ISSN: 2349-7815
International Journal of Recent Research in Electrical and Electronics Engineering (IJRREEE)
Vol. 1, Issue 2, pp: (21-27), Month: July 2014 - September 2014, Available at: www.paperpublications.org

Design and Implementation of Area Efficiency AES Algoritham with FPGA and ASIC

[1]Y. Navya Rao, [2]D.Ranjith

[1] Y. NavyaRao, Research Scholar, ECE Department, Vaagdevi College of Engineering,Telanga, India
[2] D. Ranjith, Asst Professor, ECE Department, Vaagdevi College of Engineering,Telanga, India

Abstract: **A public domain encryption standard is subject to continuous, vigilant, expert cryptanalysis. AES is a symmetric encryption algorithm processing data in block of 128 bits. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. To implement AES Rijndael algorithm on FPGA using Verilog and synthesis using Xilinx, Plain text of 128 bit data is considered for encryption using Rijndael algorithm utilizing key. This encryption method is versatile used for military applications. The same key is used for decryption to recover the original 128 bit plain text. For high speed applications, the Non LUT based implementation of AES S-box and inverse S-box is preferred. Development of physical design of AES-128 bit is done using cadence SoC encounter. Performance evaluation of the physical design with respect to area, power, and time has been done. The core consumes 10.11 mW of power for the core area of 330100.742 µm2.**

Keywords: **Encryption, Decryption Rijndael algorithm, FPGA implementation, Physical Design.**

I. INTRODUCTION

The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits.

The AES specification uses the same three key size alternatives but limits the block length to 128 bits. A number of AES parameters depend on the key length as shown in Table 1

Table 1 AES parameters

Key Size	4/16/128	6/24/192	8/32/256
(Words/Bytes/Bits)			
Plaintext Block Size	4/16/128	4/16/128	4/16/128
(Words/Bytes/Bits)			
Number Of Rounds	10	12	14
Round Key Size	4/16/128	4/16/128	4/16/128
(Words/Bytes/Bits)			
Expanded Key Size	44/176	52/208	60/240
(Words/Bytes)			

II. RELATEDWORK

An overview of crypt analysis research for the advanced encryption standard is explained by Alan Kaminsky, et al. Linear Cryptanalysis exploits approximate linear relationships that exist betweeninputs and outputs of a function block . Case of a block Cipher, linear combinations of plaintext patterns and linear combinations of Cipher text patterns are compared to linear combinations of key bits. Differential cryptanalysis exploits relationships that exist between differences in the input and output of a function block .The advantages of a software implementation include ease of use, ease of upgrade, portability, and flexibility. However, a software implementation offers only limited physical security,

ISSN: 2349-7815
International Journal of Recent Research in Electrical and Electronics Engineering (IJRREEE)
Vol. 1, Issue 2, pp: (21-27), Month: July 2014 - September 2014, Available at: www.paperpublications.org

especially with respect to key storage. Conversely, cryptographic algorithms (and their associated keys) that are implemented in hardware are, by nature, more physically secure, as they cannot easily be read or modified by an outside attacker. AES that was implemented in HDL, led to the use of a bottom-up design and test methodology Arithmetic operations, architectural requirements, scalability and cost are widely considered in. Shuenn-Shyang Wangetal proposes an efficient FPGA implementation of AES.

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. 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. W.Mc Loone et al. presented a FPGA encryptor design that utilizes look-up table to implement the entire AES Rijndael round function. This is a more efficient FPGA implementation of AES Rijndael algorithm. Every component of AES algorithm is optimally designed to reduce the critical path and chip area.

Zhang and Parhi [12] have suggested new implementation of multiplicative inverse in GF (2^4) for realizing S-box and inverse S-box using arithmetic in GF((2^4)^2) which is further decomposed into GF(((2^2)^2)^2). Zhang and Parhi have also described efficient MixColumns and Inverse MixColumns implementation and evaluated the complete architecture for realizing fully unrolled AES implementation on FPGAs using 'on the fly' pipelined key schedule. Liu and Parhi proposed pre-computation techniques using which some computation in the critical path is eliminated to further reduce the critical path of the composite field arithmetic based S-box.

Choosing the suitable value for the subfield reducing polynomial was investigated extensively by O'driscoll. through a method which computes all possible Cyclotomic Cosets over 2 of degree 4 in GF(2^8) he concluded that there were only three choices for a reducing polynomial in the subfield GF(2^4) Most implementations conventionally make use of the memory intensive look up table approach for Substitute Byte/Inverse Substitute Byte block implementations resulting in an unbreakable delay .

