IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 5, Ver. II (Sep-Oct. 2014), PP 65-69
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Implementation of AES on FPGA

Yewale Minal J[1], M. A. Sayyad[2]
[1](Department of E&TC, S.R.E.S. College of Engineering, Pune University, India)
[2](Department of E&TC, S.R.E.S. College of Engineering, Pune University, India)

Abstract: *Advanced Encryption Standard (AES) a National Institute of Standards and Technology specification is an approved cryptographic algorithm that can be used for securing electronic data. Reprogrammable devices such as Field Programmable Gate Arrays (FPGA) are highly attractive option for hardware implementations of cryptographic algorithm AES as they offer a quicker and more customizable solution. This paper proposes an efficient FPGA implementation of advanced encryption standard (AES). We implement the AES encryption algorithm on Xilinx Spartan-3 FPGA and decryption is done on PC. The coding for encryption is done in VHDL language and for decryption in Visual Basic. To implement AES Rijndael algorithm on FPGA plain text of 128 bit data is considered. Advanced Encryption Standard (AES) RIJNDAEL on FPGA offers a better performance than any other cryptographic algorithms.*

Keywords: *AES Rijndael algorithm, Decryption, Encryption, FPGA.*

I. INTRODUCTION

Ever since the Internet has emerged securing data from unauthorized access is of concern and very important. As each type of data has its own features different methods should be used to protect confidential data from unauthorized access. Each and every day millions of users generate and sends large volume of a data in various fields, such as financial documents, research data in geographical field, legal files and medical reports via Internet. Most of this information is stored on electronic computers and sent to other computer through internet. While transmitting or sending this data which could be confidential could be accessed by unauthorized person. To avoid the confidential data fall into the wrong hands vital step should be taken and is none other than securing the data. Securing confidential data is a vital, ethical and legal requirement. Also with the development of various multimedia technologies the multimedia data are generated and transmitted, via internet these digital data can be widely distributed. It becomes much easier to edit, change and copy digital information by unauthorized Internet users. These and other examples need a security. Cryptography considers one of the techniques which is used to protect the important information [1].Cryptography is the art and science of protecting information from undesirable individuals by converting it into a form non-recognizable when stored and transmitted [2]. Cryptography is the science of securing data. Data that can be read and understood without any difficulty is called plaintext or original data. Plain text converted into an unintelligible form called cipher text which is unreadable by humans & machines until it is decrypted. The method of disguising plaintext in such a way as to hide its substance is called Encryption.That is, scrambling of the content of data (Plain text) such as text, image, audio, video to make the data unreadable, unintelligible (Cipher text) during transmission or storage. The process of reverting ciphertext to its original plaintext is called decryption. Many encryption algorithms have been developed due to the large amount of information data, among the various cryptographic algorithms, the most popular example is the Data Encryption Standard (DES) algorithm. DES could not keep up with advancement in a technology and it is no longer appropriate for security. But as DES was used mostly at that time the quick solution was to introduce Triple DES which is secure enough for most purposes today. Triple DES takes three times as much CPU power than compare with its predecessor which is significant performance hit .AES outperforms 3 DES both in software and in hardware [1]. The U.S. National Institute of Standards and Technology (NIST) conducted a competition to develop a replacement for DES. The Rijndael algorithm was a winner and destined to become the new Advanced Encryption Standard. In the last phase of the selection, there were five finalist algorithms: Mars, RC6, Rijndael, Serpent and Twofish. Because all the algorithms were considered secure, hardware efficiency was given great importance in selecting Rijndael as the winning algorithm. This algorithm is documented in the freely available US government publication, FIPS-197 [3]. The hardware-based implementation of AES Rijndael algorithm is required because it can be more secure and consumes less power than software implementation [4]. Design and Implementation of Advanced Encryption Algorithm with FPGA and ASIC is explained in the paper of Leelavathi.G, Prakasha.S [5]. In it a throughput of a FPGA is faster compared to the previous FPGA implementations. The hardware-based implementation of AES Rijndael algorithm is required because it can be more secure and consumes less power than software implementation. F.X.Standaert et al. [6] proposes a methodology to efficiently implement block cipher within commercially available FPGA and it is applied to design AES Rijndael which is shown to improve previously reported results in terms of hardware cost, throughput or efficiency.

