High Speed Area Efficient FPGA Implementation of AES Algorithm

P. B. Mane, A. O. Mulani
Department of Electronics and Telecommunication, AISSMS Inst. of Information Technology, Pune, India

Article Info

Article history:

Received Agu 6, 2018
Revised Oct 7, 2017
Accepted 0ct 21, 2018

Keywords:

DKPCA
FER
Gabor
KPCA

ABSTRACT

Now a day digital information is very easy to process, but it allows unauthorized users to access this information. To protect this information from unauthorized access, Advanced Encryption Standard (AES) is one of the most frequently used symmetric key cryptography algorithm. Main objective of this paper is to implement fast and secure AES algorithm on reconfigurable platform. In this paper, AES algorithm is designed with the aim to achieve less power consumption and high throughput. Keys are generated using MATLAB and remaining algorithm is designed using Xilinx SysGen, implemented on Nexys4 and simulated using Simulink. Synthesis result shows that it consumes 121 slice registers and its operating frequency is 1102.536 MHz. Throughput of the overall system is 14.1125 Gbps.

Corresponding Author:

A. O. Mulani,
Department of Electronics and Telecommunication,
AISSMS Institute of Information Technology, Pune, India.
Email: aksaltaaf@gmail.com

1. INTRODUCTION

NIST has started development process of FIPS for AES algorithm stating that this is the replacement for Data Encryption Standard algorithm. Alternatively, this algorithm is also known as Rijndael Algorithm. Rijndael algorithm has the advantages like resistance against all recognized attacks, code and speed compactness and simple design. Cryptography is a process in which the information to be sent is added with secret key so as to transmit the data securely at the destination. There are two types of cryptography based on type of key applied: Symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptography, equal key is utilized for encryption as well as decryption whereas in asymmetric key cryptography, different keys are required in encryption and decryption. AES algorithm is selected for implementation because it is secure, its components and design principles are completely specified. AES is a symmetric key block cipher. Design of AES algorithm is based on linear transformation Due to the use of Rijndael algorithm, different block and key sizes can be selected which was not possible in DES algorithm. Block and key size can be selected from 128/160/192/224/256 bits and need not be the same. According to AES standard, this algorithm can only accept 128 bits of block and key size can be selected from 128/192/256 bits. Based on the key size, number of rounds will vary. For example, if key size is 128, 192 or 256, then number of rounds will be 10, 12 and 14 respectively. Structure of AES algorithm is as shown in Figure 1. In this paper, this algorithm is designed with 128 bits of block size and key size respectively i.e. AES generates cipher text of 128 bits for 128 bits of plaintext. After the initial round, plaintext process through 10 rounds. Each round contains processes like byte substitution, shift rows, mix columns and add round key.

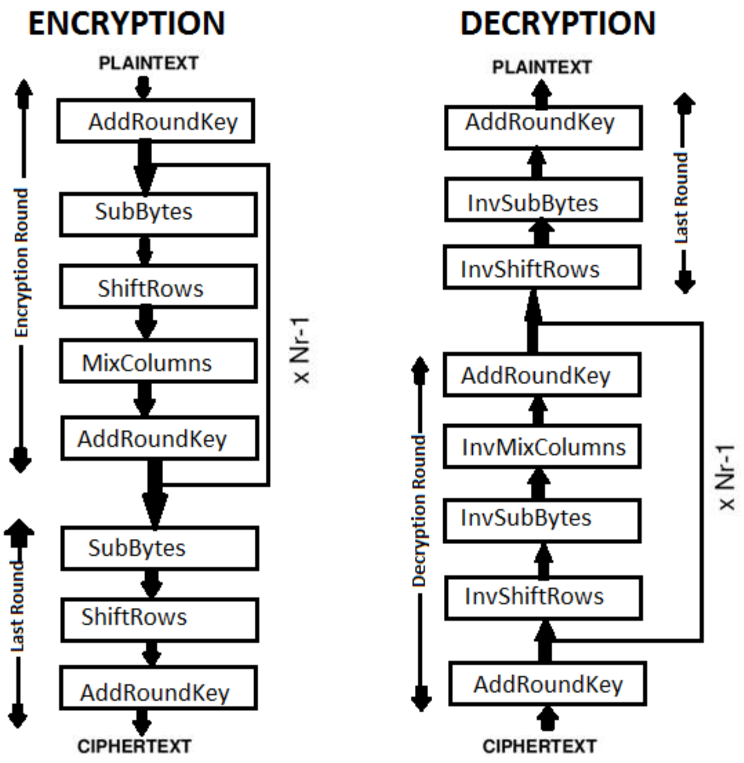
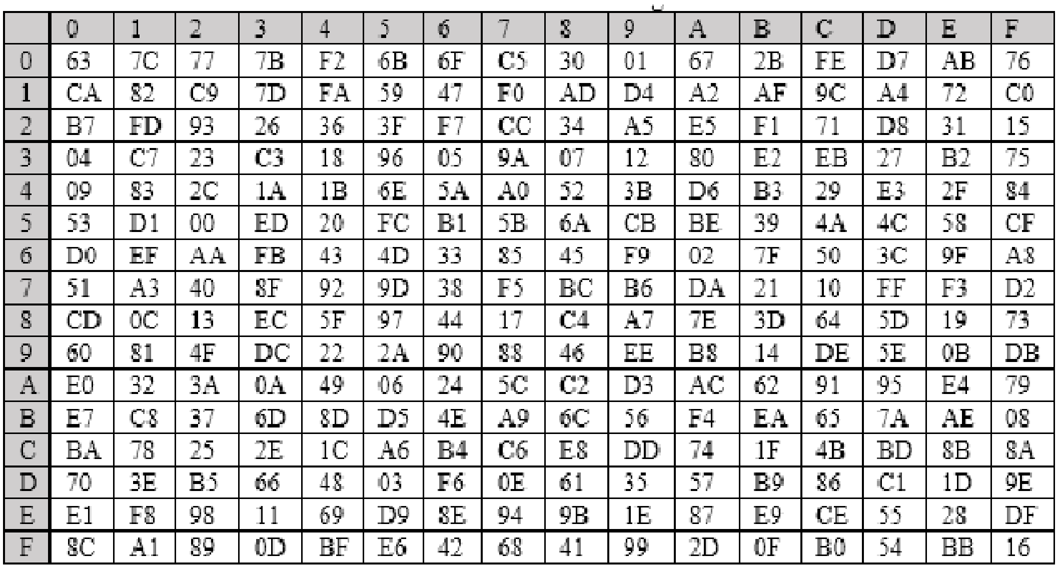


