ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 83-88 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

128-bit Advanced Encryption Standard Algorithm implementation on FPGA

P.Ravisankararao[1], Dr. M.V.Nageswararao[2]
[1]Department of ECE, India,pravisankar471@gmail.com
[2]Department of ECE, India, nageswararao.mv@gmrit.org

Abstract: In any communication, data Security plays a vital role and there is a need to protect data from malicious attacks. This paper presents 128 bit pipelining processing of Advanced Encryption Standard (AES).This implementation is compared with previous work iterative looping to show better efficiency. Encryption and decryption was carried out with the key length of 128 bits lookup table implementation of S-box.Proposed design is implemented with minimum memory, area to achieve the high throughput.

Keywords: Advanced Encryption Standard (AES),Sequential design, AES Pipeline, Parallel Processing, and FPGA.

INTRODUCTION

Cryptography is the science of secret codes, which enables the secret communication through an insecure channel.Communication parties must agree on a secret key before they wish to communicate. In generally, it uses a cryptographic system to transform a plaintext into a cipher text, using a secret key.Advanced Encryption Standard (AES) algorithm [1] is an encryption standard used for securing information.

AES is a block cipher algorithm that has been analyzed extensively and is now used widely. AES is a symmetric block cipher that is intended to replace Data Encryption Standard (DES) algorithm [2] as the approved standard for a wide range of applications.

Rijndael is very secure and has no known weakness. Rijndaelis conventional symmetric key [3]system and is relatively simple cipher in many respects. It takes an input block of a certain size, usually 128, and produces a corresponding output block of the same size.

The AES transformation [4] requires another second input, which is the secret key. It is important to know that the secret key. In this work, both encryption and decryption will be carried out with the key length of 128 bits. Hence the input block and secret key will be provided for encryption and the cipher block and same secret key will be provided to the decryption to get the proper block as output.

Earlier, software implementation of the AES begins with various loops controlled architectures [5], [6] where Hugh tables were generated with limited no of architecture optimizations. Thus, a pioneer attempts from V.Rijmen [6]proposed an s-box implementation was considered as the first step in compaction the AES implementation [7].

After that AES implementations [8] were done based on the iterative looping approach [9]. In this all the four AES transformation rounds are executed in a sequential fashion. Throughput of the sequential design is reduced since a cipher block is produced every 10 cycles i.e. since delay is more.

To reduce the delay, we proposed a pipelining processof AES algorithm. The first design does not having the advantage of pipeline and strictly use iterative looping approaching contrast, to second design which uses pipeline buffers between the 10 AES stages. Because of applying pipelining design the resulting speed in terms of throughput rate and implementation area [10], [11] is evaluated and compared with the iterative looping design implementation.

Analysis and performance evaluations were adequately performed by iterative looping algorithm on the same FPGA [12] device and compared thoroughly with our proposed design in terms of area [13], [14]and throughput [15].

The paper is organized as follows; in this section 2 we present the brief interdiction of AES encryption and decryption algorithm, iterative looping architecture of the AES is described in section 3. Section 4 presents experimental results and discussion. Finally we present the conclusion in section5.

AES ALGORITHM DESCRIPTION

The AES algorithm [1] is a symmetric block cipher that can encrypt and decrypt data or information. Encryption converts data to an unintelligible format called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plain-text.

Algorithm is a byte (a sequence of eight bits), so the input bit sequence is first transformed into sub bytes using a GaloisFields (GF) algorithm [2], and then it is stored in a state array matrix. In the next step, the sub bytes information will undergo circular left shift operation and stored in state array. Finally mixing column operation is performed using a standard matrix. After the completion of each round,key should be added then only whole data will be encrypted. The input and output for the AES algorithm consists of sequences of 128 bits. The Cipher Key [3] for the AES algorithm is a sequence of 128, 192 or 256 bits.The basic AES Structure can be shown in Fig. 1

