FPGA Based Hardware Implementation of AES Rijndael Algorithm for Encryption and Decryption

N. S. SAI SRINIVAS
Dept. of Electronics and Communication Engineering
Andhra University
Visakhapatnam, India
satya_srinivasnettimi@live.com

MD. AKRAMUDDIN
Project Engineer - I
Centre for Development of Advanced Computing (C-DAC)
Hyderabad, India
mdakramuddin@cdac.in

***Abstract*— AES algorithm or Rijndael algorithm is a network security algorithm which is most commonly used in all types of wired and wireless digital communication networks for secure transmission of data between two end users, especially over a public network. This paper presents the hardware implementation of AES Rijndael Encryption and Decryption Algorithm by using Xilinx Virtex-7 FPGA. The hardware design approach is entirely based on pre-calculated look-up tables (LUTs) which results in less complex architecture, thereby providing high throughput and low latency. There are basically three different formats in AES. They are AES-128, AES-192 and AES-256. The encryption and decryption blocks of all the three formats are efficiently designed by using Verilog-HDL and are synthesized on Virtex-7 XC7VX690T chip (Target Device) with the help of Xilinx ISE Design Suite-14.7 Tool. The synthesis tool was set to optimize speed, area and power. The power analysis is made by using Xilinx XPower Analyzer. Pre-calculated LUTs are used for the implementation of algorithmic functions, namely S-Box and Inverse S-Box transformations and also for GF (2^8) i.e. Galois Field Multiplications involved in Mix-Columns and Inverse Mix-Columns transformations. The proposed architecture is found to be having good efficiency in terms of latency, throughput, speed/delay, area and power.**

***Keywords*— Cryptography, Advanced Encryption Standard (AES), Encryption, Decryption, Rijndael, Hardware Description Language (HDL), Field Programmable Gate Array (FPGA)**

I. INTRODUCTION

In digital networks, data security is achieved by Cryptography. It involves various techniques for establishing a safe and secure communication link in presence of adversaries. Cryptographic algorithms aim to provide resistance against password attacks, spying and hacking. Many types of cryptographic algorithms are in existence. In 2001, The National Institute of Standards and Technology (NIST) has standardized AES Encryption and Decryption Algorithm which turned into Federal Information Processing Standard (FIPS-197). This algorithm was developed by two professional cryptographers Joan Daemen and Vincent Rijmen [1]-[12]. It finds applications in Mobile Phones, Smart Cards, Magnetism Cards, Intel Core Processors Family, Automated Teller Machines (ATM), WWW servers, SSD Devices, IPSec and SSL Protocols, various other transmission protocols standardized by IEEE, IEEE 802.11i WPA2 standard Wi-Fi networks for secure encryption and digital video systems, etc., ensuring safety, security and reliability of data transmission [8]-[12]. Implementation of AES algorithm can be done either in software or in hardware. But most of the practical real time applications prefer only the hardware implementation, since it is very fast, safe and highly reliable for high speed processing as compared to software implementation [1]-[11].

The structure of the paper is organized as follows. Section II briefly describes the architectures of individual blocks used in AES Encryption and Decryption respectively. All the LUTs which are used for implementing this algorithm on FPGA are presented in Section III. Section IV gives an outline of the procedure followed for implementing AES on FPGA. The simulation, synthesis and power estimation results of FPGA implementation are presented in Section V. The final conclusion is stated in Section VI.

II. AES ENCRYPTION AND DECRYPTION ALGORITHM

The AES algorithm is symmetric, block cipher and iterative type in nature. It is symmetric since it uses the same key for both encryption and decryption processes [2]. It is a block cipher because it processes individual data blocks having fixed length of 128 bits with a cipher key having variable key lengths chosen independently as 128, 192 or 256 bits [10]. Hence, this algorithm can be used with three different key lengths which results in three distinct formats referred to as AES-128, AES-192 and AES-256. It is iterative because the steps involved in this algorithm are repeated a number of times. These iterations are also called as rounds. The total number of iterations or rounds in encryption and decryption processes depends on the size of the key used. Table I illustrates the relationship between the key length and the total number of rounds. The 128-bit data block is grouped into 16 bytes and correspondingly mapped into an array of size 4 X 4 called as the State. All the internal operations are performed on the State [10].

TABLE I. RELATIONSHIP BETWEEN KEY LENGTH AND TOTAL NUMBER OF ROUNDS

AES	Key Length (*Nk*[a] *Words*)	Block Size (*Nb*[a] *Words*)	Number of Rounds (*Nr*)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

[a] Number of 32-bit words.

978-1-4673-9939-5/16/$31.00 ©2016 IEEE

Fig. 1 [2] shows the schematic block diagram of AES Encryption and Decryption blocks. Each of them incorporates four transformations (SubBytes, ShiftRows, MixColumns and AddRoundKey) in every round. But in the final round, the MixColumns transformation is ignored.

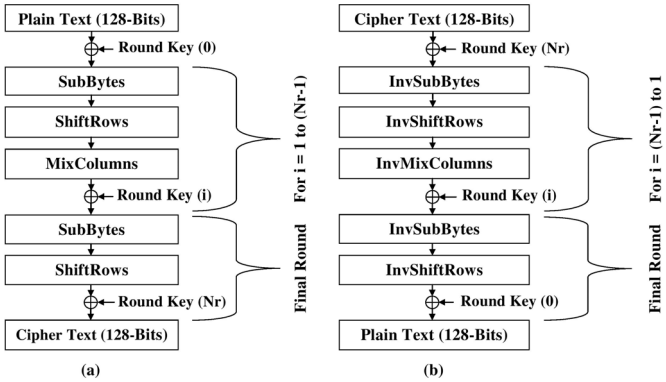

Fig. 1. AES Rijndael Algorithm. (a) Encryption Block. (b) Decryption Block

The SubBytes transformation is a non-linear byte substitution that operates independently on each byte of the State. It generally computes the multiplicative inverse of the bytes in GF (2^8) which is later followed by an affine transformation. Further details of SubBytes are given in [10]. ShiftRows is simply a cyclic left shift transformation with a constant offset which is equal to the row number (0, 1, 2 and 3) of the State. The MixColumns transformation is a linear transformation applied to the columns of the State, treating them as the coefficients of polynomial over GF (2^8) and is multiplied modulo ($x^4 + 1$) with a fixed polynomial a(x) given by (1) [10],

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \quad (1)$$

Further details of MixColumns are given in [10]. Finally, the AddRoundKey transformation is simply a bitwise XOR operation which is performed between the respective Round Key and the State.