Figure 1. Structure of AES algorithm

1.1. Byte Substitution

The sixteen input bytes are substituted by using fixed look up table known as s-box. Figure 2 shows s-box of AES algorithm. This s-box consists of all possible combinations of 8 bit sequence. The resulting new 16 bytes are organized in a matrix having four rows and four columns. Figure 3 shows byte substitution stage in AES algorithm.



	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

Figure 2. 2 S-box of AES Algorithm

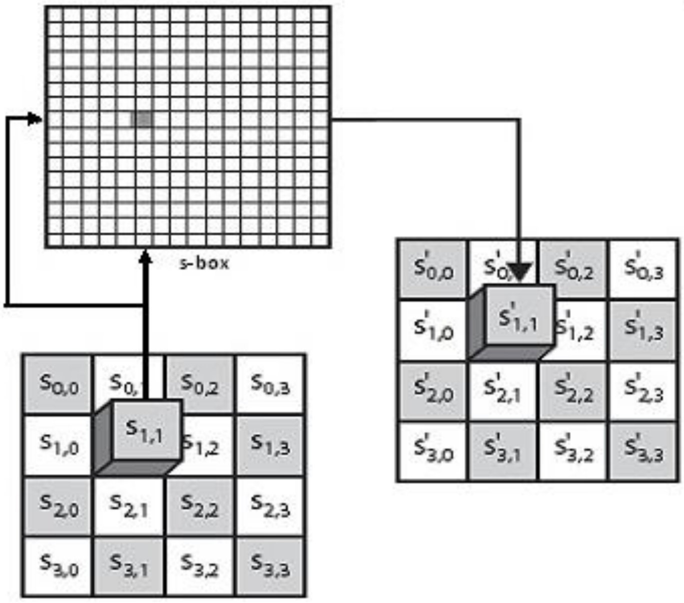


Figure 3. Byte substitution stage

1.2. Shift Row

Each row from the matrix generated from the byte substitution is cyclically shifted to the left. Any entry that is dropped off is reinserted to the right side. 1st row is kept as it is, 2nd row is shifted by one byte position to the left, 3rd row is shifted by two byte position to the left and 4th row is shifted by three byte position to the left. The resultant matrix consists of same 16 bytes but at different position. Figure 4 shows Shift row stage in AES algorithm.

