Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

An Efficient FPGA Implementation of AES Algorithm

Avantika V. Patil[1]
Dept. of Electronics and Telecommunication
Trinity College of Engineering and Research
Pune, India

Prof. Rajendra A. Pagare[2]
Dept. of Electronics and Telecommunication
Trinity College of Engineering and Research
Pune, India

***Abstract*— In the paper The Advanced security system was enforced with reconfigurable Hardware. Here Field Programmable Gate Arrays (FPGAs) provide a additional speed than existing implementations. This analysis investigates the AES algorithmic rule with reference to FPGA and also the terribly High Speed microcircuit Hardware Description language (VHDL). Here system is design so that we can use same AES architecture for both text and digital files. In Spartan3 EDK we implemented the AES algorithm with the soft core processor Micro Blaze which is used for developing a Hardware structure which is configured using System C coding.**

***Keywords*— *Advanced Encryption Standard, VHDL,FPGA*.**

I. INTRODUCTION

AES is the Advanced Encryption Standard, an US government standard for encrypting and decrypting text. This standard is delineated in Federal information science normal (FIPS) 1971.NIST wanted to "make alternatives that supply a best level of security", compared to the information encoding normal (DES), that grew susceptible to brute-force attacks as a result of its 56-bit effective key length[1][2]. An interchangeable block cipher that supported multiple key lengths was needed by AES candidates. The National Institute of Standards and Technology (NIST) is [3]published request for comments for the "Development of a Federal Information Processing Standard for AES , On January 2, 1997.

II. AES ALGORITHM

It is a symmetric block cipher with a block size of 16 bytes. Key lengths of AES can be 128 bits, 192 bits, or 256 bits. In 128,192,256 bit AES it used 10,12,14 round respectively.
The main loop of AES9 conducts the following functions:

- SubBytes()
- ShiftRows()
- MixColumns()
- AddRoundKey()

In AES subbyte,shiftrows and mix column rounds are the methods of "confusion" and "diffusion."Add round key function is used to main encryption. The concepts of confusion and diffusion whis is reported by Claude Shannon, in seminal 1949 paper, "Communication Theory of Secrecy Systems" ."Two ways recommend themselves for frustrating a applied mathematics analysis. These can be called the methods of diffusion and confusion.".Diffusion suggests that patterns within the plaintext area unit spread within the cipher text. Confusion suggests that the link between the plaintext and also the cipher text is obscured.
A simpler way to view the AES function order is:
1. change byte using s box (Sub Bytes).
2. shift each row (Shift Rows).
3. Scramble each column (Mix Columns).
4.xor operation of bytes(Add Round Key).
AES designs plaintext into 128-bit blocks, and serves each block as a 4x4 State array. After it performs four operations in each round. column and row information used in the operations were contained by arrays, that is Mix Columns() and Shift rows()[4].

A. SubBytes()

By processing each 8 bit through an S-Box Sub Bytes()adds confusion. The S-Box is substitution table, in which one 8bits are substituted for another, based on a substitution algorithm[5]. Let the AES Substitution is
| 0 1 2 3 4 5 6 7 8 9 a b c d e f g h
To carried out the S-Box operation on an example string of "ABC," take the hexadecimal value of each byte. ASCII "A" == hex 0x43, "B" == 0x44 and "C" == 0x42. Look the first (left) hex digit in the S-Box column and the second in the S-Box row. 0x43 becomes 0x1a,0x42 becomes 0x2c, and 0x44 becomes 0x1b.

B. ShiftRows()

Diffusion by changing data within rows is provided bt this. The zeroth row of the State is not shifted, first row is shifted 1 8 bits, second is shifted 2 8 bits, and third row is shifted 3 8 bits, as shown in the FIPS illustration as follows:

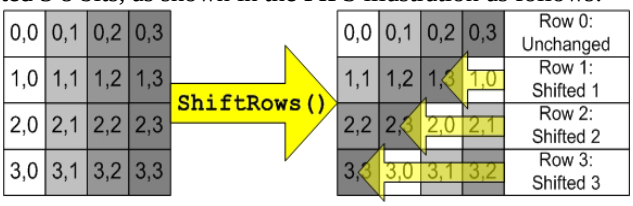

Fig. 1. ShiftRows

C. MixColumns()

Diffusion by changing data within columns is also provided by Mixcolumns(). The 4 8 bits of each column in the State are treated as a 4-8 bits number and transformed to another 4-8 bits number through finite field mathematics, as viewed in the FIPS picted as follows:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

