ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 12, December 2015

Design and Implementation of Advanced Encryption Standard algorithm on Spartan 3E FPGA

S.Rizwan[1], Mr.V.Sai Kumar[2]
M.Tech Student[1], Assistant Professor[2]
Department of Electronics and Communication Engineering,
Madanapalle Institute of Technology and Science, Angallu, Madanapalle-517 325, A.P, India

Abstract **-- A projected Advanced Encryption Standard (AES) Algorithm is implemented on Spartan 3E FPGA is obtainable in this paper. This design uses input, output and key data which consist of sequence of 128 bits. The approach which is used by this design is iterative looping with block, key and replacement box (S-Box) for encryption and decryption. AES Encryption and decryption is programming using software Xilinx 14.7. By this design we calculate high throughput and Area. To achieve high security to data, less power consumption, low latency and gives low complexity architecture. It's applicable in Defense and Research Organizations for send secure data from third party.**

Keywords **-- *AES, encryption, decryption, FPGA, Throughput.***

I. INTRODUCTION

For the symmetric key encryption, Data Encryption Standard (DES) remained as a standard for long time. But the key length of DES is considered as small and easily broken. The key length of DES is 56 bits only. For this reason, In September 1997 the National Institute of Standards and Technology (NIST) opened an official call for algorithms. In August 1998 NIST announce group of fifteen AES applicant algorithms. In August 2000, five algorithms were selected as the final competitors: Rijndael, Serpent, RC6, Twofish and Mars. For the selection of the best algorithm, these algorithms were theme to further analysis. Finally, on October 2, 2000, Rijndael algorithm was announced as a winner. Maximum and Minimum of Key and block sizes are 256 bits and 128 bits respectively; it can be precise with key and block sizes in any multiple of 32 bits. Therefore, in AES breaking the key is very difficult. In cryptography, the AES is also known as Rijndael. It has fixed block length of 128 bits and key length of 128,192 and 256 bits respectively. But in this paper we are using key length of 128 bits only. The Rijndael algorithm is submitted by Joaen Daemen and Vincent Rijmen.

Rijndael algorithm was also designed to handle additional key and block sizes, However Additional features were not adopted by AES algorithm. The AES algorithm can be resourcefully implemented by software and hardware. Moreover, hardware implementation take place for growing requirements of high volume, high speed secure communications combined with physical security. Software implementations propose a limited physical security and expenditure the smallest resources, but they are slowest process.

For better solution than application specific integrated circuits (ASICs) and general purpose processors (GPPs) is an FPGA implementation. It provides wider applicability than ASICs and faster hardware solution over GPPs. Its configuring software makes use of the wide range of functionality supported by the reconfigurable device.

This paper deals with the design and implementation of Advanced Encryption Standard (AES) Algorithm which encrypts and decrypts block and key size of 128 bits and using iterate looping approach and lookup implementation of replacement box (S-Box). This design is to achieve high throughput, low latency and low power consumption.

II. LETERATURE SURVEY

AES Algorithm is a Substitution-Permutation Network. The Encryption means converting plain-text (encryption input data) into jumbled form called cipher-text and Decryption means converting cipher-text (Decryption input data (or) encrypted output data) into original form, which is a plain-text. The AES algorithm is a symmetric block cipher which can encrypts and decrypts the needed data. Using of iterative looping approach minimal number of clock cycles required to perform encryption and decryption for each data block of 128 bits.

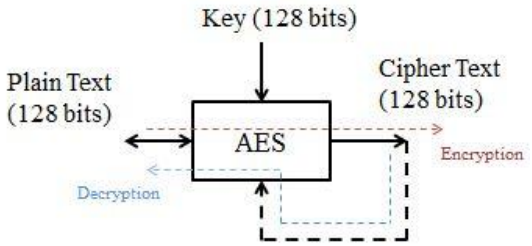

Fig.1: AES Encryption and Decryption

The AES Algorithm has two functions: AES Encryption and AES Decryption.

A. AES Encryption:

The AES Encryption operates on fixed block size of 128 bits and executed (Nr – 1) loop times and Nr means rounds which depends on the key lengths is 128,192 or 256 bits in length respectively. Nr can be 10, 12 or 14 depends upon key sizes. The first and last round of AES algorithm is differing from other rounds.

The additional AddRoundKey function is adding at the beginning of the first round and Mixcolumn function is not performed at last round. In this Paper we are using key and block size of 128 bits and the number of iteration of the loop are 10 for key length of 128 bits. The AES Encryption goes through four transformations.