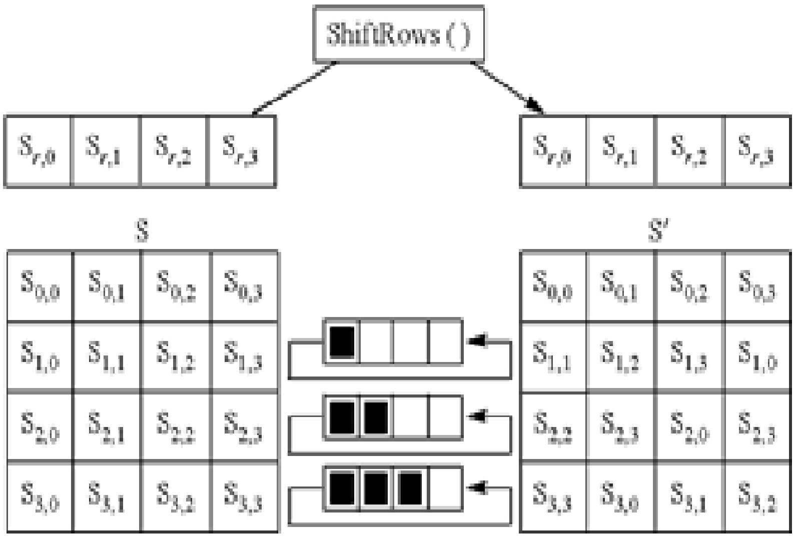
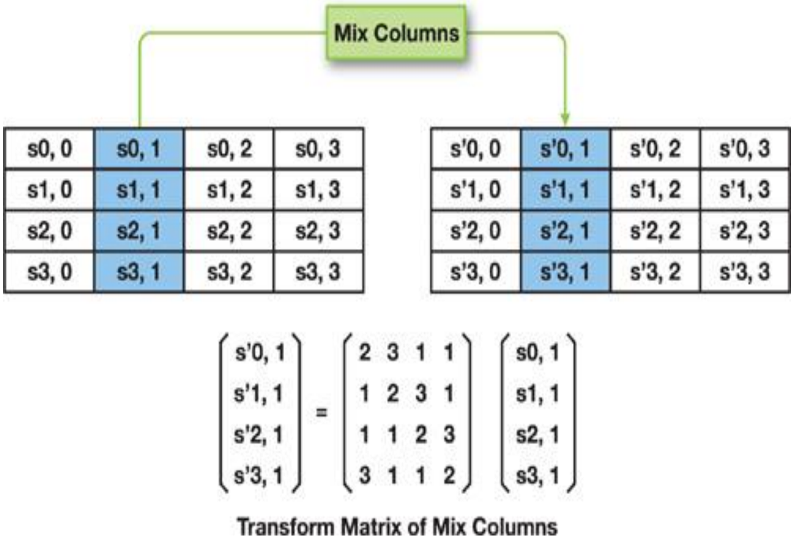


Figure 4. Shift row stage

1.3. Mix Column

Each column of four bytes is now transformed using special arithmetical function of Galois field (GF) 2^8. This function takes four bytes of column as input and outputs completely new four bytes that replaces the original four bytes. Figure 5 shows Mix column stage in AES algorithm.



$$\begin{pmatrix} s'0,1 \\ s'1,1 \\ s'2,1 \\ s'3,1 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} s0,1 \\ s1,1 \\ s2,1 \\ s3,1 \end{pmatrix}$$

Figure 5. Mix column stage

1.4. Add Round Key

The sixteen bytes of the resultant matrix generated from mix column stage are then considered as 128 bits. In add round key stage, 128 bits of state are bitwise EX-ORed with 128 bits of round key. If this result belongs to last round, then the output is ciphertext else the resulting 128 bits considered as 16 bytes and another round is started with new byte substitution process. This is a column wise operation between four bytes of state column and one word of round key. In the last round, there is no mix column step. Figure 6 shows add round key stage in AES algorithm.

Decryption of cipher text generated from AES encryption contains all the stages in encryption but in reverse order. AES decryption starts with inverse initial round. Remaining nine rounds in decryption consists of processes like add round key, inverse shift rows, inverse byte substitution and inverse mix columns.

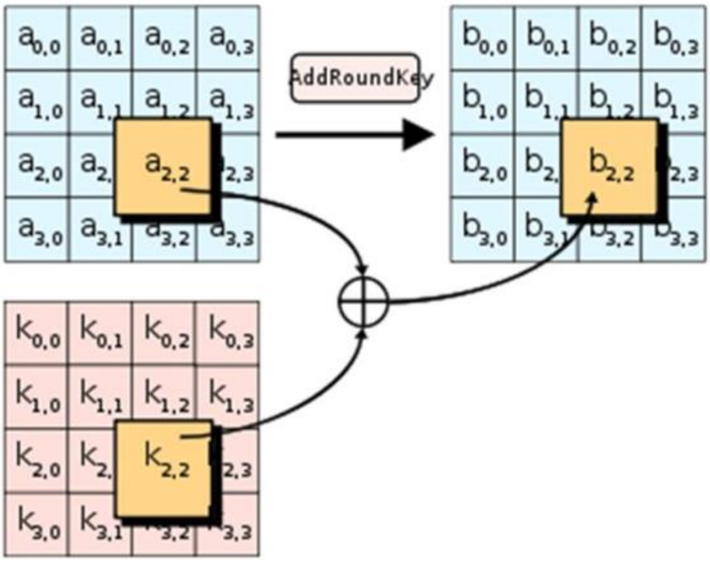

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
AddRoundKey
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,1}$ $k_{2,2}$ $k_{2,3}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$