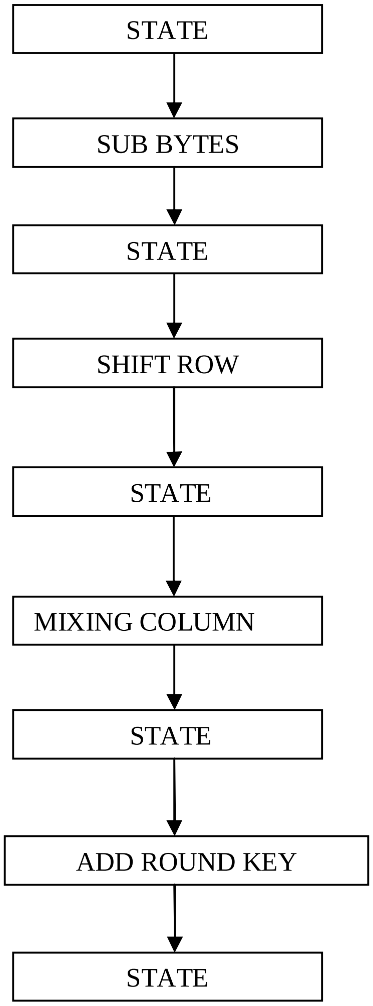


Fig 1.Basic AES Process

A. Advanced Encryption Standard - Encryption

The AES algorithm [4] operates on a 128-bit block of data and executed Nr – 1(n>9) loop times. A loop is called a round. The number of rounds of a loop can be 10, 12, or 14 depending on the key length. The key length is 128, 192 or 256 bits in length respectively. In generally first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and there will be no MixCoulmns transformation is performed in the last round.Basic EncryptionStructure can be shown in Fig. 2

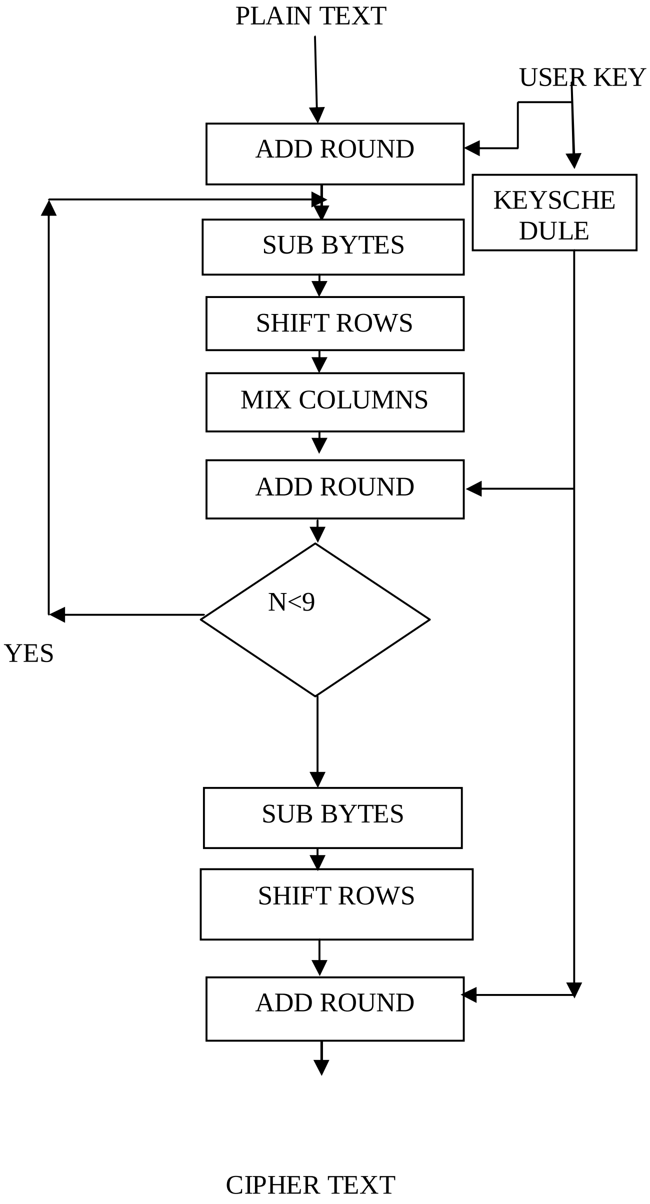


Fig 2.Basic Encryption Process

(a) Sub Bytes Transformation:

In generally, The Sub Bytes transformation is a byte substitution based on non-linearity principle. It operates on each of the state bytes independently. The Sub Bytes transformation is done using a Substitution table calculated from GaloisFields (GF) called S-box [4]. That S-box table contains 256 numbers (from 0 to 255).This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation.

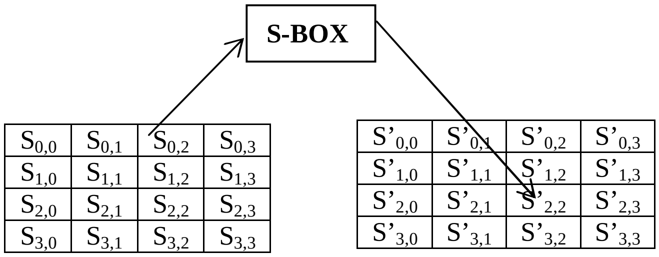

S-BOX
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig 3.Subbytes Transformation

This approach avoids complexity of hardware implementation and has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency.