III. PROBLEM DEFINATION

As Implementation of AES- IP Core for 128 bit by employing a memory less combinational design for the generation of S-box and Inverse S-box. As an alternative to achieve higher speeds by eliminating memory access delay while reducing overall area occupied and power consumed by the AES core.

IV. DESIGN METHODOLOGY OF AES ALGORITHM

The two basic hardware design methodologies currently available are Language-Based (High-Level) design and Schematic Based (Low-Level) design. Language-Based design relies upon synthesis tools to implement the desired hardware. While synthesis tools continue to improve, they rarely achieve the most optimized implementation in terms of both area and speed when compared to a schematic implementation. As a result, synthesized designs tend to be (slightly) larger and slower than their schematic based counterparts. Additionally, implementation results can greatly vary depending on the synthesis tool as well as the design being synthesized.

4.1 Block Diagram- Encryption

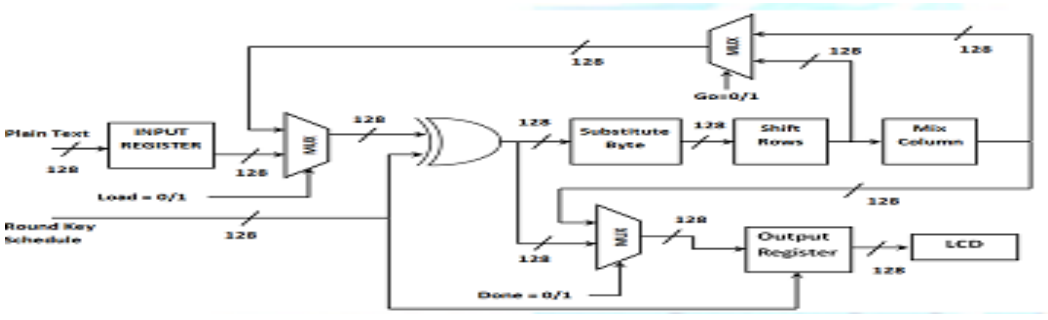

Figure 4.1 AES Decryption Block Diagram

ISSN: 2349-7815
International Journal of Recent Research in Electrical and Electronics Engineering (IJRREEE)
Vol. 1, Issue 2, pp: (21-27), Month: July 2014 - September 2014, Available at: www.paperpublications.org

The encryption component processes plain text – 128 bit, round key schedule and produces the encrypted cipher text – 128 bit depending on the control signals as follows, i.e., In Encryption mode of operation load will be forced to 1 to load the plain text - 128 bit and key schedule – 128 bit. Reset will be made 1, after loading the data, load will be forced to zero to process the internal blocks, Subbytes, shift rows, and mix column after 10 rounds of process, done signal goes to high and encrypted output is available in the output register.

4.2 Block Diagram- Decryption

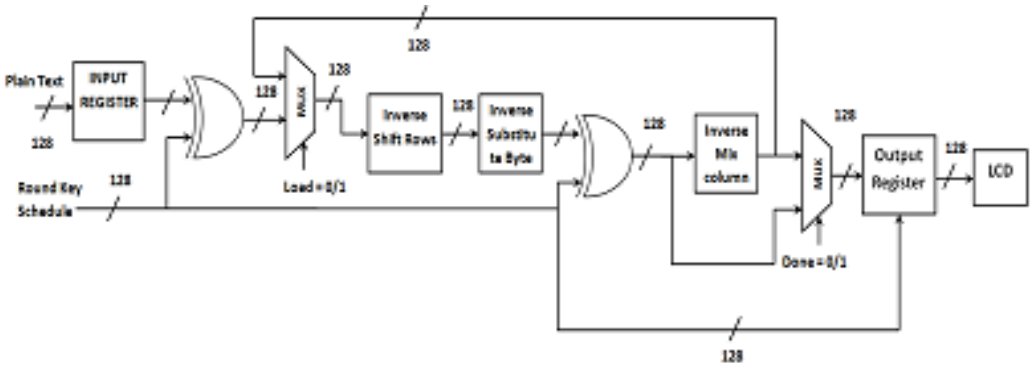

Figure 4.2 AES Decryption Block Diagram

The decryption component process encrypted data (cipher text – 128 bit) and encrypted key – 128 bit which has obtained in the 10th round operation of encryption and produces the decrypted output (plain text – 128 bit) and the key - 128 bit (which has been used in the initial round operation of encryption process). Depending upon the control signals, In decryption mode of operation load will be forced to 1 to load Encrypted input data 128 bit and key schedule 128 bit and reset is made as 1 after loading the inputs load is made to zero and internal blocks inverse shift rows, inverse substitute byte and inverse mix columns are processed after 10 round of operation, done signal will goes to high and decrypted output will be available in the output register.