Figure 6. Add round key stage

Add round key: Add round key has its own inverse function since XOR functions its own inverse and the round keys should be selected in reverse order. Inverse shift rows: Inverse shift rows functions exactly in the same way as shift row stage but in opposite direction. The 1st row is kept as it is, 2nd row is shifted by one byte position to the right, 3rd row is shifted by two byte position to the right and 4th row is shifted by three byte position to the right. The resultant matrix consists of same 16 bytes but at different position. Figure 7 shows Inverse Shift row stage in AES algorithm.

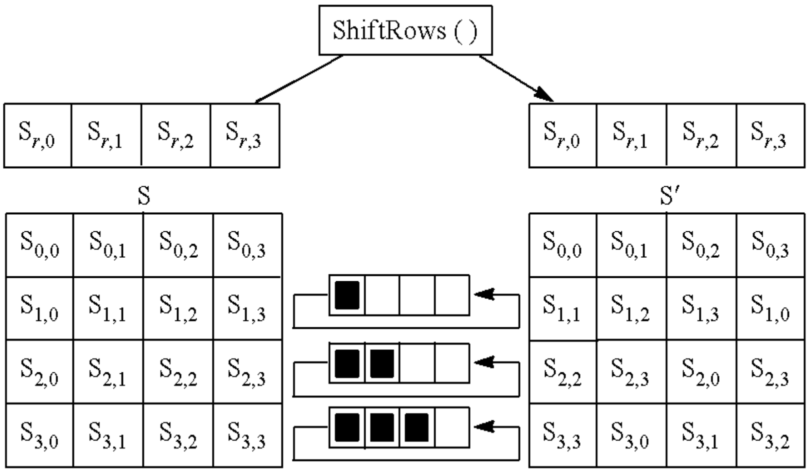
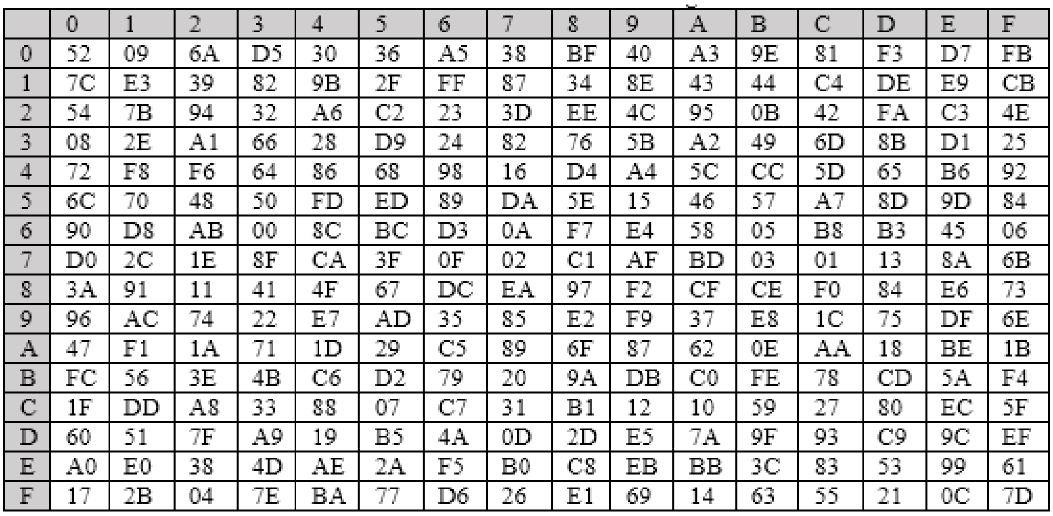

ShiftRows ( )
$S_{r,0}$ $S_{r,1}$ $S_{r,2}$ $S_{r,3}$
$S_{r,0}$ $S_{r,1}$ $S_{r,2}$ $S_{r,3}$
S
S′
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,1}$ $S_{1,2}$ $S_{1,3}$ $S_{1,0}$
$S_{2,2}$ $S_{2,3}$ $S_{2,0}$ $S_{2,3}$
$S_{3,3}$ $S_{3,0}$ $S_{3,1}$ $S_{3,2}$

Figure 7. Inverse Shift row

Inverse byte substitution: Inverse byte substitution is done using predefined substitution table known as inverse s-box. Figure 8 shows inverse s-box in AES algorithm. Inverse mix column: Transformation in inverse mix column is done using polynomials of degree less than 4 over Galois field (GF) 2^8 in which coefficients are the elements from the column of the state.