1. SubByte Transformation:

It is a "Non-Linear Substitution" that independently on each byte of the state using an S-Box. Each state byte is replaced by another number in the S-Box (Replacement Box).the replacement follows a matrix where first hexadecimal values related to the "Row Positioning" and Second hexadecimal value corresponds to the "Column Positioning". In this design, we use an S-Box which is shown in Table I. S-Box is invertible and more effective than Multiplicative inverse followed by Affine Transformation.

Table 1: S Box Table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Complexity of hardware implementation can be avoided by this approach and can perform the S-Box Calculation in a single clock cycle, thus reducing the latency.

2. ShiftRow Transformation:

Each row of the state is cyclically shifted to the left depending on the row index. Last three rows of the state are shifted left and replacing their byte position.

3. MixColumn Transformation:

It operates on the state column-by-column and treating each column as a four term polynomial. Each Column is considered as word polynomials of GF (2^8) and multiplied by modulo x^4+1 with fixed polynomial {03} x^3+ {01} x^2+ {01} x+ {02}.

4. AddRoundKey Transformation:

Round key is added to the state which is obtain after MixColumns Transformation by a simple bitwise XOR operation. By using Key Expansion algorithm, the round key of each round key is derived from the main key. AES-128 needs 11 round keys. The first round key (RoundKey[0]) is the main key. The decryption and encryption needs 128 bit RoundKey which denoted by RoundKey[0] to RoundKey[10].

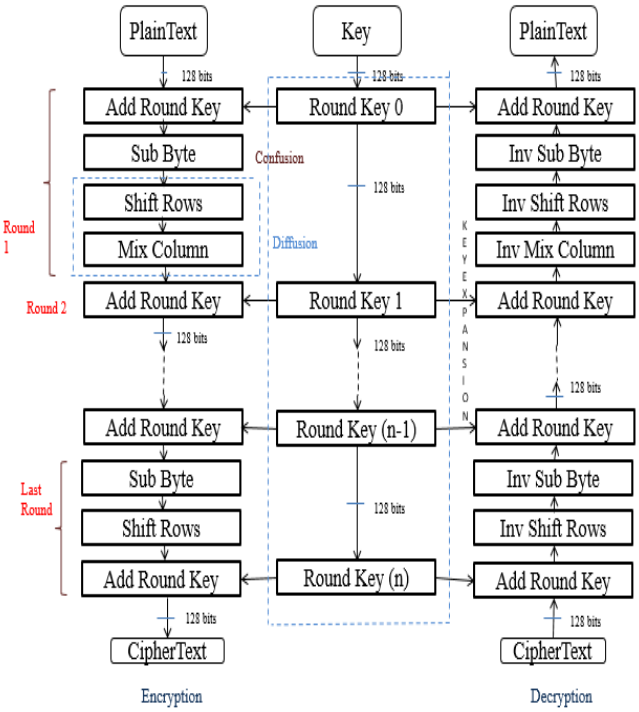

Fig.2: AES Algorithm Architecture

B. AES Decryption:

Decryption is an inverse whole encryption. Decryption computes out the original plain text of an encrypted cipher text in reverse order.

Apply all operations backwards as the key schedule stays the same. Only operations needs to implement are InvSubBytes, InvShift-Rows and InvMix-Columns while AddRoundKey stays the same.

1. AddRoundKey Transformation:

AddRoundKey is its own inverse function. Because XOR function is its own inverse. In Decryption Roundkey is taken in reverse order.

2. InvShiftRows Transformation:

It is same as ShiftRows only in opposite direction. The first row is not shifted, while 2nd, 3rd and 4th row is shifted to right by one, two and three respectively.

3. InvSubByte Transformation:

It is done using an Inverse S-Box. Inverse S-Box contains 256 numbers (0 to 255) and their corresponding values. Inverse S-Box is presented in Table II. Inverse S-Box uses inverse of Affine Transformation and takes multiplicative inverse in GF (2^8).

Table 2: Inverse S-Box Table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1x	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2x	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3x	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4x	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5x	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6x	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7x	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8x	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9x	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
ax	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
bx	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
cx	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
dx	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
ex	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
fx	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

4. InvMixColumns Transformation:

It operates on the state column-by-column and treating each column as a four term polynomial. Each Column is considered as word polynomials of GF (2^8) and multiplied by modulo x^4+1 with fixed polynomial {0B} x^3+ {0D} x^2+ {09} x+ {0E}.