V. VERILOG IMPLEMENTATION

The Verilog HDL is an IEEE standard hardware description language. It is widely used in the design of digital integrated circuits. Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis and for logic synthesis [18]. Verilog HDL allows designers to design at various levels of abstraction like register transfer level, gate level and switch level. Verilog is used as an input for synthesis programs which will generate a gate-level description for the circuit. Xilinx ISE 13.2 is a software tool developed by Xilinx for synthesis and analysis of HDL designs.

5.1 Device Utilization Summary of AES Encryption

Slice Logic Distribution				
Number of LUT-Flop pairs	711	69,120	1%	
IO Utilization				
Number of bonded IOBs	388	440	88%	
Other Utilization				
Number of BlockRAM/FIFO	9	128	7%	
Number using BlockRAM only	9			
Total Memory used (KB)	306	4,608	6%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			
Total equivalent gate count for design	1,121,593			
Additional JTAG gate count for IOBs	1,940			

Shows the Device Utilization Summary of AES Encryption it comprises of Add Round Key schedule Substitute Byte, Mix Column.

ISSN: 2349-7815
International Journal of Recent Research in Electrical and Electronics Engineering (IJRREEE)
Vol. 1, Issue 2, pp: (21-27), Month: July 2014 - September 2014, Available at: www.paperpublications.org

5.2 Device Utilization Summary of AES Decryption

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	402	69120	0%
Number of Slice LUTs	577	69120	0%
Number of LUT-Flop pairs	577	69120	0%
Number of bonded IOBs	388	440	88%
Number of Block RAM/FIFO	17	128	13%
Number of BUFG/BUFGCTRLs	1	32	3%

Above Shows the Device Utilization Summary of AES Decryption it comprises of Add Round Key Schedule Inverse-Substitute Byte,Inverse-Mix Column.

VI. PHYSICAL DESIGN OF AES 128BIT

The first step in the physical design flow is floorplanning. Followed by Partitioning, which is a process of dividing the chip into small blocks. After partitioning, Placement is performed in four optimization phases:

1. Pre-Placement optimization

2. In placement optimization

3. Post placement optimization (PPO) before clock tree synthesis (CTS)

4. PPO after CTS.

The goal of clock tree synthesis (CTS) is to minimize skew and insertion delay. After Routing Physical verification checks the correctness of the layout design. Once the design has been physically verified, opticallithography masks are generated for manufacturing. The layout is represented in the GDSII stream format that is sent to a semiconductor fabrication plant.

Throughput calculation:

Maximum output required time after clock (T) = Frequency (F) =1/T =1/4.496ns = 222.41 MHz. Throughput for 128 bit AES =128*F/10 = 2.84 Gbps.

VII. COMPARISON RESULTS

Table 5: FPGA Comparison Table for Frequency and Throughput

Design	Device	Fmax(Mhz)	throughput	Pslices
Elbirtel al	XCV1000-4	31.8	0.0470	10992
Monica liberatori et al	XC3S500E-4FG320	9.375	0.1202	1643
ours	XC3S500E-4FG320	222.41	2.846	402

ISSN: 2349-7815
International Journal of Recent Research in Electrical and Electronics Engineering (IJRREEE)
Vol. 1, Issue 2, pp: (21-27), Month: July 2014 - September 2014, Available at: www.paperpublications.org