The transformations in the Decryption process perform an inverse operation of the corresponding transformations present in the encryption block. The InvSubBytes is also a non-linear byte substitution, operating on the State. InvShiftRows is simply a cyclic right shift transformation with the same constant offset. The InvMixColumns multiplies the polynomial which is formed by each column of the State with $a^{-1}(x)$ modulo ($x^4 + 1$), where $a^{-1}(x)$ is given by (2) [10],

$$a^{-1}(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\} \quad (2)$$

Though the Decryption block can be derived by inverting the entire Encryption block, the sequence of transformations would be different from that of Encryption. Say for example, the InvShiftRows and InvSubBytes can be exchanged without affecting the Decryption process. This feature restricts the sharing of resources between Encryption and Decryption blocks [2].

In AES, the key expansion process generates the necessary round keys iteratively from the initial key. The generation process of round keys is unique for AES-128, AES-192 and AES-256. The process is clearly mentioned in [10]. Within the key expansion process, SubWord applies SubByte transformation to each of the four bytes in a word. The RotWord performs a cyclic left shift by one byte on each byte of the word. The Rcon is a round constant word array with a non-zero leftmost byte in each word [10].

III. LUTs USED FOR IMPLEMENTATION OF AES ON FPGA

The LUT based implementation of AES algorithm on FPGA is a traditional approach. It is very simple and easy to implement the desired functionality. When it is synthesized, it considerably occupies a less amount of area on FPGA [9]. So for this purpose, the following LUTs are used. Fig. 2 and Fig. 3 show LUTs for implementing SubBytes and InvSubBytes transformations respectively. Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8 and Fig. 9 show LUTs for obtaining the outcomes of GF (2^8) multiplications involved in (1) and (2) of MixColumns and InvMixColumns transformations respectively.

	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. 2. S-Box

	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. 3. Inverse S-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
1	20	22	24	26	28	2A	2C	2E	30	32	34	36	38	3A	3C	3E
2	40	42	44	46	48	4A	4C	4E	50	52	54	56	58	5A	5C	5E
3	60	62	64	66	68	6A	6C	6E	70	72	74	76	78	7A	7C	7E
4	80	82	84	86	88	8A	8C	8E	90	92	94	96	98	9A	9C	9E
5	A0	A2	A4	A6	A8	AA	AC	AE	B0	B2	B4	B6	B8	BA	BC	BE
6	C0	C2	C4	C6	C8	CA	CC	CE	D0	D2	D4	D6	D8	DA	DC	DE
7	E0	E2	E4	E6	E8	EA	EC	EE	F0	F2	F4	F6	F8	FA	FC	FE
8	1B	19	1F	1D	13	11	17	15	0B	09	0F	0D	03	01	07	05
9	3B	39	3F	3D	33	31	37	35	2B	29	2F	2D	23	21	27	25
A	5B	59	5F	5D	53	51	57	55	4B	49	4F	4D	43	41	47	45
B	7B	79	7F	7D	73	71	77	75	6B	69	6F	6D	63	61	67	65
C	9B	99	9F	9D	93	91	97	95	8B	89	8F	8D	83	81	87	85
D	BB	B9	BF	BD	B3	B1	B7	B5	AB	A9	AF	AD	A3	A1	A7	A5
E	DB	D9	DF	DD	D3	D1	D7	D5	CB	C9	CF	CD	C3	C1	C7	C5
F	FB	F9	FF	FD	F3	F1	F7	F5	EB	E9	EF	ED	E3	E1	E7	E5

Fig. 4. Galois Multiplication Lookup Table for Multiply by 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	03	06	05	0C	0F	0A	09	18	1B	1E	1D	14	17	12	11
1	30	33	36	35	3C	3F	3A	39	28	2B	2E	2D	24	27	22	21
2	60	63	66	65	6C	6F	6A	69	78	7B	7E	7D	74	77	72	71
3	50	53	56	55	5C	5F	5A	59	48	4B	4E	4D	44	47	42	41
4	C0	C3	C6	C5	CC	CF	CA	C9	D8	DB	DE	DD	D4	D7	D2	D1
5	F0	F3	F6	F5	FC	FF	FA	F9	E8	EB	EE	ED	E4	E7	E2	E1
6	A0	A3	A6	A5	AC	AF	AA	A9	B8	BB	BE	BD	B4	B7	B2	B1
7	90	93	96	95	9C	9F	9A	99	88	8B	8E	8D	84	87	82	81
8	9B	98	9D	9E	97	94	91	92	83	80	85	86	8F	8C	89	8A
9	AB	A8	AD	AE	A7	A4	A1	A2	B3	B0	B5	B6	BF	BC	B9	BA
A	FB	F8	FD	FE	F7	F4	F1	F2	E3	E0	E5	E6	EF	EC	E9	EA
B	CB	C8	CD	CE	C7	C4	C1	C2	D3	D0	D5	D6	DF	DC	D9	DA
C	5B	58	5D	5E	57	54	51	52	43	40	45	46	4F	4C	49	4A
D	6B	68	6D	6E	67	64	61	62	73	70	75	76	7F	7C	79	7A
E	3B	38	3D	3E	37	34	31	32	23	20	25	26	2F	2C	29	2A
F	0B	08	0D	0E	07	04	01	02	13	10	15	16	1F	1C	19	1A