(b)Shift Rows Transformation:

In the next step, the sub bytes information will undergo circular left shift operation[5] and stored in state array. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

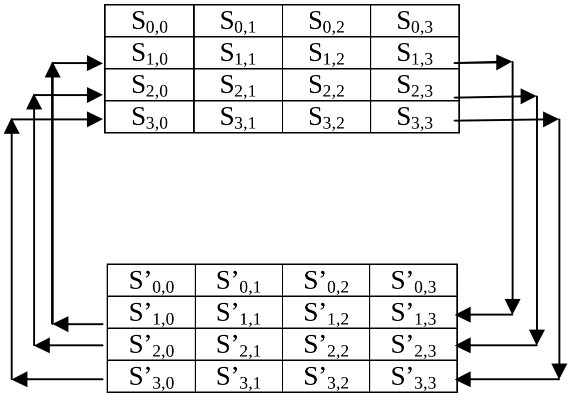

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig 4. Shift Rows Transformation

(c)Mix Columns Transformation:

Finally mixing column operation is performed using a standard matrix [5]. The columns of the state are considered as polynomials over GF (2^8) and multiplied by modulo x_4 + 1 with a fixed polynomial c(x), given by: c(x) = {03} x3 + {01} x2 + {01} x + {02}.

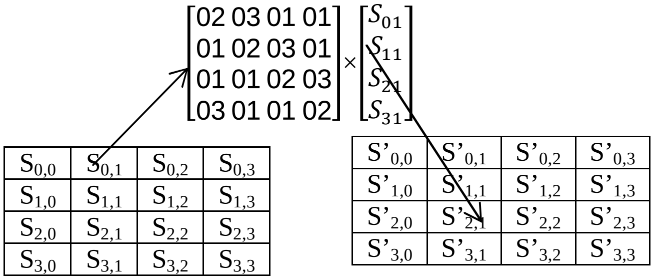

$\begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \times \begin{bmatrix} S_{01} \\ S_{11} \\ S_{21} \\ S_{31} \end{bmatrix}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig 5.MixColumns Transformation

(d)AddRoundKey Transformation:

In the AddRoundKey transformation [5], a Round Key is added to the stateafter the completion of each round, key should be added then only whole data will be encrypted. The operation of the Mix Columns transformation - by a simple bitwise XOR operation. By using the key expansion algorithm we can derive the round key. The encryption algorithm needs eleven 128-bitRound Key, which are denoted Round Key [0] Round Key [10] (the firstRoundKey [0] is referred as the main key).

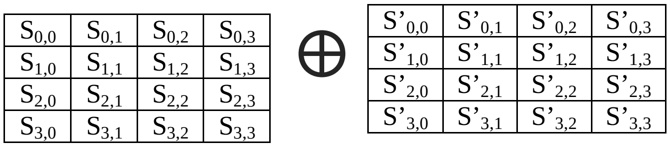

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$\oplus$
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Fig 6.AddRoundKeyTransformation