C. ***Key Expansion****:*

AES algorithm takes the cipher key and performs key expansion routine to generate a key schedule. Key expansion generates a total of Nb(Nr+1) words. The algorithm requires an initial set of Nb words and each Nr rounds requires Nb words of key data. The expansion of the input key into the key schedule consists of 4 byte words proceeds according to pseudo code. Key Expansion function basically needs only two things: Input cipher Key and Output Expanded Key.

The key schedule is responsible for expanding a short key into a larger key. Each key size is expanded to different size 128,192 and 256 bits into 176,208 and 240 byte key. It is depends on Cipher key size.

Most of the operations of the key expansion can be implemented by 32 bits XOR's plus use of the S-Box and a cyclic byte rotation. The Key Management is another segment in AES Algorithm and it contains Key Distribution, key storage and backup, Key Disposal and Key change.

III. AES ALGORITHM IMPLEMENTED ON SPARTAN 3E KIT

Using Verilog algorithm designed AES Algorithm and it is implemented on a Spartan-3 XC3S1600E device FPGA using the ISE 14.7 design tool. Input for AES Algorithm is a 128 bit length plaint text, but AES algorithm is sort out when the inputs are in bytes (16 bytes). In AES, a sequence of 8 bits (Byte) treated as a single entity. The basic unit for processing in the AES algorithm is a byte.

In this design, input is given from keyboard and it is encoded in Hexadecimal. The experimental setup is shown in below figure. With the help of Visual Basic program Send plain text (input block) of data 4096 bytes and cipher key send to Spartan-3E XC3S1600E-FG320-5 device. VB is used as it is great at putting together a Graphical User Interface (GUI) quickly and it can create simple windows programs faster than we could do using just C.

FPGA kit is used for encryption and decryption of AES Algorithm and coding is done in Verilog language. AES architecture is implemented on FPGA of device family Spartan-3 using an efficient EDA tool Xilinx. Boundary Scan Mode of configuration is used to configure the XC3S1600E device. In Boundary Scan Mode of configuration the Spartan-3 FPGA is directly configured via a JTAG port using the pins TCK, TMS, TDI, and TDO. An on board JTAG connector is provided for configuring the FPGA through parallel port of PC via a parallel cable. A serial cable is used to connect the Spartan-3 kit and PC's serial port. The inputs will serially be sent to FPGA and cipher text from FPGA to PC.

IV. RESULT

The inputs of 4096 bytes data (ASCII Data) will be sent to Spartan 3E 1600E-FS320-5. But, it takes serial transformation of single entity (Byte) at a time and after all 16 byte transformed to FPGA then AES Encryption is done for 128 bits and again same processes for next 128 bits block data up to 4096 bytes and the whole data/text is implemented on Spartan 3E 1600Efs320-5. In this design the cipher key is programmed in Verilog and added to this design. So, its need not to given the cipher key. The TMFT Terminal is used for transforming data of 4096 bytes into FPGA by using UART, which interface between Computer

and Spartan 3E 1600E FPGA Kit. The input and output can be seen in TMFT Terminal software.

For decryption we use cipher text as input and use the same cipher key for decryption algorithm. The decryption is done on pc. When decryption is done, we get decryption output that is original data. Original data is obtained after ten rounds R10 of AES as shown.

Encryption:

INPUT: 637c777bf26b6fc53001672bfed7ab76

CIPHER KEY: 703eb5664803f60e63557b986c1d9e

OUTPUT: 51a3408f929d38f5bcb6da2110fff3d2

Decryption:

INPUT: 51a3408f929d38f5bcb6da2110fff3d2

CIPHER KEY: 703eb5664803f60e63557b986c1d9e

OUTPUT: 637c777bf26b6fc53001672bfed7ab76

Design and Implementation of AES based on Spartan 3E FPGA implementation is shown in detailed.

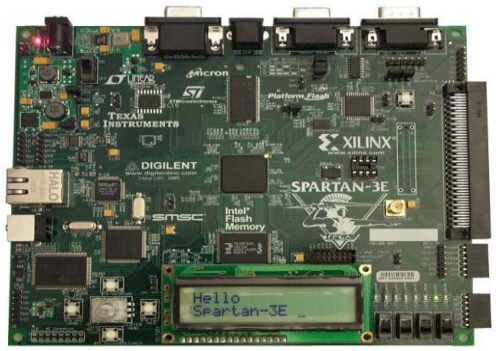

Fig.3: Spartan 3E 1600 Starter Kit

In one clock cycle, each round and Round Key is completed and the round keys is finished before the first round calculated in the design.

V. SYNTHESIS AND POWER DISTRIBUTION REPORT

The design has been coded by Verilog. The simulation and Synthesized processes based on Xilinx 14.7 and design is implemented on Spartan 3E 1600 FPGA.