Fig. 5. Galois Multiplication Lookup Table for Multiply by 3

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	09	12	1B	24	2D	36	3F	48	41	5A	53	6C	65	7E	77
1	90	99	82	8B	B4	BD	A6	AF	D8	D1	CA	C3	FC	F5	EE	E7
2	3B	32	29	20	1F	16	0D	04	73	7A	61	68	57	5E	45	4C
3	AB	A2	B9	B0	8F	86	9D	94	E3	EA	F1	F8	C7	CE	D5	DC
4	76	7F	64	6D	52	5B	40	49	3E	37	2C	25	1A	13	08	01
5	E6	EF	F4	FD	C2	CB	D0	D9	AE	A7	BC	B5	8A	83	98	91
6	4D	44	5F	56	69	60	7B	72	05	0C	17	1E	21	28	33	3A
7	DD	D4	CF	C6	F9	F0	EB	E2	95	9C	87	8E	B1	B8	A3	AA
8	EC	E5	FE	F7	C8	C1	DA	D3	A4	AD	B6	BF	80	89	92	9B
9	7C	75	6E	67	58	51	4A	43	34	3D	26	2F	10	19	02	0B
A	D7	DE	C5	CC	F3	FA	E1	E8	9F	96	8D	84	BB	B2	A9	A0
B	47	4E	55	5C	63	6A	71	78	0F	06	1D	14	2B	22	39	30
C	9A	93	88	81	BE	B7	AC	A5	D2	DB	C0	C9	F6	FF	E4	ED
D	0A	03	18	11	2E	27	3C	35	42	4B	50	59	66	6F	74	7D
E	A1	A8	B3	BA	85	8C	97	9E	E9	E0	FB	F2	CD	C4	DF	D6
F	31	38	23	2A	15	1C	07	0E	79	70	6B	62	5D	54	4F	46

Fig. 6. Galois Multiplication Lookup Table for Multiply by 9

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	0B	16	1D	2C	27	3A	31	58	53	4E	45	74	7F	62	69
1	B0	BB	A6	AD	9C	97	8A	81	E8	E3	FE	F5	C4	CF	D2	D9
2	7B	70	6D	66	57	5C	41	4A	23	28	35	3E	0F	04	19	12
3	CB	C0	DD	D6	E7	EC	F1	FA	93	98	85	8E	BF	B4	A9	A2
4	F6	FD	E0	EB	DA	D1	CC	C7	AE	A5	B8	B3	82	89	94	9F
5	46	4D	50	5B	6A	61	7C	77	1E	15	08	03	32	39	24	2F
6	8D	86	9B	90	A1	AA	B7	BC	D5	DE	C3	C8	F9	F2	EF	E4
7	3D	36	2B	20	11	1A	07	0C	65	6E	73	78	49	42	5F	54
8	F7	FC	E1	EA	DB	D0	CD	C6	AF	A4	B9	B2	83	88	95	9E
9	47	4C	51	5A	6B	60	7D	76	1F	14	09	02	33	38	25	2E
A	8C	87	9A	91	A0	AB	B6	BD	D4	DF	C2	C9	F8	F3	EE	E5
B	3C	37	2A	21	10	1B	06	0D	64	6F	72	79	48	43	5E	55
C	01	0A	17	1C	2D	26	3B	30	59	52	4F	44	75	7E	63	68
D	B1	BA	A7	AC	9D	96	8B	80	E9	E2	FF	F4	C5	CE	D3	D8
E	7A	71	6C	67	56	5D	40	4B	22	29	34	3F	0E	05	18	13
F	CA	C1	DC	D7	E6	ED	F0	FB	92	99	84	8F	BE	B5	A8	A3

Fig. 7. Galois Multiplication Lookup Table for Multiply by 11

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	0D	1A	17	34	39	2E	23	68	65	72	7F	5C	51	46	4B
1	D0	DD	CA	C7	E4	E9	FE	F3	B8	B5	A2	AF	8C	81	96	9B
2	BB	B6	A1	AC	8F	82	95	98	D3	DE	C9	C4	E7	EA	FD	F0
3	6B	66	71	7C	5F	52	45	48	03	0E	19	14	37	3A	2D	20
4	6D	60	77	7A	59	54	43	4E	05	08	1F	12	31	3C	2B	26
5	BD	B0	A7	AA	89	84	93	9E	D5	D8	CF	C2	E1	EC	FB	F6
6	D6	DB	CC	C1	E2	EF	F8	F5	BE	B3	A4	A9	8A	87	90	9D
7	06	0B	1C	11	32	3F	28	25	6E	63	74	79	5A	57	40	4D
8	DA	D7	C0	CD	EE	E3	F4	F9	B2	BF	A8	A5	86	8B	9C	91
9	0A	07	10	1D	3E	33	24	29	62	6F	78	75	56	5B	4C	41
A	61	6C	7B	76	55	58	4F	42	09	04	13	1E	3D	30	27	2A
B	B1	BC	AB	A6	85	88	9F	92	D9	D4	C3	CE	ED	E0	F7	FA
C	B7	BA	AD	A0	83	8E	99	94	DF	D2	C5	C8	EB	E6	F1	FC
D	67	6A	7D	70	53	5E	49	44	0F	02	15	18	3B	36	21	2C
E	0C	01	16	1B	38	35	22	2F	64	69	7E	73	50	5D	4A	47
F	DC	D1	C6	CB	E8	E5	F2	FF	B4	B9	AE	A3	80	8D	9A	97

Fig. 8. Galois Multiplication Lookup Table for Multiply by 13

These LUTs are implemented on FPGA in the form of Read Only Memories (ROMs). Each LUT ROM has address lines and data lines of size 8-bits [4].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	0E	1C	12	38	36	24	2A	70	7E	6C	62	48	46	54	5A
1	E0	EE	FC	F2	D8	D6	C4	CA	90	9E	8C	82	A8	A6	B4	BA
2	DB	D5	C7	C9	E3	ED	FF	F1	AB	A5	B7	B9	93	9D	8F	81
3	3B	35	27	29	03	0D	1F	11	4B	45	57	59	73	7D	6F	61
4	AD	A3	B1	BF	95	9B	89	87	DD	D3	C1	CF	E5	EB	F9	F7
5	4D	43	51	5F	75	7B	69	67	3D	33	21	2F	05	0B	19	17
6	76	78	6A	64	4E	40	52	5C	06	08	1A	14	3E	30	22	2C
7	96	98	8A	84	AE	A0	B2	BC	E6	E8	FA	F4	DE	D0	C2	CC
8	41	4F	5D	53	79	77	65	6B	31	3F	2D	23	09	07	15	1B
9	A1	AF	BD	B3	99	97	85	8B	D1	DF	CD	C3	E9	E7	F5	FB
A	9A	94	86	88	A2	AC	BE	B0	EA	E4	F6	F8	D2	DC	CE	C0
B	7A	74	66	68	42	4C	5E	50	0A	04	16	18	32	3C	2E	20
C	EC	E2	F0	FE	D4	DA	C8	C6	9C	92	80	8E	A4	AA	B8	B6
D	0C	02	10	1E	34	3A	28	26	7C	72	60	6E	44	4A	58	56
E	37	39	2B	25	0F	01	13	1D	47	49	5B	55	7F	71	63	6D
F	D7	D9	CB	C5	EF	E1	F3	FD	A7	A9	BB	B5	9F	91	83	8D