ENCRYPTION-SEQUENTIAL DESIGN

The AES sequential algorithm[8], [9] encryption flow is presented here Fig.7

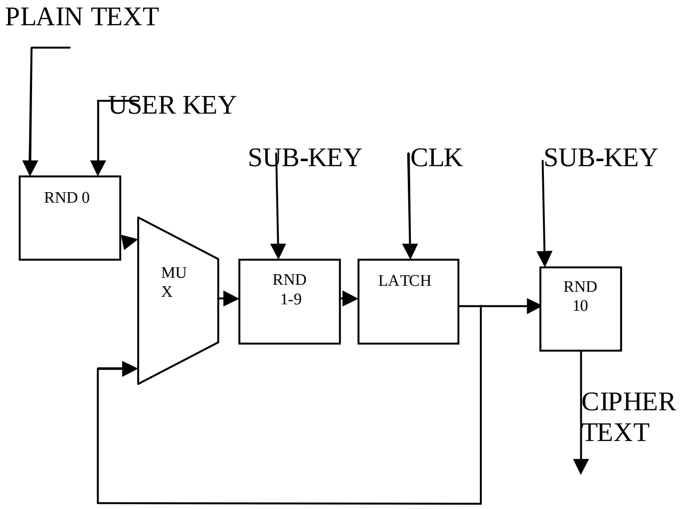


Fig 7.Sequential Encryption Process

All the four AES transformation rounds BS, SR, MC, and ARKare executed in a sequential fashion.After completion of these 9 rounds a final round is implemented separately.Throughput of the sequential design is reduced since a cipher block is produced every 10 cycles.On the other hand,it occupies small area as it can be implemented in low density Field Programmable Gate Array (FPGA) s. In sequential encryption [8] process the system complete the whole process at 200 MHz clock rate and the minimum time period to complete the whole process is 4.917ns. By utilizing the area of 935(LUTS) and 40,960 Megabytes memory, sequential encryption [13] process achieves throughput of 1188Mbps for encryption.

ENCRYPTION-PIPELINE DESIGN

In this section, we evaluate pipeline [10],[11] designs for AES algorithm against previous iterative looping(sequential design).The first design does not having the advantage of pipeline and strictly use iterative looping approaching contrast, to second design which uses pipeline buffers between the 10 AES stages. Because of applying pipelining design the resulting speed in terms of throughput rate and implementation area is evaluated and compared with the iterative looping design implementation. Inpipeline encryption process, the system completes the whole process at 285 MHz clock rate and the minimum time period [12] to complete the whole process is 3.51ns. By utilizing the area [13] of 1777(LUTS) and 55,535Megabytes memory, pipeline Encryption process achieves throughput of 3646.22Mbps for encryption. Hence pipeline encryption process provides high data rate [14], [15] of 2245 Mbps at the cost of more area (1777 LUTS).

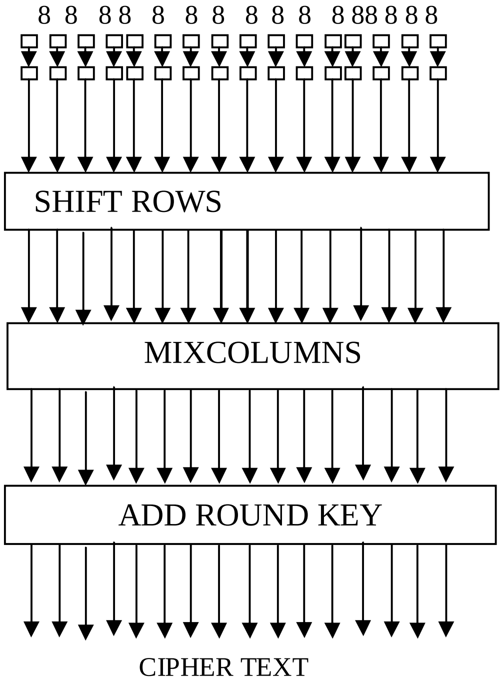
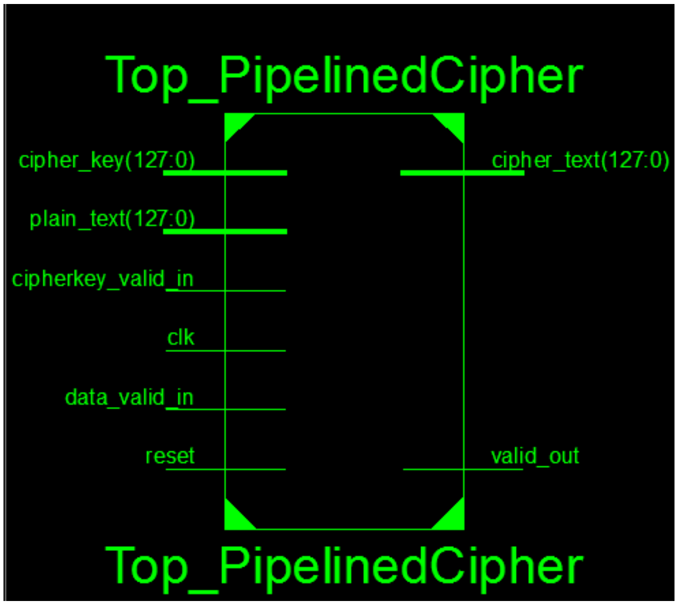


Fig 8.Encryption-Pipeline Design

FPGA IMPLEMENTATION OF AES ALGORITHM

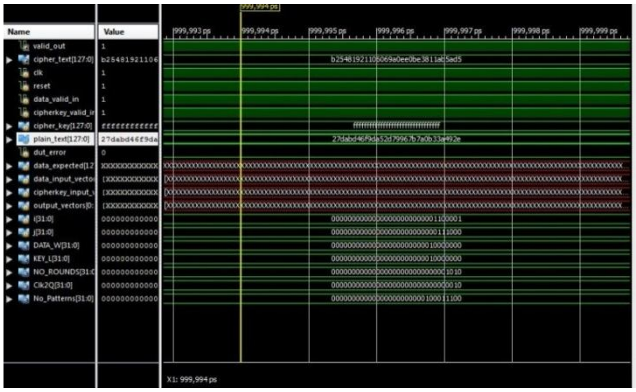
The detailed design of AES core based on FPGA implementation is shown below.