II. DESCRIPTION ON AES ALGORITHM

The AES is a 128 bit symmetric block cipher with three different key lengths of 128,192, or 256 bit. It performs encryption and decryption processes on an iterative basis, every iterative step is known as an operation round. For AES algorithm, number of rounds to be performed during the execution of the algorithm is dependent on the key length [4]. It is equal to 10 rounds for 128-bit, 12 for 192 bit and 14 for 256 bit keys.

The AES algorithm operates on a changeable square array of dimension 4×4, known as the state array [7]. The AES algorithm is composed of three steps namely Key Expansion, Cipher also known as encryption process, Inverse Cipher known as decryption process. Key Expansion generates a Key Register that is used in Encryption and decryption process. In each round, a different round key is being used. Before any round-based processing for encryption can begin, the input state array is XORed with the first four words of the key schedule. The same thing happens during decryption except that now we XOR the ciphertext state array with the last four words of the key schedule. The remaining 40 words of the key schedule are used four words at a time in each of the 10 rounds.

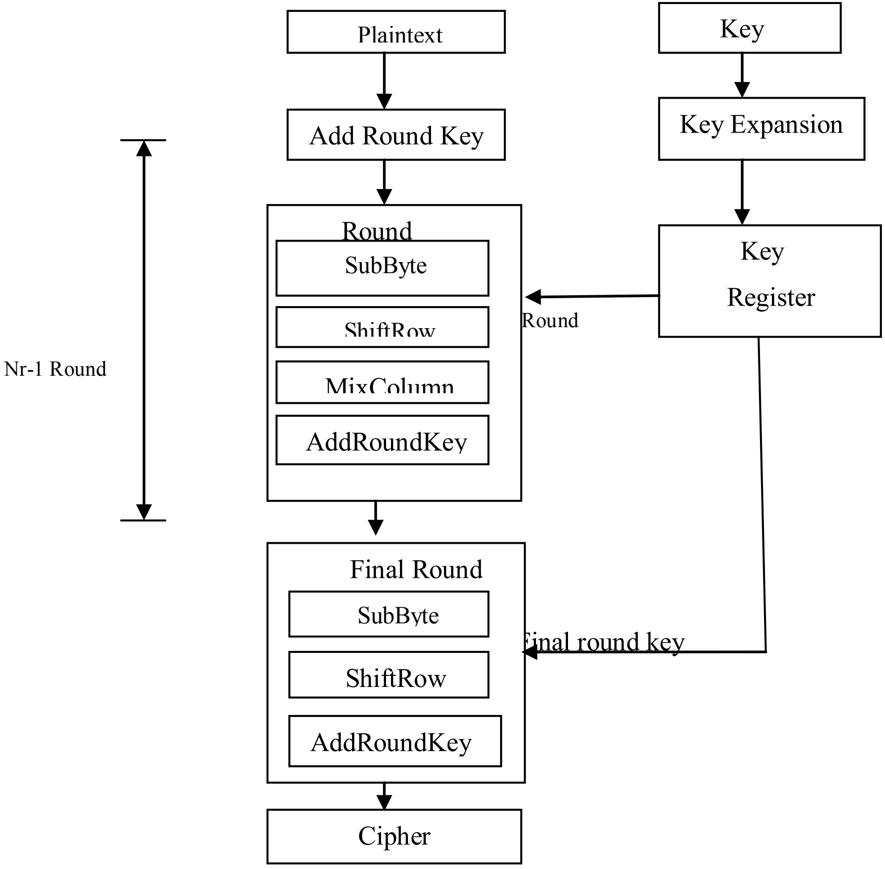


Fig. 1 The structure of AES Encryption

The encryption process starts by setting plaintext in the state array then perform changes and produces the encrypted data i.e. ciphertext. AES encryption includes an initial round, Nr-1 normal rounds and a final round. In the decryption process, the state begins with the ciphertext and performs changes until the plaintext is recovered.