7.1 Simulation Results

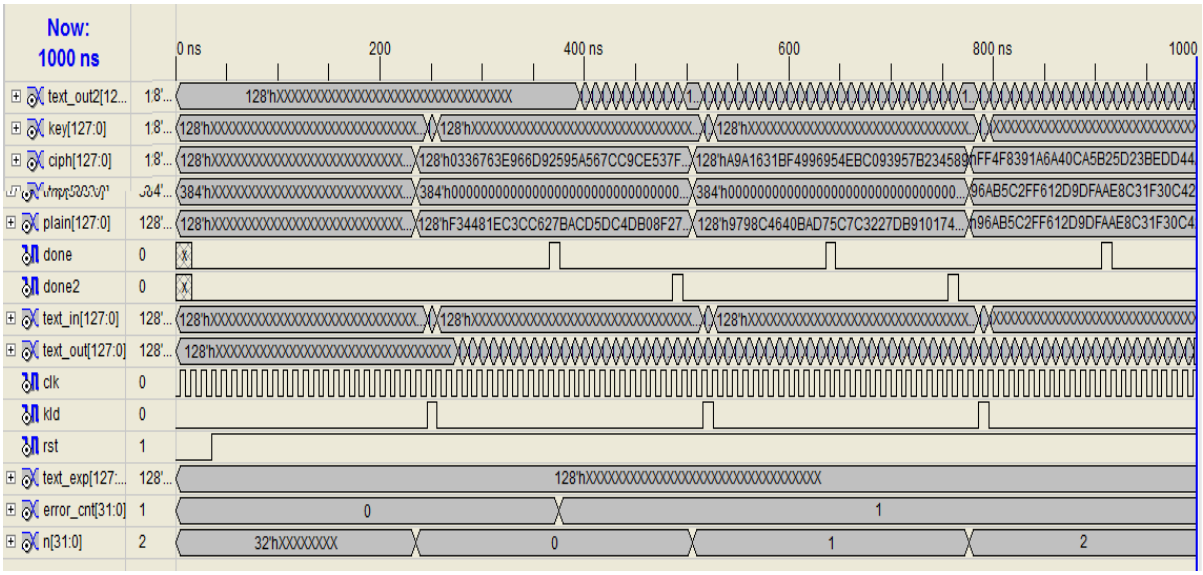

Fig 7.1 –Simulation of both encrypted and decrypted data

7.2 Implementation using FPGA

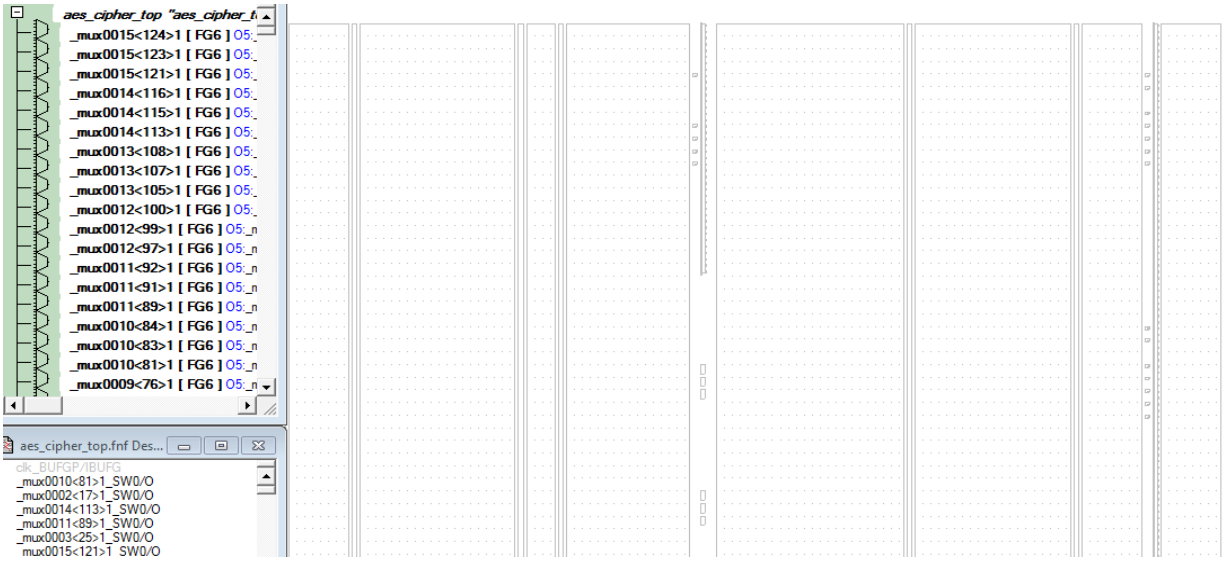

Fig 7.2 –Figure shows the FPGA structure

VIII. CONCLUSION AND FUTURE WORK

The designed core supports both encryption and decryption standards. Its functionality has been verified using simulation, by taking various inputs and is synthesized by using Xilinx 13.2.The design is targeted on FPGA (spartan-3E).The design uses 406 slices, 711 input look up tables and operates at 2.84 Gbps (Throughput).