The entire design consists of 390 pins. It requires the text_in, text_out and key which have a 128 bits length. And the controls signals are used to control the proper operations of the core areclk, reset_n, and write, direction, done and enable pins.

SIMULATION RESULTS

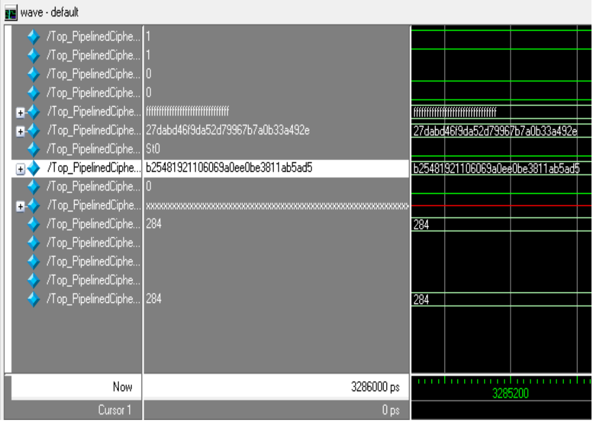
The design has been coded by VERILOG HDL. All the results are synthesized and simulated basing on the XILINX 14.2 ISIM simulator –on SPARTAN 3E device. The results of simulating the encryption and decryption algorithm from the simulator are shown below.The results of simulating the encryption algorithm from the simulator are shown in Fig.9



Name	Value
valid_out	1
cipher_text[127:0]	b25481921106
clk	1
reset	1
data_valid_in	1
cipherkey_valid_ir	1
cipher_key[127:0]	ffffffffffff
plain_text[127:0]	27dabd46f9da
dut_error	0
data_expected[12	xxxxxxxxxxxx
data_input_vecto	[xxxxxxxxxxx
cipherkey_input_v	[xxxxxxxxxxx
output_vectors[0:	[xxxxxxxxxxx
i[31:0]	000000000000
j[31:0]	000000000000
DATA_W[31:0]	000000000000
KEY_L[31:0]	000000000000
NO_ROUNDS[31:0	000000000000
Clk2Q[31:0]	000000000000
No_Patterns[31:0]	000000000000

Fig9.Encryption Simulation Results

The results of simulating the decryption algorithm from the simulator are shown in Fig.10

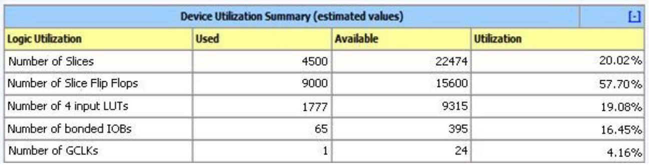


wave - default

Name	Value
/Top_PipelinedCiphe...	1
/Top_PipelinedCiphe...	1
/Top_PipelinedCiphe...	0
/Top_PipelinedCiphe...	0
/Top_PipelinedCiphe...	ffffffffffffffffffffffffffffffff
/Top_PipelinedCiphe...	27dabd46f9da52d79967b7a0b33a492e
/Top_PipelinedCiphe...	St0
/Top_PipelinedCiphe...	b25481921106069a0ee0be3811ab5ad5
/Top_PipelinedCiphe...	0
/Top_PipelinedCiphe...	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
/Top_PipelinedCiphe...	284
/Top_PipelinedCiphe...	
/Top_PipelinedCiphe...	
/Top_PipelinedCiphe...	
/Top_PipelinedCiphe...	284
Now	3286000 ps
Cursor 1	0 ps

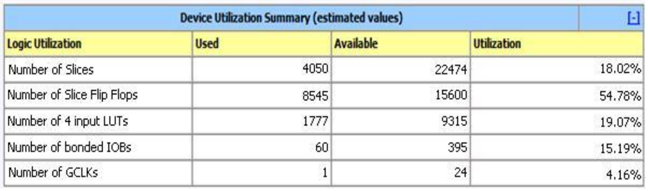
ffffffffffffffffffffffffffffffff
27dabd46f9da52d79967b7a0b33a492e
b25481921106069a0ee0be3811ab5ad5
284
284
3285200