	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	82	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	89	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	84	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	87	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	38	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

Figure 8. Inverse S-box of AES Algorithm

2. PREVIOUS WORK

In this section, focus is given on work done by various researchers on FPGA based implementation of AES algorithm. There are various researchers which have either concentrated on area optimization or speed optimization. A. O. Mulani et al [1] discussed integrating of DWT and AES algorithm for implementation of watermarking on FPGA. The design was implemented on xc6vcx75t-2ff484 and it utilizes

2117 slices at maximum operating frequency of 228.064 MHz. Raatheesh T. et al [2] proposed implementation of AES algorithm with low power MUX LUT based s-box on FPGA. This design achieved total power distribution of 0.55 W. A. Agarwal et al [4] suggested implementation of AES algorithm using Verilog on Spartan3E FPGA. This design utilizes 1464 slices. U. Farooq et al [5] discussed implementation of AES algorithm on FPGA device using five different techniques which are suitable for area critical applications and speed critical applications. This design was implemented on Spartan-6 FPGA device and it utilizes 161 slices at maximum operating frequency is 886.64 MHz. The throughput of this system is 113.5 Gbps. N. S. Sai Srinivas et al [6] proposed less complex hardware implementation of AES Rijndael algorithm on Xilinx Virtex-7 XC7VX90T FPGA. In the proposed design, synthesis tool was set to optimize speed, area and power.

Nishtha Mathur et al [7] proposed a cryptosystem which is a combination of AES algorithm and ECC. This is a hybrid encryption scheme and the key size is 192 bits and there are 12 number of iterations in this system. K. Kalaiselvi et al [8] proposed low power and high throughput FPGA implementation of AES algorithm using key expansion technique. This design accepts key size of 256 bits for both encryption and decryption. This design utilizes 5493 slices and its maximum operating frequency is 277.4 MHz. The throughput of this system is 0.06 Gbps. H. S. Deshpande et al [9] suggested BRAM based FPGA based implementation of AES algorithm. Due to use of BRAMs for implementing s-box, this design utilizes less number of slices. The design was implemented on XC3S1400AN and it utilizes 3376 slices. Atef Ibrahim [10] presented FPGA implementation of AES encryption core that is suitable for limited resource limited applications. This design was implemented on Spartan-3 and it utilizes 150 slices at maximum operating frequency of 90 MHz. Khose P. N. et al [11] proposed implementation of AES algorithm on FPGA in order to achieve high speed of data processing and also to reduce time for generating key. This design utilizes 201 slices and 2 BRAMs at maximum operating frequency of 70 MHz. A. O. Mulani et al [12] proposed FPGA implementation of DES algorithm. The design was implemented on XC2S200 and it utilizes 2118 slices and 97 IOBs.

Yewale Minal J. et al [13] proposed implementation of AES encryption using VHDL and decryption using Visual basic. With this approach, 1403 slices are utilized at maximum operating frequency of 160.875 MHz and it has a throughput of 2.059 Gbps. H. S. Deshpande et al [14] discussed FPGA based optimized architecture that utilizes less area. This design was intended for plaintext of 128 bits and key of 128 bits. A. R. Tonde et al [15] discussed FPGA based implementation of AES algorithm using iterative looping approach for 128 bits of block and key size. Sonali A. Varhade et al [18] proposed FPGA based AES algorithm which utilizes 1746 logic elements and 32768 memory bits. This design was synthesized on Cyclone-II using Altera. Salim M Wadi et al [19] proposed some modifications like decreasing number of rounds and replacing S-box with new s-box to reduce hardware requirements in order to enhance the performance of AES algorithm in terms of time ciphering and pattern appearance.

Wei Wang et al [21] suggested high speed implementation of AES algorithm on FPGA to transmit the data securely using pipelining and parallel processing methods. Shylashree N. et al [22] focused on various novel FPGA architectures of AES algorithm. Borkar A. M. et al [23] proposed iterative design approach for FPGA implementation of AES algorithm using VHDL. This design utilizes 1853 slices and its operating frequency is 140.390 MHz. A. M. Deshpande et al [24] presented very low complexity FPGA base architecture for integrated AES encryptor and decryptor. This design is synthesized on Spartan-3 XC3S400 FPGA. S. Kaur et al [25] suggested an efficient implementation of AES algorithm on FPGA in which multiple rounds are processed simultaneously. Due to this implementation, speed is increased but it increases area. This design utilizes 6279 slices and 5 BRAMs and its operating frequency is 119.954 MHz. Sounak Samanta [26] proposed fast and efficient reconfigurable platform based implementation of AES algorithm using pipelining. This design utilizes 1051 slices and 11 BRAMs and its operating frequency is 76.699 MHz. T. Good et al [27] discussed hardware implementation of fastest and slowest AES algorithm which utilizes 16,693 slices at maximum operating frequency of 184.8 MHz.