Fig. 9. Galois Multiplication Lookup Table for Multiply by 14

IV. AES ALGORITHM IMPLEMENTATION USING VERILOG-HDL

Structural level of abstraction is used for describing the AES algorithm in Verilog-HDL. All the individual blocks are coded separately and tested for their functionality. Finally, top modules representing the Encryption and Decryption blocks are designed by instantiating all the individual blocks. Then the design is simulated by using ISim simulator with sample inputs given in [10]. After verification of simulation results, the design is synthesized and implemented on Virtex-7 FPGA. The synthesis tool is set to optimize speed and power. Power estimations are then made by using Xilinx XPower Analyzer.

V. IMPLEMENTATION RESULTS

The simulation, synthesis and power estimation results obtained after FPGA implementation of AES algorithm as described in Section IV are as follows.

The obtained results are well organized in tabular forms starting from Table II to Table XXXVII.

Table II shows the specifications of FPGA [13] used for AES implementation. Table III shows the details of memory utilization by LUTs. Table IV, XIV and XXIV show the details of device utilization, Table V, XV and XXV show the details of slice logic distribution, Table VI, XVI and XXVI show the details of timing summary, Table VII, XVII and XXVII show the details of timing constraints for AES-128, AES-192 and AES-256 Encryption respectively. Similarly, Table IX, XIX and XXIX show the details of device utilization, Table X, XX and XXX show the details of slice logic distribution, Table XI, XXI and XXXI show the details of timing summary, Table XII, XXII and XXXII show the details of timing constraints for AES-128, AES-192 and AES-256 Decryption respectively.

Table VIII, XVIII and XXVIII show the results of AES-128, AES-192 and AES-256 Encryption respectively. Similarly, Table XIII, XXIII and XXXIII show the results of AES-128, AES-192 and AES-256 Decryption respectively.

Table XXXIV shows the details of power analysis for AES-128, AES-192 and AES-256 Encryption and Decryption respectively.

Table XXXV shows the details of latency and time taken for encrypting a single block of data by using AES-128, AES-192 and AES-256 Encryption respectively. Similarly

Table XXXVI shows the details of latency and time taken for decrypting a single block of data by using AES-128, AES-192 and AES-256 Decryption respectively.

Table XXXVII shows the details of throughput efficiencies and area constraint ratios of AES-128, AES-192 and AES-256 Encryption and Decryption respectively. Throughput and Throughput per Slice (TPS) are computed manually by using (3) and (4) respectively [3].

$$\text{Throughput} = \frac{\text{128 Bits x Clock Frequency}}{\text{Clock Cycles Per Encrypted or Decrypted Block}} \quad (3)$$

$$\text{TPS} = \frac{\text{Encryption or Decryption Rate}}{\text{CLB Slices Used}} \quad (4)$$

TABLE II. FPGA SPECIFICATIONS

Parameters	Values
Family	Virtex-7
Device	XC7VX690T
Package	FFG1761
Speed Grade	-1
System Clock Frequency (Differential)	200 MHz

TABLE III. ROM UTILIZATION BY LUTS

Parameters	Values
Memory Size of each LUT ROM	16 x 16 Bytes
Total Memory Size Occupied by all LUT ROMs	8 X (16 X 16) Bytes

TABLE IV. DEVICE UTILIZATION SUMMARY FOR AES-128 ENCRYPTION

Slice Logic Utilization			
Parameters	***Used***	***Available***	***Utilization***
No. of Slice Registers	3760	866400	0%
No. of Slice LUT's	10773	433200	2%
No. of Fully used LUT-FF Pairs	1465	13068	11%
IO Utilization			
No. of Bonded IOB's	385	850	45%
Specific Feature Utilization			
No. of BUFG/BUFGCTRL/BUFHCEs	1	272	0%

TABLE V. SLICE LOGIC DISTRIBUTION FOR AES-128 ENCRYPTION

Parameters	Values	
No. of LUT Flip Flop Pairs used	13068	
No. with an unused Flip Flop	9308 out of 13068	71%
No. with an unused LUT	2295 out of 13068	17%
No. of fully used LUT-FF Pairs	1465 out of 13068	11%
No. of unique control sets	11	

TABLE VI. TIMING SUMMARY FOR AES-128 ENCRYPTION

Parameters	Values
Minimum Period	4.806 ns
Maximum Frequency	208.073 MHz
Minimum Input arrival time before clock	4.623 ns
Maximum Output required time after clock	4.458 ns
Maximum Combinational Path Delay	4.288 ns

TABLE VII. TIMING CONSTRAINTS FOR AES-128 ENCRYPTION

	Parameter	Worst Case Slack	Best Case Achievable
Auto Time Spec Constraint for Clock Net CLK BUFGP	Setup		4.846 ns
	Hold	0.026 ns	

TABLE VIII. AES-128 ENCRYPTION RESULTS

I/O	Value
Input Data	3243F6A8885A308D313198A2E0370734
Cipher Key	2B7E151628AED2A6ABF7158809CF4F3C
Encrypted Data	3925841D02DC09FBDC118597196A0B32