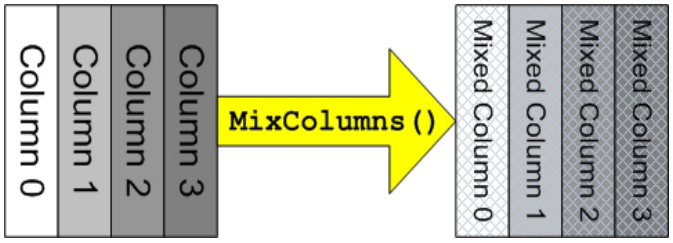

Fig. 2. Mix Columns

D. AddRoundKey()

In the Add Round Key() function actual 'encryption' is done, in which each byte in the State is XORed with the sub key. In key expansion shedule sub key is generated from the key, as shown in the FIPS picted as follows:

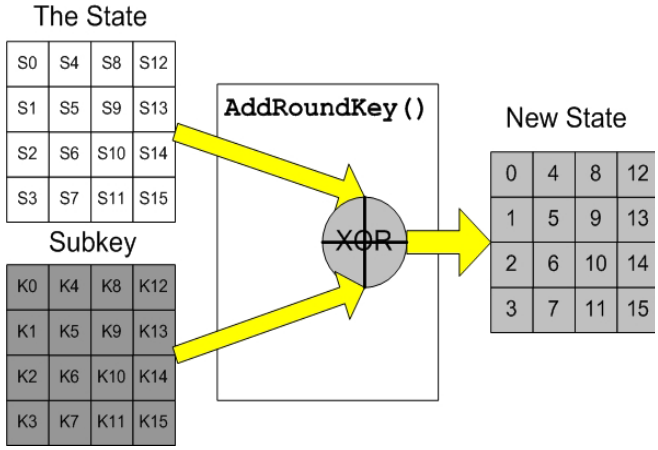

Fig. 3. Add RoundKey

One Round of AES

Fig. 4. In this round of AES encryption shown in the FIPS two dimensionally:

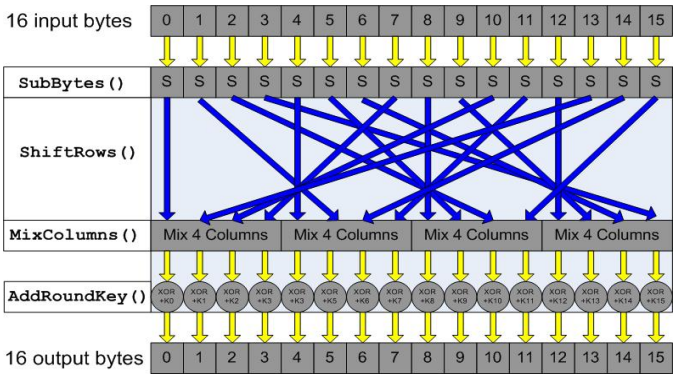

Fig. 5. One Round of AES

III. AES DECRYPTION

Through the process Add RoundKey(), plus the inverse AES functions InvShiftRows(),and InvMixColumns(), InvSubBytes() decryption occurs . Inverse function is not required by AddRoundKey() [6], because it simply XORs the state with the sub key (XOR encrypts when applied once, and decrypts when applied again).

A. Implementation

FPGA (field programmable gate array) which is mostly used to generate ASIC IC's to the computations.It offers more speed in execution process. So, to generate ASIC IC's FPGA's are mostly used.

TABLE I. CONFIGURATION OF FPGA

Name	Value
Family	Spartan 3
Device	XC3S200
Package	TQG144
Speed Grade	-4

B. Xilinx Platform Studio

It is the development impression or it is a user interface used for planning the hardware portion of your embedded processor system. EDK Xilinx Embedded Development Kit (EDK) is correlate integrated system tool set for developing systems with Xilinx Micro Blaze and PowerPC CPUs. EDK tools includes applications to help the designer to develop relate embedded hardware system right from the hardware creation to final implementation of the system on an FPGA. hardware and software system parts of the embedded processor system is generated by this system and also create of a verification element is elective. This system also involves: hardware platform generation, hardware platform verification (simulation), software system platform generation, software system application generation, and software system verification. Base System Builder is that the wizard that's will not to mechanically create a hardware platform in assigning with the consumers specifications that's defined by the MHS (Microprocessor Hardware Specification) file[7]. The Microblaze hardware specification file defines the system design, peripherals and embedded processors. The Platform Generation tool generates the hardware platform mistreatment the MHS file as input. MSS file is used for defining a software system platform that defines driver and library customization parameters for internal and external devices, processor customization parameters, one hundred ten devices, interrupt handler routines, and different software system connected routines. The Microblaze software specification file is used as a input for the Library Generator tool for personalization of drivers, libraries and interrupts handlers.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