3. IMPLEMENTATION OF PROPOSED DESIGN

The proposed design is implemented with the aim to achieve both area and speed optimization. This is achieved by generating the keys required for each round using MATLAB and then the keys are used in the VHDL code. Due to this approach, the design occupies less number of slices and also the speed is faster as compared to normal approach. The design is implemented using Xilinx system generator. Figure 9 shows Xilinx system generator based Simulink model for AES algorithm

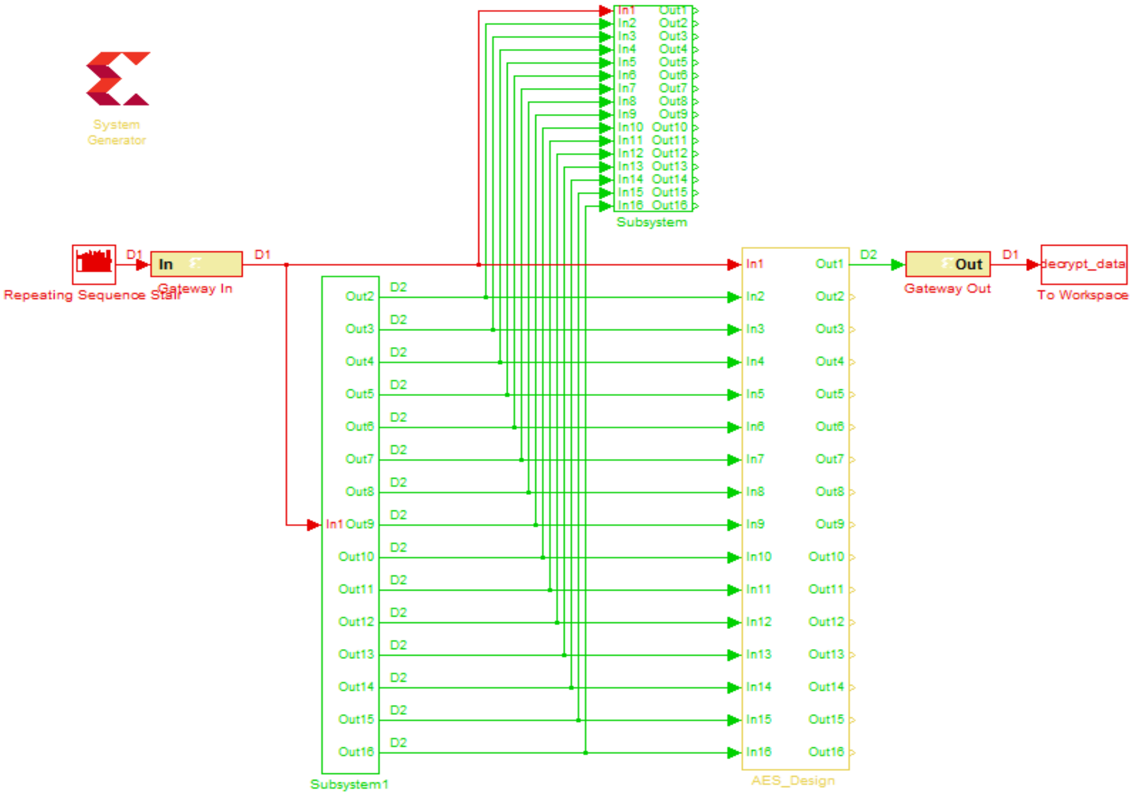


Figure 9. System Generator based Simulink model for AES algorithm

4. EXPERIMENTAL RESULTS

4.1. RTL Schematic

Figure 10 shows detailed RTL schematic of AES algorithm.