A normal round is composed of four different transformations: SubByte, ShiftRow, MixColumn and AddRoundKey. Each Encryption processes these four transformations. After an initial Add Round Key the State array is transformed by implementing a round function 10. In the final (10th) round, there is no Mix-column transformation. The final State is then copied to the output. The structure of AES Encryption is shown in figure 1. In the decryption process, the state begins with the ciphertext and performs changes until the plaintext is recovered. The decryption process is exact opposite process of encryption. The decryption which performs the reverse consists of inverse S-box, inverse Shift Rows, inverse Mix Columns and Add Round Key. The final round is equal to the normal round except that inverse MixColumn is omitted.

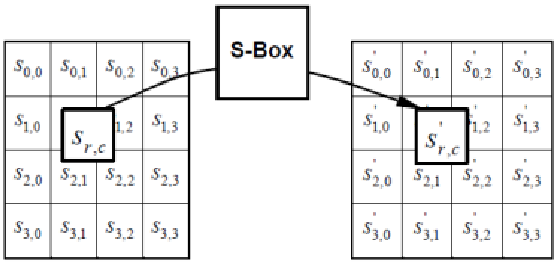

S-Box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{r,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{r,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig. 2 The S-box applied to each byte of the State

SubByte transformation (S-box) can be implemented as a look-up table. This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation [8]. AES contains 128 bit data block, which means each of the data blocks has 16 bytes. Figure 2 shows each byte (8-bit) of a data block is transformed into another block using an 8-bit substitution box which is also known as Rijndael Sbox (matrix). In ShiftRow transformation bytes in the last three rows of the state, depending upon the row location, are cyclically shifted. For 2nd row, 1 byte circular left shift is performed for 3rd row, 2 bytes circular left shift is performed and for 4th, 3 bytes circular left shift is performed .In MixColumn transformation operation the bytes are taken as polynomials rather than numbers. In the AddRoundKey transformation, AddRoundKey XORs each column of the State with a word from the key schedule.This transformation is its own inverse [9].

III. IMPLEMENTATION

The design of AES is done using VHDL and implemented on a Spartan-3 XC3S400 device with package PQ208 FPGA using the ISE 8.2i design tool. Generally,128 bit plaintext is an input for AES, but AES algorithm is processed when the inputs are in bytes. So, input is in 16 bytes. The basic unit for processing in the AES algorithm is a byte, a sequence of eight bits treated as a single entity. In this the input from the keyboard is considered to be encoded in Hexadecimal, thus the only valid characters are 0-9, A-F.

The experimental setup is shown in figure 3. PC is used for sending a plain text or a input block and a cipher key with the help of Visual Basic program to Spartan-3 XC3S400-FG456 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. For encryption purpose FPGA kit is used. In software design, coding is done in VHDL 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 XC3S400 device. In Boundary Scan Mode of configuration the Spartan-3 FPGA is directly configured via a JTAG port using the pins TCK,TMS,TDI,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 ciphertext from FPGA to PC.