TABLE IX. DEVICE UTILIZATION SUMMARY FOR AES-128 DECRYPTION

Slice Logic Utilization			
Parameters	***Used***	***Available***	***Utilization***
No. of Slice Registers	6088	866400	0%
No. of Slice LUT's	15240	433200	3%
No. of Fully used LUT-FF Pairs	4479	16849	26%
IO Utilization			
No. of Bonded IOB's	385	850	45%
Specific Feature Utilization			
No. of BUFG/BUFGCTRL/BUFHCEs	1	272	0%

TABLE X. SLICE LOGIC DISTRIBUTION FOR AES-128 DECRYPTION

Parameters	Values	
No. of LUT Flip Flop Pairs used	16849	
No. with an unused Flip Flop	10761 out of 16849	63%
No. with an unused LUT	1609 out of 16849	9%
No. of fully used LUT-FF Pairs	4479 out of 16849	26%
No. of unique control sets	11	

TABLE XI. TIMING SUMMARY FOR AES-128 DECRYPTION

Parameters	Values
Minimum Period	4.219 ns
Maximum Frequency	237.023 MHz
Minimum Input arrival time before clock	4.205 ns
Maximum Output required time after clock	1.147 ns
Maximum Combinational Path Delay	1.096 ns

TABLE XII. TIMING CONSTRAINTS FOR AES-128 DECRYPTION

	Parameter	Worst Case Slack	Best Case Achievable
Auto Time Spec Constraint for Clock Net CLK BUFGP	Setup		4.108 ns
	Hold	0.005 ns	

TABLE XIII. AES-128 DECRYPTION RESULTS

I/O	Value
Encrypted Input Data	3925841D02DC09FBDC118597196A0B32
Cipher Key	2B7E151628AED2A6ABF7158809CF4F3C
Decrypted Output Data	3243F6A8885A308D313198A2E0370734

TABLE XIV. DEVICE UTILIZATION SUMMARY FOR AES-192 ENCRYPTION

Slice Logic Utilization			
Parameters	***Used***	***Available***	***Utilization***
No. of Slice Registers	4435	866400	0%
No. of Slice LUT's	12227	433200	2%
No. of Fully used LUT-FF Pairs	1831	14831	12%
IO Utilization			
No. of Bonded IOB's	449	850	52%
Specific Feature Utilization			
No. of BUFG/BUFGCTRL/BUFHCEs	1	272	0%

TABLE XV. SLICE LOGIC DISTRIBUTION FOR AES-192 ENCRYPTION

Parameters	Values	
No. of LUT Flip Flop Pairs used	14831	
No. with an unused Flip Flop	10396 out of 14831	70%
No. with an unused LUT	2604 out of 14831	17%
No. of fully used LUT-FF Pairs	1831 out of 14831	12%

No. of unique control sets	13

TABLE XVI. TIMING SUMMARY FOR AES-192 ENCRYPTION

Parameters	Values
Minimum Period	3.839 ns
Maximum Frequency	260.516 MHz
Minimum Input arrival time before clock	4.177 ns
Maximum Output required time after clock	3.263 ns
Maximum Combinational Path Delay	3.151 ns

TABLE XVII. TIMING CONSTRAINTS FOR AES-192 ENCRYPTION

	Parameter	Worst Case Slack	Best Case Achievable
Auto Time Spec Constraint for Clock Net CLK BUFGP	Setup		4.264 ns
	Hold	0.003 ns	

TABLE XVIII. AES-192 ENCRYPTION RESULTS

I/O	Value
Input Data	00112233445566778899AABBCCDDEEFF
Cipher Key	000102030405060708090A0B 0C0D0E0F1011121314151617
Encrypted Output Data	DDA97CA4864CDFE06EAF70A0EC0D7191

TABLE XIX. DEVICE UTILIZATION SUMMARY FOR AES-192 DECRYPTION

Slice Logic Utilization			
Parameters	*Used*	*Available*	*Utilization*
No. of Slice Registers	7286	866400	0%
No. of Slice LUT's	17742	433200	4%
No. of Fully used LUT-FF Pairs	5468	19560	27%
IO Utilization			
No. of Bonded IOB's	449	850	52%
Specific Feature Utilization			
No. of BUFG/BUFGCTRL/BUFHCEs	1	272	0%

TABLE XX. SLICE LOGIC DISTRIBUTION FOR AES-192 DECRYPTION

Parameters	Values	
No. of LUT Flip Flop Pairs used	19560	
No. with an unused Flip Flop	12274 out of 19560	62%
No. with an unused LUT	1818 out of 19560	9%
No. of fully used LUT-FF Pairs	5468 out of 19560	27%
No. of unique control sets	13	

TABLE XXI. TIMING SUMMARY FOR AES-192 DECRYPTION

Parameters	Values
Minimum Period	4.292 ns
Maximum Frequency	232.992 MHz
Minimum Input arrival time before clock	4.278 ns
Maximum Output required time after clock	1.147 ns
Maximum Combinational Path Delay	1.092 ns

TABLE XXII. TIMING CONSTRAINTS FOR AES-192 DECRYPTION

	Parameter	Worst Case Slack	Best Case Achievable
Auto Time Spec Constraint for Clock Net CLK BUFGP	Setup		4.168 ns
	Hold	0.003 ns	

TABLE XXIII. AES-192 DECRYPTION RESULTS

I/O	Value
Encrypted Input Data	DDA97CA4864CDFE06EAF70A0EC0D7191
Cipher Key	000102030405060708090A0B 0C0D0E0F1011121314151617
Decrypted Output Data	00112233445566778899AABBCCDDEEFF

TABLE XXIV. DEVICE UTILIZATION SUMMARY FOR AES-256 ENCRYPTION

Slice Logic Utilization			
Parameters	*Used*	*Available*	*Utilization*
No. of Slice Registers	5356	866400	0%
No. of Slice LUT's	15376	433200	3%
No. of Fully used LUT-FF Pairs	2309	18423	12%
IO Utilization			
No. of Bonded IOB's	513	850	60%
Specific Feature Utilization			
No. of BUFG/BUFGCTRL/BUFHCEs	1	272	0%