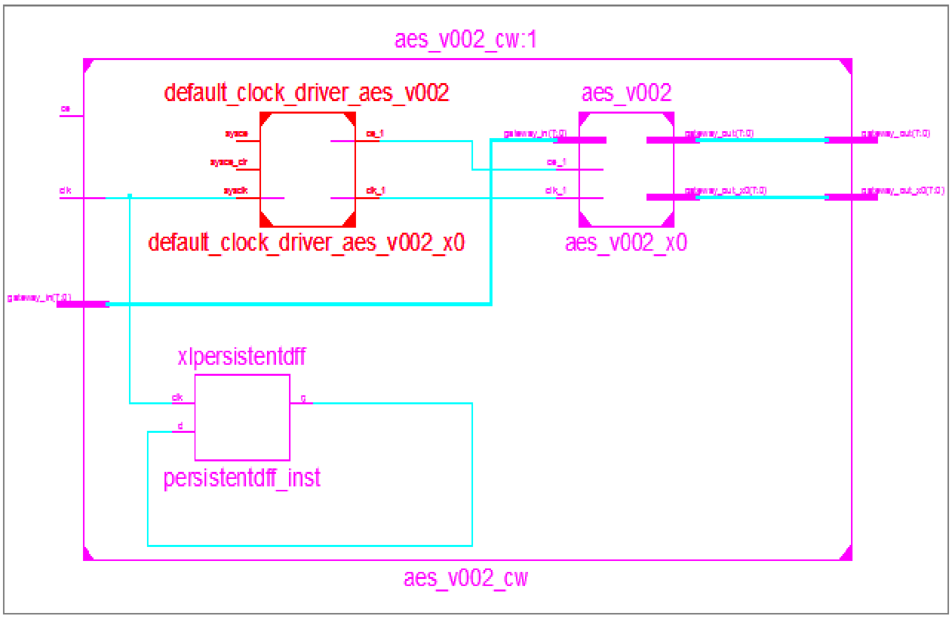


Figure 10. Detailed RTL Schematic of AES algorithm

4.2. RTL Schematic

The design is synthesized using Xilinx XST synthesizer. In the proposed design, an optimized and synthesizable VHDL code for the implementation of image as well as 128-bit data encryption is developed so as to utilize less area and increase the speed. Table 1 shows design utilization summary of proposed design.

Table 1. Design Utilization Summary

Design Utilization Summary			
Logic Utilization	Used	Available	% Utilization
Number of Slice registers	121	126800	0
Number of slice LUTs	4782	63400	7
Number of bonded IOBs	25	210	11

From the synthesis results of the proposed design, it is clear that this system utilizes only 121 slice registers and its maximum operating frequency is 1102.536 MHz. The throughput of the system is calculated using the following formula:

$$\text{Throughput of the system} = \frac{128\text{ bits} \times \text{Clock frequency}}{\text{Cycles per Encrypted block}} \quad (1)$$

By substituting the values in equation (1), throughput of the systems is 14.1125 Gbps.

5. PERFORMANCE ANALYSIS

Performance analysis is must to compare the performance of proposed implementation with existing methods. The performance is compared on the basis of area and operating frequency. Till date various researchers have worked on FPGA based implementations of AES algorithm, some of them have optimized speed and some have optimized area. In this proposed system, both area and speed is optimized. Table 2 shows performance comparison of proposed system with previous work.

Table 2. Performance Comparison of Proposed System with Previous Work

Sr. No.	Authors	Slices	Operating Freq. (MHz)
1	Proposed work	121	1102.536
2	[4]	1464	--
3	[5]	161	886.64
4	[8]	5493	277.4
5	[9]	3376	--
6	[10]	150	90
7	[11]	201	70
8	[13]	1403	160.875
9	[18]	1746	--
10	[23]	1853	140.390
11	[25]	6279	119.954

6. CONCLUSION

In this paper, fast and secure implementation of AES algorithm on FPGA is suggested. As per the literature survey, it is clear that [5] achieves better performance in terms of speed whereas [10] achieves better performance in terms of area. In this design, due to offline key generation and better Xilinx System Generator based design the system is optimized and it utilizes only 121 slice registers at maximum operating frequency of 1102.536 MHz. Also, throughput of the proposed system is 14.1125 Gbps.

REFERENCES

[1] Altaf O. Mulani and P.B.Mane, “Watermarking and Cryptography based Image Authentication on reconfigurable platform”, Bulletin of Electrical Engineering and Informatics, June 2017.

[2] Ratheesh T. and Seena Narayanan, “FPGA based implementation of AES Encryption and Decryption with low power multiplexer LUT based S-box”, *IOSR Journal of Electronics and Communication Engineering*, April 2017.

[3] Altaf O. Mulani and Dr. P. B. Mane, “Area Efficient high speed FPGA based invisible watermarking for image authentication”, *Indian Journal of Science and Technology*, Vol.9 (39), October 2016.

[4] Abhinandan Agarwal, Gagandeep Singh and Prof. (Dr.) Neelam Sharma, "Implementation of AES algorithm", *International Journal of Engineering Research and Science (IJOER)*, April 2016.

[5] U. Farooq and M. Faisal Aslam, "Comparative analysis of different AES implementation techniques for efficient resource usage and better performance of an FPGA", *Journal of King Saud University-Computer and Information Sciences*, March 2016.