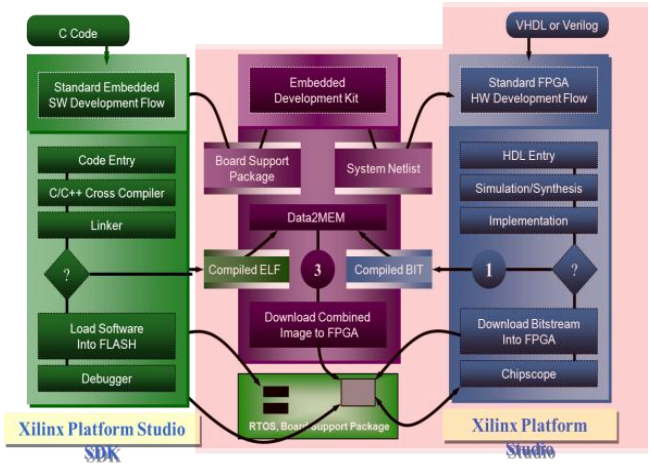

Fig 4. Design Flow of EDK

On the hardware platform the generation of the verification platform is facultative and is forecasted. To make simulation files for a particular machine the MHS file is taken as Associate in Nursing input by the tool 3varieties of simulation models will be created by the Siegen tool: behavioral, structural and temporal arrangement models. another helpful tools in the market is EDK Platform Studio that provides the GUI for making the MHS and MSS files[8]. Import IP Wizard that permits the generation of the designer's own peripheral and import them into EDK comes. Platform Generator customizes and creates the processor system within the sort of hardware net lists. Library Generator tool configures libraries, file systems,device drivers, and interrupt handlers for embedded processor system. Bit stream Initializes tool initializes the instruction memory of processors in the Filed programmable gate arry shown in figure2. antelope Compiler tools ar used for collecting and linking application executables for every processor within the system. There ar2choiceson the market for debugging the appliance generated victimization EDK namely: Xilinx micro chip correct (XMD) for debugging the appliance package employing a micro chip correct Module (MDM) within the embedded processor system[9], and package programme that invokes the package programme appreciate the compiler taking used for the processor. C. package Development Kit XPS package Development Kit (SDK) is Associate in Nursing integrated development atmosphere, complimentary to XPS, which is used for C/C++ embedded package application creation and verification. SDK is made on the Eclipse open source framework. SDK may be a suite of tools that allows you to style a package application for elite Soft IP Cores within the Xilinx EDK.The package application will be written during a "C or C++" building process then the entire embedded processor system for user application are completed, else correct download the bit file into FPGA[10]. Then Filed programable gate array behaves like processor implemented on it in a Xilinx Field Programmable Gate Array device.

IV. RESULTS

Hardware implementation was through system C coding and its results are as follows