TABLE XXV. SLICE LOGIC DISTRIBUTION FOR AES-256 ENCRYPTION

Parameters	Values	
No. of LUT Flip Flop Pairs used	18423	
No. with an unused Flip Flop	13067 out of 18423	70%
No. with an unused LUT	3047 out of 18423	16%
No. of fully used LUT-FF Pairs	2309 out of 18423	12%
No. of unique control sets	15	

TABLE XXVI. TIMING SUMMARY FOR AES-256 ENCRYPTION

Parameters	Values
Minimum Period	3.004 ns
Maximum Frequency	332.941 MHz
Minimum Input arrival time before clock	2.985 ns
Maximum Output required time after clock	2.473 ns
Maximum Combinational Path Delay	2.315 ns

TABLE XXVII. TIMING CONSTRAINTS FOR AES-256 ENCRYPTION

	Parameter	Worst Case Slack	Best Case Achievable
Auto Time Spec Constraint for Clock Net CLK BUFGP	Setup		4.210 ns
	Hold	0.036 ns	

TABLE XXVIII. AES-256 ENCRYPTION RESULTS

I/O	Value
Input Data	00112233445566778899AABBCCDDEEFF
Cipher Key	000102030405060708090A0B0C0D0E0F 101112131415161718191A1B1C1D1E1F
Encrypted Output Data	8EA2B7CA516745BFEAFC49904B496089

TABLE XXIX. DEVICE UTILIZATION SUMMARY FOR AES-256 DECRYPTION

Slice Logic Utilization			
Parameters	*Used*	*Available*	*Utilization*
No. of Slice Registers	8656	866400	0%
No. of Slice LUT's	20324	433200	4%
No. of Fully used LUT-FF Pairs	6458	22522	28%
IO Utilization			
No. of Bonded IOB's	513	850	60%
Specific Feature Utilization			
No. of BUFG/BUFGCTRL/BUFHCEs	1	272	0%

TABLE XXX. SLICE LOGIC DISTRIBUTION FOR AES-256 DECRYPTION

Parameters	Values	
No. of LUT Flip Flop Pairs used	22522	
No. with an unused Flip Flop	13866 out of 22522	61%
No. with an unused LUT	2198 out of 22522	9%
No. of fully used LUT-FF Pairs	6458 out of 22522	28%
No. of unique control sets	28	

TABLE XXXI. TIMING SUMMARY FOR AES-256 DECRYPTION

Parameters	Values
Minimum Period	3.205 ns
Maximum Frequency	312.012 MHz

Minimum Input arrival time before clock	3.222 ns
Maximum Output required time after clock	1.147 ns
Maximum Combinational Path Delay	1.101 ns

TABLE XXXII. TIMING CONSTRAINTS FOR AES-256 DECRYPTION

	Parameter	Worst Case Slack	Best Case Achievable
Auto Time Spec Constraint for Clock Net CLK BUFGP	Setup		3.881 ns
	Hold	0.129 ns	

TABLE XXXIII. AES-256 DECRYPTION RESULTS

I/O	Value
Encrypted Input Data	8EA2B7CA516745BFEAFC49904B496089
Cipher Key	000102030405060708090A0B0C0D0E0F 101112131415161718191A1B1C1D1E1F
Decrypted Output Data	00112233445566778899AABBCCDDEEFF

TABLE XXXIV. POWER ANALYSIS FOR AES-128, AES-192 & AES-256 ENCRYPTION AND DECRYPTION

Parameters	Values		
Temperature Grade	Commercial		
Ambient Temperature	25ºC		
Airflow LFM	250		
Heat Sink	Medium Profile		
Board Selection	Medium (10'' X 10'')		
# of Board Layers	12 to 15		
Supply Power	**Total**	**Dynamic**	**Quiescent**
	0.289 W	0	0.289 W
Thermal Properties	**C/W**	**Max. Ambient**	**Junction Temp**
Effective TJA	1.1	84.7ºC	25.3ºC

TABLE XXXV. ENCRYPTION LATENCY AND ENCRYPTION TIME FOR A SINGLE BLOCK OF DATA

Parameters	Latency (Clock Cycles)	Time (ns)
AES-128 Encryption	20	92.5
AES-192 Encryption	24	112.5
AES-256 Encryption	28	132.5

TABLE XXXVI. DECRYPTION LATENCY AND DECRYPTION TIME FOR A SINGLE BLOCK OF DATA

Parameters	Latency (Clock Cycles)	Time (ns)
AES-128 Decryption	30	142.5
AES-192 Decryption	32	152.5
AES-256 Decryption	41	197.5

TABLE XXXVII. THROUGHPUT, THROUGHPUT PER SLICE (TPS), AND AREA CONSTRAINT RATIO (ACR) FOR AES ENCRYPTION AND DECRYPTION

Parameter	Throughput	TPS	ACR
AES-128 Encryption	1.28 Gbps	118816	7%
AES-192 Encryption	1.07 Gbps	87511	8%
AES-256 Encryption	0.91 Gbps	59183	11%
AES-128 Decryption	0.85 Gbps	55774	8%
AES-192 Decryption	0.8 Gbps	45091	9%
AES-256 Decryption	0.62 Gbps	30506	11%

The simulation waveforms of AES Encryption and Decryption are shown in Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14 and Fig. 15. Out of all these, Fig. 10, Fig. 12 and Fig. 14 correspond to the Timing Diagrams of AES-128, AES-192 and AES-256 Encryption, while Fig. 11, Fig. 13 and Fig. 15 correspond to the Timing Diagrams of AES-128, AES-192 and AES-256 Decryption respectively.