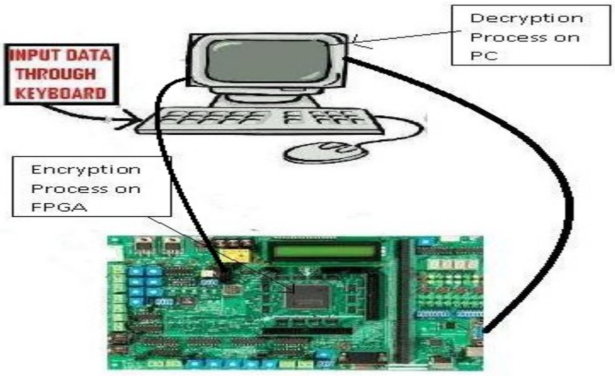
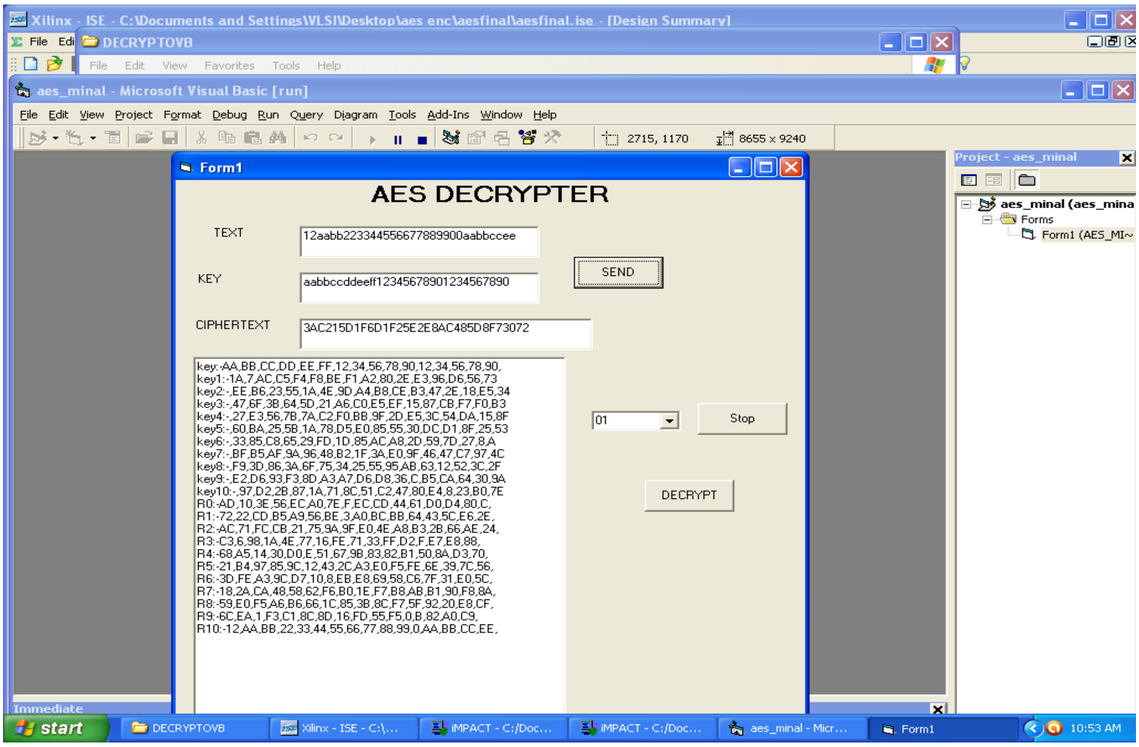


Fig 3. System experimental setup

3.1 RESULT

These both inputs i.e., Plaintext & Cipher Key will be sent to FPGA. The AES encryption algorithm implemented in FPGA will process on the Plaintext & Cipher Key.

INPUTS-
PLAINTEXT: 12aabb2233445566778899 00aabbccee
KEY: aabbccddeeff1234567890 1234567890


Xilinx - ISE - C:\Documents and Settings\VLSI\Desktop\aes enc\aesfinal\aesfinal.ise - [Design Summary]
DECRYPTOVB
aes_minal - Microsoft Visual Basic [run]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
2715, 1170
8655 x 9240
Form1
Project - aes_minal
aes_minal (aes_mina
Forms
Form1 (AES_MI~
Immediate
start
Xilinx - ISE - C:\...
iMPACT - C:/Doc...
aes_minal - Micr...
10:53 AM