Fig10.Decryption Simulation Results



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	4500	22474	20.02%
Number of Slice Flip Flops	9000	15600	57.70%
Number of 4 input LUTs	1777	9315	19.08%
Number of bonded IOBs	65	395	16.45%
Number of GCLKs	1	24	4.16%

Fig11.LUT's required for Encryption



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	4050	22474	18.02%
Number of Slice Flip Flops	8545	15600	54.78%
Number of 4 input LUTs	1777	9315	19.07%
Number of bonded IOBs	60	395	15.19%
Number of GCLKs	1	24	4.16%

Fig12.LUT's required for Decryption

COMPARISIONS

In this section, the results obtained by our design, and comparison between our results and other equivalent implementations is given and discussed.

Our design is found to be more efficient [14], [15] in terms of, throughput, area and memory. Therefore it allows us to process data in communication applications requiring a high security communication with high throughput and small area.

The design is compared with another implementation such as iterative looping, pipelining of individual 8-bits. Which uses the similar architecture with our design, but it provides high throughput& occupies less area.

PARAMETER	ITERATIVE LOOPING	PIPELINING(ENCRYPTIO N)	PIPELINING(DECRY PTION)
CLOCKRATE(MHz)	200	285	265.24
AREA(LUT'S)	935	1777	1777
MEMORY	40,960	31,316	31,316
THROUGHPU T(Mbps)	1188	3646.72	3395.22

Fig13.comparison between different encryption approaches on AES algorithm.

The design is tested with the sample vectors provided by FIPS 197. The algorithm achieves the throughput of 3646Mbit/sec for encryption and 3395.22 for decryption**.**

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, and 256 bits. The design is implemented on XILINX SPARTAN 3E FPGA. The proposed pipelining design is implemented on Cryptographic algorithms (Advanced Encryption Standard) for transferring the data with high data rate using minimum area and memory.

REFERENCES

[1]Advanced encryption standard (AES),nov.26,2001

[2]FIPS 197, "Advanced Encryption Standard (AES)",November 26,2001.

[3]A.Elbirt"reconfigurable computing for symmetric key algorithms",ph.dthesis,department of electrical engineering Worecester polytechnic institute,2002

[4]N.;Hasan,R.;Jubadi,W.M;"Design of AES S-BOX using combinational logic optimization",IEEE Symposium on Industrial Electronics & Applications

[5]Akashi satish,andsumiomorioka "an optimized s-box circuit architecture for low power AES design" IBM Research.CHES 2002;(2523);172-186.

[6]Kris gaj,andpawelchodowiec "hardware performance of the AES finalist's survey and analysis of results", George mason university. Proc. 3rd Advance Encryption standard (AES) candidate conference,newyork,2000:1-5

[7]Akashi satish, and sumiomorioka "unified hardware architecture for 128-bits block cipher AES" CHES 2003 ;(2779):304-318.

[8]N.sklavos,O.koufopavlou "architecture and vlsi implementation of the AES proposed rijndael" IEEE transactions on computers,vol.51,issue 12,pp.1454-1459,2002.Ahmad,

[9]WolformDrescher and Gerhard "VLSI architectures for non sequential inversion using Ecludian algorithm" SFB 358

[10]Mr.AtulM.Borkar,Dr.R.V.Kshirsagar and Mrs.M.v.vyawahare," FPGA Implementation of AES Algorithm", International Conference on Electronics Computer Technology (ICECT), pp.401-450,2011 3rd.\

[11]Hoang Trang,Nguyen Van Loi "An efficient FPGA implementation of the advanced Encryption Standard algorithm"©2012 IEEE.

[12]Pawelchowic, and krisgaj "very compact FPGA implementation of the AES algorithm", George mason university CHES;(2779):319-333.

[13]DaemenJ.,andRijmenV,"The Design of Rijndael: AES-the Advanced Encryption Standard",springer-verlag.2002

[14]Alex Panto,MarceloBarcelos,RicardoReis,"An IP of an Advanced Encryption Standard for Altera Devices",SBCCI 2002,197-202,porto Alegre,Brazil,9 and 14 September 2002.

[15]Amos R.omondi "micro architecture of pipelined and super scalar computers"Kluwer academic publishers 1999.