[6] N.S.Sai Srinivas and Md. Akramuddin, "FPGA Based Hardware Implementation of AES Rijndael Algorithm for Encryption and Decryption", *IEEE International Conference on Electrical, Electronics and Optimization Techniques*, March 2016.

[7] Nishtha Mathur and Rajesh Bansode, "AES based Text Encryption using 12 rounds with dynamic key selection", *International Conference on Communication, Computing and Virtualization*, 2016.

[8] K. Kalaiselvi and H. Mangalam, "Power Efficient and high performance VLSI Architecture for AES Algorithm", *Journal of Electrical Systems and Information Technology*, September 2015.

[9] H.S.Deshpande, Dr.K.J.Karande and A.O.Mulani, "Area Optimized Implementation of AES Algorithm on FPGA", *IEEE International Conference on Communications and Signal Processing (ICCSP)*, April 2015.

[10] Atef Ibrahim, "FPGA Based Hardware Implementation of Compact AES Encryption Hardware Core", WSEAS Transactions on Circuits and Systems, 2015.

[11] Khose P.N. and Raut V.G., "Implementation of AES algorithm on FPGA for low area consumption", *IEEE International Conference on Pervasive Computing (ICPC)*, January 2015.

[12] Altaf O. Mulani and Dr. P. B. Mane, "Area optimization of Cryptographic algorithm on less dense reconfigurable platform", *IEEE International Conference on Smart Structures and Systems (ICSSS)*, October 2014.

[13] Yewale Minal J. and M. A. Sayyad, "Implementation of AES on FPGA", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), October 2014.

[14] H.S.Deshpande, Dr.K.J.Karande and A.O.Mulani, "Efficient Implementation of AES Algorithm on FPGA", *IEEE International Conference on Communications and Signal Processing (ICCSP)*, April 2014.

[15] Ashwini R. Tonde and Akshay P. Dhande, "Review paper on FPGA based implementation of Advanced Encryption Standard (AES) Algorithm", *International Journal of Advanced Research in Computer and Communication Engineering*, January 2014.

[16] Anup Gujar, "Image encryption using AES algorithm based on FPGA", *International Journal of Computer Science and Information Technologies (IJCSIT)*, 2014.

[17] Ritu Pahal and Vikas Kumar, "Efficient Implementation of AES algorithm", *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, July 2013.

[18] Sonali A. Varhade and N. N. Kasat, "Implementation of AES algorithm using FPGA and its performance analysis", *International Journal of Science and Research*, May 2013.

[19] Salim M. Wadi and Nasharuddin Zainal, "Rapid Encryption method based on AES algorithm for Grey scale HD image encryption", *International Conference on Electrical Engineering and Informatics*, 2013.

[20] Sliman Arrag, Abdellatif Hamdoun, Abderrahim Tragha and Salah Eddine Khamlich, "Several AES variants under VHDL language in FPGA", *International Journal of Computer Science Issues (IJCSI)*, September 2012.

[21] Wei Wang, Jie Chen and Fei Xu, "An Implementation of AES Algorithm based on FPGA", *IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, May 2012.

[22] Shylashree N., Nagarjun Bhat and V. Shridhar, "FPGA Implementations of Advanced Encryption Standard: A Survey", *International Journal of Advances in Engineering and Technology (IJAET)*, May 2012.

[23] Borkar A. M., Kshirsagar R.V. and Vyawahare M.V., "FPGA implementation of AES algorithm", *IEEE International Conference on Electronics Computer Technology (ICECT)*, April 2011.

[24] Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar, "FPGA Implementation of AES Encryption and Decryption", *IEEE International Conference on Control, Automation, Communication and Energy Conservation*, June 2009.

[25] Kaur Swinder and Vig R., "Efficient Implementation of AES algorithm in FPGA Device", *IEEE International Conference on Computational Intelligence and Multimedia Applications*, December 2007.

[26] Sounak Samanta, "FPGA implementation of AES Encryption and Decryption", Sardar Vallabhbhai National Institute of Technology, Surat, 2007.

[27] T. Good, M. Benaissa, "AES on FPGA from fastest to smallest", Proceedings of International Workshop on Cryptographic Hardware and Embedded systems, 2005.

[28] Xinmiao Zhang and and Keshab K. Parhi, "High Speed VLSI Architectures for the AES Algorithm", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 9, September 2004.

[29] S. M. Farhan, S. A. Khan and H. Jamal, "Mapping of High bit algorithm to low bit for optimized hardware implementation", *IEEE International Conference on Microelectronics (ICM)*, December 2004.

[30] Adam J. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA Based Performance Evaluation of AES Block Cipher Candidate Algorithm Finalists", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol.9 No.4, August 2001.