AES DECRYPTER

TEXT: 12aabb22334455667788990 0aabbccee

KEY: aabbccddeeff12345678901234567890

CIPHERTEXT: 3AC215D1F6D1F25E2E8AC485D8F73072

SEND

01 ▾ Stop

DECRYPT

```
key:-AA,BB,CC,DD,EE,FF,12,34,56,78,90,12,34,56,78,90,
key1:-1A,7,AC,C5,F4,F8,BE,F1,A2,80,2E,E3,96,D6,56,73
key2:-,EE,B6,23,55,1A,4E,9D,A4,B8,CE,B3,47,2E,18,E5,34
key3:-,47,6F,3B,64,5D,21,A6,C0,E5,EF,15,87,CB,F7,F0,B3
key4:-,27,E3,56,7B,7A,C2,F0,BB,9F,2D,E5,3C,54,DA,15,8F
key5:-,60,BA,25,5B,1A,78,D5,E0,85,55,30,DC,D1,8F,25,53
key6:-,33,85,C8,65,29,FD,1D,85,AC,A8,2D,59,7D,27,8,A
key7:-,BF,B5,AF,9A,96,48,B2,1F,3A,E0,9F,46,47,C7,97,4C
key8:-,F9,3D,86,3A,6F,75,34,25,55,95,AB,63,12,52,3C,2F
key9:-,E2,D6,93,F3,8D,A3,A7,D6,D8,36,C,B5,CA,64,30,9A
key10:-,97,D2,2B,87,1A,71,8C,51,C2,47,80,E4,8,23,B0,7E
R0:-AD,10,3E,56,EC,A0,7E,F,EC,CD,44,61,D0,D4,80,C,
R1:-72,22,CD,B5,A9,56,BE,3,A0,BC,BB,64,43,5C,E6,2E,
R2:-AC,71,FC,CB,21,75,9A,9F,E0,4E,A8,B3,2B,66,AE,24,
R3:-C3,6,98,1A,4E,77,16,FE,71,33,FF,D2,F,E7,E8,88,
R4:-68,A5,14,30,D0,E,51,67,9B,83,82,B1,50,8A,D3,70,
R5:-21,B4,97,85,9C,12,43,2C,A3,E0,F5,FE,6E,39,7C,56,
R6:-3D,FE,A3,9C,D7,10,8,EB,E8,69,58,C6,7F,31,E0,5C,
R7:-18,2A,CA,48,58,62,F6,B0,1E,F7,B8,AB,B1,90,F8,8A,
R8:-59,E0,F5,A6,B6,66,1C,85,3B,8C,F7,5F,92,20,E8,CF,
R9:-6C,EA,1,F3,C1,8C,8D,16,FD,55,F5,0,B,82,A0,C9,
R10:-12,AA,BB,22,33,44,55,66,77,88,99,0,AA,BB,CC,EE,
```

Fig.4 AES output

Then after ten rounds of the AES, the ciphertext will appear as
ENCRYPTION OUTPUT-
CIPHERTEXT: 3AC215D1F6D1F25E2E8AC48508F73072 as shown in figure 4.

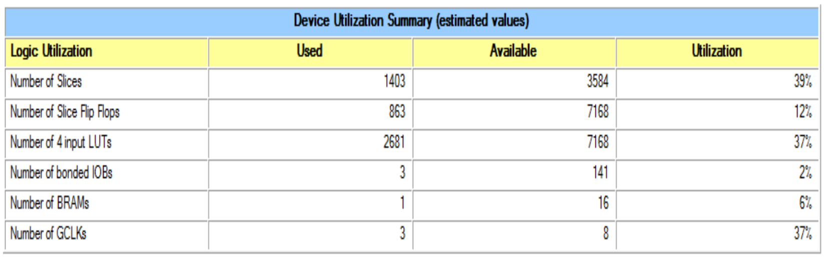
For decryption we use ciphertext as input and use the same cipher key for decryption algorithm. The decryption is done on pc. When decryption is done, we get decryption output that is original data. Original data is obtained after ten rounds R10 of AES as shown in figure 4.

DECRYPTION OUTPUT-
ORIGINAL DATA : 12AABB2233445566778899 00AABBCCEE

3.2 Comparision With Previous Work

Table 1 shows the overall device utilization of the FPGA .Table 2 shows the timing report through which throughput is achieved [5]. Timing report shows the time required for AES algorithm to execute.