Development of physical design of AES-128 bit is done using cadence SoC encounter. Performance evaluation of the physical design with respect to area, power, and time has been done. The core consumes 10.21 mW of power for the core area of 332128.742 µm2.

REFERENCES

[1] Alan Kaminsky, Michael Kurdziel, StanisławRadziszowski, "An overview of cryptanalysis research for the advanced encryption standard", IEEE, the 2010 military communications conference - unclassified program - cyber security and network management, pp. 1310, 2010.

[2] M.Matsui, "Linear cryptanalysis method for DES cipher", Eurocrupt, Lncs765, pp.386-397, Springer, 1994.

[3] I.Ben-Aroya. E.Biham, "Differential Cryptanalysis of Lucifer".crypto, journal of cryptology, pp.187-19l, Springer, 1994.

[4] R.Doud, "Hardware Crypto Solutions boost VPN," Electron. Eng. Times, pp. 57–64, April.12, 1999.

[5] C.Phillipsand K.Hodor, "Breaking the 10 k FPGA barrier calls for an ASIC-like design style," Integrated Syst. Design, 1996.

[6] M.Riaz and H. Heys, "The FPGA implementation of RC6 and cast-256 encryption algorithms," in Proc. IEEE 1999 CAN. Conf. electrical and computer engineering, Edmonton, Alta., Canada, march.1999.

[7] Elbirt, "An FPGA implementation and performance evaluation of the cast-256 block cipher," cryptography and information security group, ECE department, Worcester polytechnic institute, Worcester, Ma, Tech. Rep., may 1999.

[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] J.Daemenand V.Rijmen, "AES submission document on Rijndael", version 2, September 1999.(http:// csrc.nist.gov /cryptotoolkit/ AES/Rijndael/Rij ndael.pdf).

[10] F.X.Standaert, "AMethodology 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.

[11] W.Mclooneand J.V.Mccanny, "Rijndael FPGA implementation utilizing look-up tables signal processing systems", 2001 IEEE workshop on signal processing systems, pp.349-360, 2001. IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 3, June-July, 2013

[12] X.Zhang and K. K. Parhi, "High speed VLSI architectures for AES algorithm", IEEE transactions on VLSI systems, vol.12, no. 9, pp 957-967, 2004.

[13] M. M. WONG, M.L.D. Wong, "A high throughput low power compact AES s-box implementation using composite field arithmetic and algebraic form representation", proc. IEEE 2nd Aseasymposium on quality electronic design, pp 318-323, 2010.

[14] R. Liu, K.K.Parhi "Fast Composite field architectures for advanced encryption standard" proceedings GLSVLSI'08, Orlando, Florida, USA,pp.65-70, may 4–6, 2008.

[15] Menezes, P. Van Oorschot, and S. Vanstone, "Handbook of applied cryptography", CRC press, New York, 1997, pp. 81-83.

[16] Rudra, P.K. Dubey, C.S. Jutla, V.Kumar, J.R. Rao, and P. Rohatgi. "Efficient Implementation of RijndaelEncryption with composite Field Arithmetic."In proceedings of the cryptographic hardware andembedded systems conference, lecture notes in computer science vol 2162, pp.171-185, Paris, France, may 2001.

[17] C.O'Driscoll. "Hardware Implementation aspects of the Rijndael block cipher". mastersthesis. National University of Ireland, cork, Ireland 2001.

[18] Douglas JSmith, "HDL chip design using VHDL or Verilog", Doonepublications, 1996.

[19] Pong P. Chu, "FPGA prototyping by Verilog examples", Wiley publications, 2008.

[20] Sumanth Kumar Reddy S, R.Sakthivel, P Praneeth, "International Journal Of Advanced Engineering Sciences And Technologies" (IJAEST) Vol No. 6, Issue No. 1, pp.022 – 026, 2011.

[21] Nalini C, Dr.Anandmohan P V, Poonaih D.Vand V.D Kulkarni, "Compact Designs of SubBytes and MixColumn for AES" IEEE International Advance Computing Conference (IACC), pp.1241- 1247, March 2009.

[22] Monicalibertori, Fernando Otero, J.C.Bonadero, Jorge Castineira, "AES-128 Cipher High Speed, Low Cost FPGA Implementation" IEEE, pp.195-198, April 2007