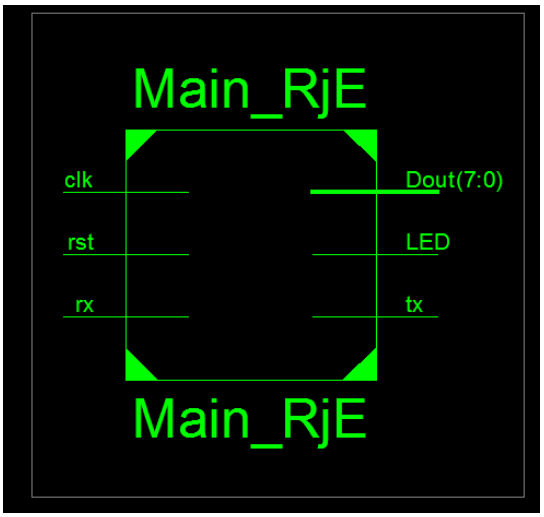
The Inv Sub Bytes transformation is done using a once-pre calculated substitution table called as Inv S-box able. Inv S-box table is having 256 numbers (from 0 to 255) and their corresponding values. Inv S-box is presented in following Table II.

TABLE II. INVS-BOX TABLE

X \ 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

IV. FIGURES AND TABLES

MAIN RTL



V. CONCLUSION

Thus with the help of matlab and FPGA Rijindael's encryption and decryption algorithm will be implemented. The performance of the system will be calculated by using performance counter. We can also increase the performance of the system by introducing the custom hardware. The combined design using hardware and software is known as Co-design. As the design can also reduces number of gate required by using Xilinx,

ACKNOWLEDGMENT

I would like to express my gratitude & sincere thanks to my guide Prof.v.g puranik , of Electronics and Telecommunication Engineering, for his constant motivation and support during the course of my work.

We would like to thank to Dr. Y.S.Angal, HOD, Electronics and Telecommunication

Engineering, Pune for his support and providing us the facilities.

RESULT

AES Text encryption and decryption Result using 128 bits key.

Input message: hi how are you.

Encrypted message: £5Y íï·-_p»ø «1h =|
Q»¶Þç□«[NíØ¦©U

Decrypted message: hi how are you.

REFERENCES

[1] Dr. Tariq Jamil, "The Rijindael Algorithm" Department of Electrical and Computer Engineering, Sultan Qaboos University (Oman). 0278-6648/04 2004 IEEE

[2] Shunwen Xiao, Yajun Chen, PengLuo, "The Optimized Design of Rijindael Algorithm Based on SOPC", College of Physics and Electronic information China West Normal University, Nanchong, China, 978-0-7695-3922-5/09 2009 IEEE

[3] Andre Luis PescoAlcalde, MarcioSilveiraOrtmann, Samir Ahmad Mussa, "NIOS II Processor Implemented in FPGA: An Application on Control of a PFC Converter" Federal University of Santa Catarina (UFSC), Department of Electrical Engineering (EEL), Power Electronics Institute (INEP), 978-1-4244-1668-4/08 2008, IEEE

[4] Meghana A. Hasamnis, Shri Ramdeobaba college of Engg and Management, S. S. Limaye, Jhulelal Institite og Technology, "Custome Hardware Interface using NIOS II Processor through GPIO", department of Electronics Engg., Nagpur, India, 978-1-4577-2119-9/12/ 2011 IEEE

[5] Madhav M. Deshpande, Meghana A. Hasamnis, "Design of Encryption System using NIOS II Processor", Electronics Department, R.C.O.E.M, Nagpur University, International Journal of Computer Applications, Volume 68- No. 21, April 2013

[6] N. Sklavos and O. Koufopavlou, Member IEEE, "Architectures and VLSI Implementations of the AES-Proposal Rijindael", Electrical and Computer Engineering Department, University of Patras, Greece, 0018-9340/02 2002 IEEE

[7] R. Sever, N. Ismailoglu, M. Askar, Y.C. Tekmen, "A High Speed ASIC Implementation of the Rijindael's Algorithm," 2004 IEEE International Symposium on Circuits and Systems, May 2004, Vancouver, Canada

[8] Refik Sever, A. NeslinIsmailoglu, Yusuf C. Tekmen, Murat Askar, BurakOkcan, "A High Speed FPGA Implementation of the Rijindael Algorithm", Ankara, Turkey, Proceedings of the EUROMICRO Systems on Digital System Design (DSD'04) 0-7695-2203-3/04 IEEE