Table 1 Resources utilization



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1403	3584	39%
Number of Slice Flip Flops	863	7168	12%
Number of 4 input LUTs	2681	7168	37%
Number of bonded IOBs	3	141	2%
Number of BRAMs	1	16	6%
Number of GCLKs	3	8	37%

Table 2 Timing reports

Speed Grade	-5
Minimum period	8.539ns
Minimum input arrival time before clock:	5.375ns
Maximum output required time after clock:	6.216ns

3.2.1 Calculation for frequency and throughput

Maximum output required time after clock (T) = $6.216*10^{-9}$

$$\text{Frequency} = 1/T \qquad (1)$$

$$=1/(6.216*10^{-9})$$

Frequency=160.875Mhz

$$\text{Throughput for 128 bit AES} = (128*F)/10 \qquad (2)$$
$$= (128*160.875)/10$$
$$= 2.059 \text{ Gbps}$$
$$\text{Throughput per slices (Mbps/slice)} = (2059*10^6)/1403 \qquad (3)$$
$$= 1.467$$

Table 3 shows the comparison of our work with the previous work. Comparison with other AES Encryptor Design is done in Table 3. From the comparisons, our implementation achieves a throughput of 2059 Mbps. Our AES encryptor outputs each ciphertext for working frequency 160 MHz. The throughput of our design is very close to [5] but our design needs less the hardware slices than [5]. Therefore, our AES encryptor design is more efficient than the published approaches in terms of the efficiency of throughput/area.

Table 3 Comparison with previous work

Design	Device	Frequency	Throughput	Slices	Throughput/Area
			(Mbits/s)		(Mbps/slices)
Standaert et al. design[6]	Virtex 1000		1563	2257	0.69
Shuenn-Shyang Wang et al. [8]	Virtex-E BG860		1604	1857	0.867
Leelavathi et al. [5]	XC3S500E-4FG320	222.41	2846	2439	1.166
Ours	XC3S400-FG456	160.875	2059	1403	1.467

IV. Conclusion

The Advanced Encryption Standard algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128,192,256 bits. An efficient FPGA implementation of 128 bit block and 128 bit key AES encryption algorithm is done. The AES cipher is implemented on the FPGA provides an excellent platform for high security applications. A synthesizable VHDL code is developed for the implementation of encryption process and is synthesized by using Xilinx 8.2.The AES encryption uses 1403 slices and operates at 2059 Mbps (Throughput).Whereas, the decryption process is done in a software platform.

This Implementation of 128 bit AES using Rijndael algorithm, and the same can be extended to encrypt 192 and 256 bits of plain text data with proper key length, which makes even tougher to decrypt the original data form an unauthorized receivers.

For a future scope this approach can be extended to make an AES encrypted block which can be used as an ASIC for AES implementation in advanced microprocessors or microcontrollers.

References

[1] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani, "New Comparative Study between DES, 3DES and AES within Nine Factors", *Journal of Computing, Volume 2, Issue 3, March 2010, ISSN 2151-9617*

[2] A.A.Zaidan, B.B.Zaidan, Anas Majeed, "High Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm", *World Academy of Science Engineering and Technology (WASET), Vol.54, ISSN: 2070-3724, P.P 468-479.*

[3] National Institute of Standards and Technology, Advanced Encryption Standard (AES), *Federal Information Processing Standards Publications – FIPS 197*

[4] J. Daemen and V. Rijmen, *AES submission document on Rijndael, Version 2, September 1999.(http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf)*

[5] Leelavathi.G, Prakasha.S, Shaila.K, Venugopal K.R, L M Patnaik," *Design and Implementation of Advanced Encryption Algorithm with FPGA and ASIC", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 3, June-July, 2013*

[6] F.X.Standaert, "A Methodology to implement block ciphers in reconfigurable hardware and as application to fast and compact AES Rijndael. "*The field programmable logic array conference, Monterey, California, pp.216-224. 2003.*

[7] Gaj, Kris, Pawel Chodowiec, "FPGA and ASIC implementations of AES", *Cryptographic Engineering*, Springer *US, 2009, 235-294.*

[8] Sheunn-Shyang Wang and Wan-Sheng Ni, "An Efficient FPGA implementation of advanced encryption Standard Algorithm", *International Symposium on circuits and systems, IEEE, pp 597, 2004.*

[9] A. Amaar, I. Ashour and M. Shiple," Design and Implementation A Compact AES Architecture for FPGA Technology", *World Academy of Science, Engineering and Technology 59 2011*