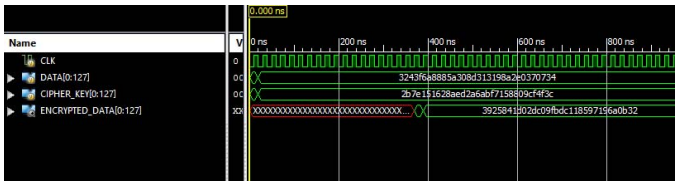

Fig. 10. AES-128 Encryption Timing Diagram

Fig. 11. AES-128 Decryption Timing Diagram

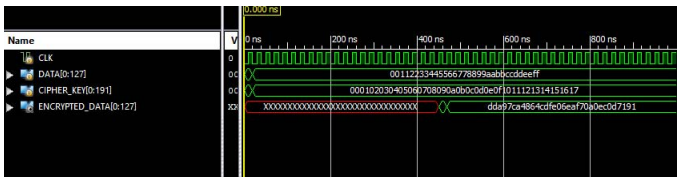

Fig. 12. AES-192 Encryption Timing Diagram

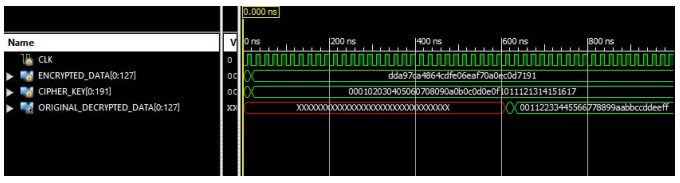

Fig. 13. AES-192 Decryption Timing Diagram

Fig. 14. AES-256 Encryption Timing Diagram

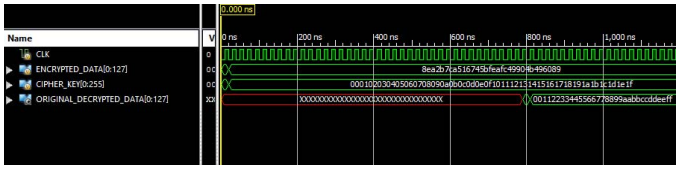

Fig. 15. AES-256 Decryption Timing Diagram

A bar graph illustrating the comparison of the area occupied on FPGA by AES-128, AES-192 and AES-256 Encryption and Decryption logic is shown in Fig. 16. From this figure, it is observed that AES-128 Encryption and Decryption logic occupies less area, while for AES-192 it is moderate and for AES-256 it is relatively high. This is evident

from the fact that, as the key size increases, the number of rounds in the encryption and decryption processes increases. This considerably leads to an increase in the area occupied on FPGA due to utilization of additional hardware resources.

The comparison of time taken for encrypting and decrypting a single block of data by using AES-128, AES-192 and AES-256 Encryption and Decryption logic is shown in Fig. 17. From this figure it is observed that AES-128 has less encryption and decryption time, while for AES-192 it is moderate and for AES-256 it is relatively high. This is evident from the fact that, as the key size increases, the number of rounds in the encryption and decryption processes increases. Therefore, this leads to a significant increase in the encryption and decryption time due to additional round computations. This reason is also valid for the case of latency.

The comparison of latency or clock cycles taken for encrypting and decrypting a single block of data by using AES-128, AES-192 and AES-256 Encryption and Decryption logic is shown in Fig. 18. From this figure, it is observed that AES-128 has less encryption and decryption latency, while for AES-192 it is moderate and for AES-256 it is relatively high.

The comparison of throughput achieved by AES-128, AES-192 and AES-256 Encryption and Decryption logic is shown in Fig. 19. From this figure, it is observed that AES-128 has high encryption and decryption throughput, while for AES-192 it is moderate and for AES-256 it is relatively less. From (3), we can say that throughput is a function of clock frequency and latency. The clock frequency is almost the same for both encryption and decryption processes. But latency is increasing with respect to an increase in the key size. As a result, due to this increase in latency, there is a corresponding decrease in the achieved throughput.

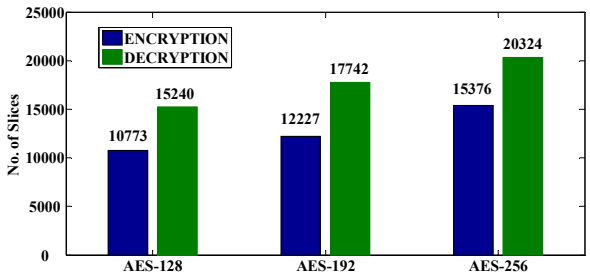

Fig. 16. Comparison of Area Occupied on FPGA by AES-128, AES-192 and AES-256 Encryption and Decryption Logics

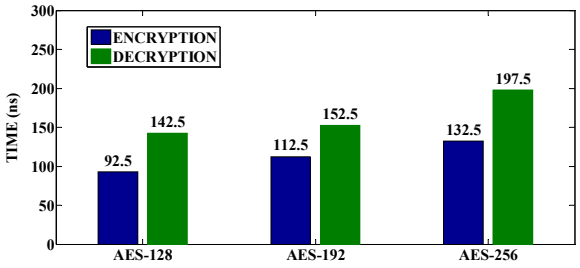

Fig. 17. Comparison of Time taken to Encrypt and Decrypt a Single Block of Data using AES-128, AES-192 and AES-256 respectively

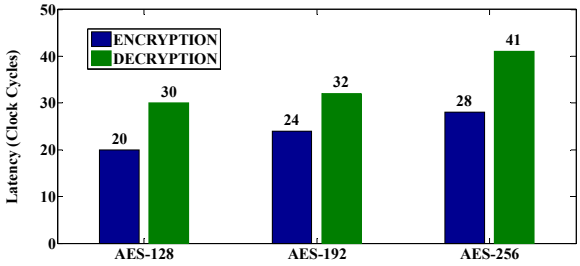

Fig. 18. Comparison of Latency (Clock Cycles) taken to Encrypt and Decrypt a Single Block of Data using AES-128, AES-192 and AES-256 respectively

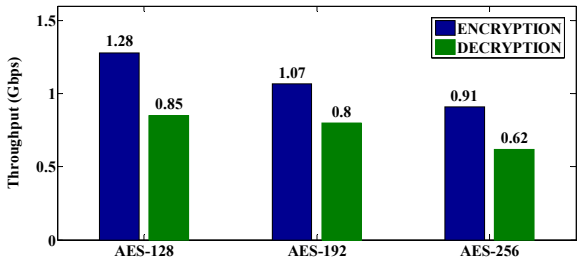

Fig. 19. Comparison of Throughput achieved by AES-128, AES-192 and AES-256 Encryption and Decryption Logics

VI. Conclusion

In this paper, FPGA based hardware implementation of AES Rijndael Algorithm is presented. LUTs are used for efficient implementation of various algorithmic functions. The proposed design is implemented on Xilinx Virtex-7 XC7VX690T FPGA. Virtex-7 is new to FPGA family and is based on 28 nm technology. It is well designed to meet the requirements of high performance. Its power efficiency helps to mitigate the power requirements of an increased design area.