Table 4: Utilization of Resources

Device Utilization Summery (Estimated Data)				
Utilization Logic		**Used**	**Available**	**Utilization**
Number of Slices	Encryption	947	14752	6%
	Decryption	1170		7%
Number of Slice Flip Flops	Encryption	829	29504	2%
	Decryption	628		2%
Number of 4 Input LUTs	Encryption	1589	29504	5%
	Decryption	2252		7%
Number of bonded IOBs	Encryption	6	250	2%
	Decryption	6		2%
Number of BRAMs	Encryption	16	36	44%
	Decryption	20		55%
Number of GCLKs	Encryption	2	24	8%
	Decryption	2		8%

AES Algorithm is implemented on Spartan 3E: Calculate Baud Rate, Memory, Throughput and Area. Over all Device Utilization of the FPGA is shown in table and timing report shows the required time taken by AES Algorithm to execute is shown in table 4.

Table 5: Timing Report

FACTORS		**VALUES**
Speed Grade	Encryption	-5
	Decryption	
Maximum Frequency	Encryption	154.89 MHz
	Decryption	128.7 MHz
Minimum Period	Encryption	6.456 ns
	Decryption	7.77 ns
Minimum input arrival time before clock	Encryption	4.726 ns
	Decryption	4.455 ns
Maximum output required time after clock	Encryption	5.640 ns
	Decryption	5.64 ns

For Encryption and Decryption, the baud rate is same and memory allocation is different and it has different throughputs.

Table 6: Simulation and Synthesis Report of this design

Designs		Our Design
FPGA Vendor		Xilinx
FPGA Chip		Spartan 3E 1600
Baud Rate		9600
Memory	Encryption	179720 Kbytes
	Decryption	183816 Kbytes
Throughput (Mbps)	Encryption	2269 Mbps
	Decryption	1132 Mbps

Table 7: Comparing with other Designs

Authors	**Device**		**Throughput (Mbps)**	**Slices**
Standaert et al. design	Vertex - 1000		1563	2257
Shuenn Shyang Wang et al.	Vertex-E BG860		1604	1857
Ours	XC3s1600E	Encrypt	2269	947
		Decrypt	1134	1170

The Comparisons of different authors, practical results must satisfy. To test the system, a test bench is used. The test bench applies encryption/decryption input pulse to activate the system. The simple victors provided for testing the design is FIPS 197. It achieves high throughput and low memory usage than other designs.

VI. CONCLUSION

The Advanced Encryption Standard algorithm uses data and cipher key length of 128 bits. This design is implemented on Spartan 3E 1600 E of AES Algorithm of key and data length of 128 bits using Xilinx 14.7 Software. It gives high performance architecture. The algorithm achieves a high throughput of 2296 Mbps for Encryption i.e., Encryption rate and 1134 Mbps for Decryption i.e., Decryption rate. Our architecture is better in terms of memory, throughputs as well as area. For further Defense Applications and Smart Card Applications we can use this algorithm for High Level Security purpose.

REFFERENCES

[1] Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advanced Encryption Standard", Springer-Verlag, 2002
[2] FIPS 197, “Advanced Encryption Standard (AES)”, November 26, 2001.
[3] Ahmad, N.; Hasan, R.; Jubadi, W.M; “Design of AES S-Box using combinational logic optimization”, IEEE Symposium on Industrial Electronics & Applications (ISIEA), pp. 696-699, 2010.
[4] Alex Panato, Marcelo Barcelos, Ricardo Reis, “An IP of an Advanced Encryption Standard for Altera Devices”, SBCCI 2002, pp. 197-202, Porto Alegre, Brazil, 9 and 14 September 2002.
[5] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare, “FPGA Implementation of AES Algorithm”, International Conference on Electronics Computer Technology (ICECT), pp. 401-405, 2011

AUTHORS DESCRIPTION

1. S.Rizwan is pursuing his M.Tech in the stream of Micro & Nano Electronics (M&NE) from Madanapalle Institute of Technology and Sciences, Madanapalle. He completed his B.Tech in the stream of Electronics and Communication Engineering, from BIT Institute of Technology, Hindupur. His areas of interest are Cryptography, VLSI, Embedded systems, IC fabrication, Micro & Nano Electronics.

2. Mr. V.Sai Kumar is currently working as Assistant Professor in department of ECE at Madanapalle Institute of Technology and Sciences, Madanapalle. He had done M.Tech in stream of VLSI and his areas of interest are VLSI, Micro and Nano Electronics, IC Fabrication, Cryptography. He published number of technical papers.