The LUT based design approach gives less complex architecture and saves the processing time to a great extent by retrieving the necessary values from memory locations. Fetching values from memory locations is generally faster than executing complex computation operations. The overall proposed design is found to be having good efficiency in terms of various performance metrics like latency, throughput, speed/delay, area and power. The implemented design is having low latency, high throughput, high speed, low delay, low area occupancy and low power consumption. If high throughput is of major concern, then the synthesis tool has to be set to optimize speed. This will result in achieving better possible throughput with an associated cost being an increase in the area occupied on FPGA. Similarly, if the synthesis tool is set to optimize the area, then it will result in achieving less area occupancy on FPGA, but at the cost of reduced throughput. However, in this proposed design which is based on LUTs, it is observed that there is not much difference in the obtained speed and area constraints pertaining to both of these optimization techniques. So as a result, speed is chosen and given the topmost priority during synthesis. Therefore all the

results presented in this paper are with respect to speed and power optimization.

The design utilizes a very less supply power of 0.289 W at a junction temperature of 25.3ºC and occupies very less area roughly in the range of 1% - 4% when implemented on FPGA. The overall latency is less and typical values are in the range of 20-30 clock cycles for encryption and 30-40 clock cycles for decryption respectively. Throughput is significantly high. Achieved throughput is in the range of 0.90-1.28 Gbps for encryption and 0.6-0.85 Gbps for decryption respectively. Achieving throughput of this order is a challenging task. This high throughput is equivalent to the data transmission rates of several modern wired and wireless digital communication systems. This feature enables to incorporate the AES encryption and decryption hardware at the ends of transmitter and receiver respectively without affecting the data transmission rates of the communication system. The overall delay is very less and is reasonable. The delay lies within the acceptable limits and doesn't affect the functionality and timing constraints when this design is embedded with other complex designs. The operating frequency achieved is in the range of 200-300 MHz for both encryption and decryption logics.

In general, synthesis tools assume the worst possible operating conditions. So it is very common for actual design implementation to achieve much better performance results than those obtained from the synthesis reports. This LUT based design approach is the simplest of all the existing design approaches. This approach can be adopted for hardware designs which require a short time to market with no compromise in performance.

REFERENCES

[1] Tim Good, Student Member, IEEE, and Mohanned Benaissa, Member, IEEE, "Very Small FPGA Application-Specific Instruction Processor for AES," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, VOL. 53, ISSUE. 7, pp. 1477-1486, July 2006.

[2] Xinmiao Zhang, Student Member, IEEE, and Keshab K. Parhi, Fellow, IEEE, "High-Speed VLSI Architectures for the AES Algorithm," IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 12, ISSUE. 09, pp. 957-967, September 2004.

[3] Adam J. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 9, ISSUE. 4, pp. 545-557, August 2001.

[4] P. S. Abhijith, M. Srivastava, A. Mishra, M. Goswami and B. R. Singh, "High Performance Hardware Implementation of AES using Minimal Resources," IEEE International Conference on Intelligent Systems and Signal Processing, Gujarat, pp. 338-343, March 2013.

[5] Trang Hoang and Van Loi Nguyen, "An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm," IEEE International Conference on Computing and Communication Technologies, Research, Innovation and Vision for the Future, Ho Chi Minh City, pp. 1-4, February-March 2012.

[6] WANG Wei, CHEN Jie and XU Fei, "An Implementation of AES Algorithm Based on FPGA," IEEE 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Sichuan, pp. 1615-1617, May 2012.

[7] Meghana A. Hasamnis, S. S. Limaye, "Design and Implementation of Rijindael's Encryption Algorithm with Hardware/Software Co-design Using NIOS II Processor", 7th IEEE Conference on Industrial Electronics and Applications, Singapore, pp. 1386-1389, July 2012.

[8] A. M. Deshpande, M. S. Deshpande and D. N. Kayatanavar, "FPGA Implementation of AES Encryption and Decryption," IEEE International Conference on Control, Automation, Communication and Energy Conservation, Perundurai, Tamilnadu, pp. 1-6, June 2009.

[9] M. McLoone, and J. V. McCanny, "Rijndael FPGA Implementation Utilizing Look-Up Tables," IEEE Workshop on Signal Processing Systems, Antwerp, pp. 349-360, September 2001.

[10] AES (Advanced Encryption Standard), FIPS-197 (Federal Information Processing Standard), November 26, 2001, FIPS Publications. [Online] . Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[11] J. Daemen and V. Rijmen, "The Design of Rijndael," Springer-Verlag, 2002, ISBN: 978-3-662-04722-4. [Online]. Available: http://www.springer.com/in/book/978-3-540-42580-9

[12] William Stallings, "Cryptography and Network Security-Principles and Practice," Fifth Edition, Prentice Hall, Pearson, ISBN: 978-0-13-609704-4

[13] Xilinx Virtex-7 FPGA Data Sheets [Online]. Available: http://www.xilinx.com

N. S. Sai Srinivas is currently pursuing Integrated Dual Degree (B.E. + M.E.) in the Department of Electronics and Communication Engineering (ECE) at Andhra University College of Engineering Autonomous (AUCE-A), Andhra University, Visakhapatnam, India. His areas of interest include Digital Circuits and Systems, Communications etc. In 2015, he was awarded with the credential Mathworks Certified Matlab Associate.

MD. Akramuddin received his B.E. and M.E. Degrees in Electronics and Communication Engineering from Muffakham Jah College of Engineering and Technology, Osmania University, Telangana, Hyderabad, India in 2011 and 2013 respectively. He received Gold Medal and Merit Certificate in his Digital Systems specialization. Presently he is working as Project Engineer-I in Centre for Development of Advanced Computing (C-DAC), Hyderabad, A Scientific Society of the Ministry of Communication and Information Technology, Government of India. His area of research includes Digital FPGA Designs.