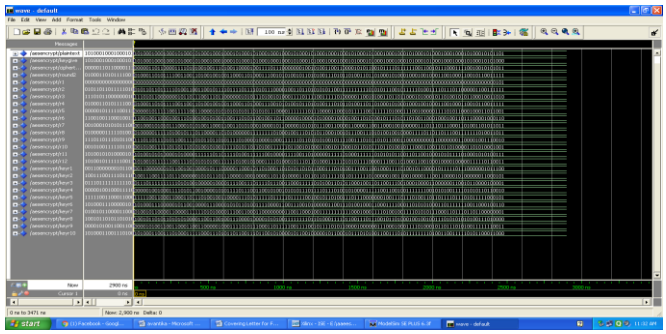

Fig 5. Simulation results in modelsim

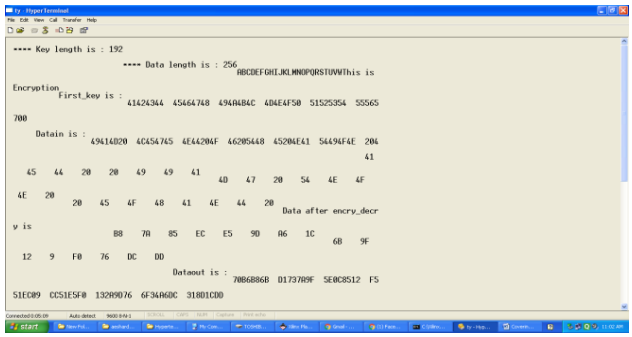

Fig 6. Encrypted text data process on FPGA and display on Hyperterminal

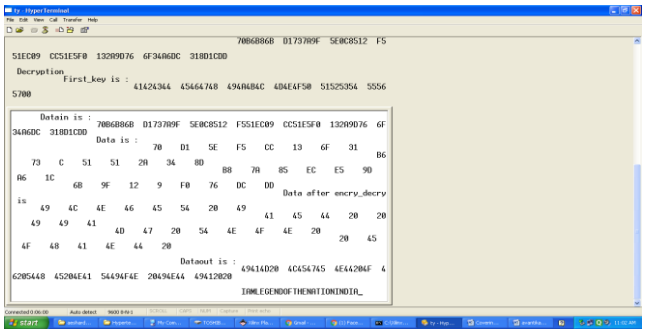

Fig 7. Decrypted text data process on FPGA and display on Hyperterminal

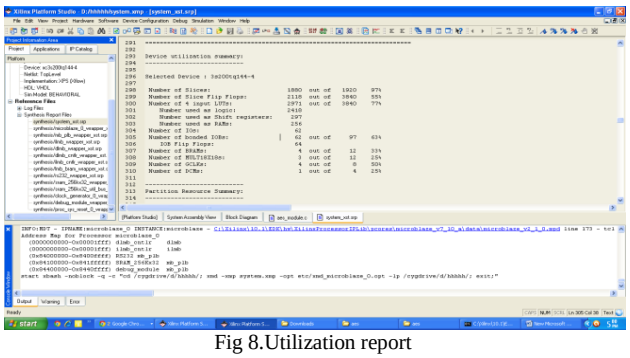

Fig 8.Utilization report

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

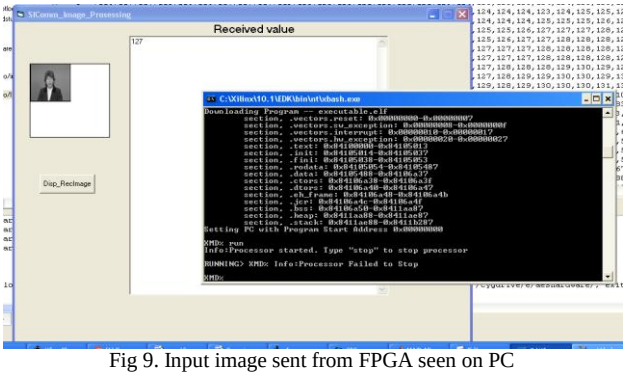

Fig 9. Input image sent from FPGA seen on PC

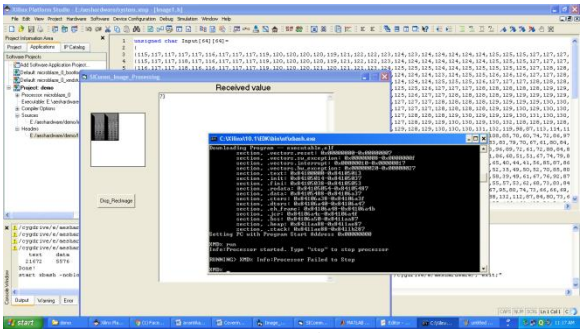

Fig 10.Encrypted image sent from FPGA seen on PC

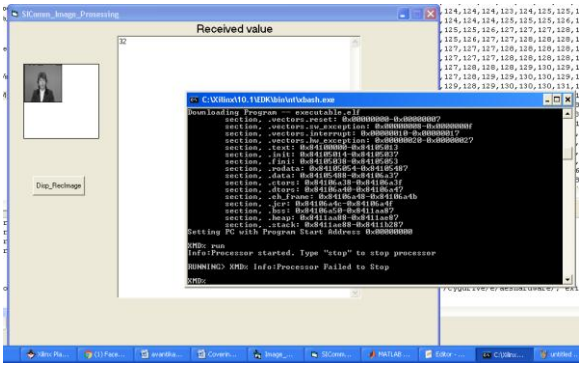

Fig 11. Decrypted image sent from FPGA seen on PC

V. CONCLUSION

Advanced Encryption algorithm for text and digital data is implemented using FPGA in my work. This system works on a period pipelined flow of the xilinx Micro Blaze architecture of Spartan3 EDK. On the another hand, synthesis results shows that space consumption is low and also permitting the implementation of this method over inexpensive FPGAs.

ACKNOWLEDGMENT

I am sincerely thankful to Prof. R.A.Pagare, ME co-ordinator Mrs. Handore, HoD Mr.V.S.Hendre for their precious guidance. I thank all others in the department without their help I could not have attained this hard success. Also I thank all those who were involved directly and indirectly

REFERENCES

[1] Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advanced Encryption Standard", Springer-Verlag, 2002

[2] FIPS 197, “Advanced Encryption Standard (AES)”, November 26, 2001

[3] J. Nechvatal, et. al., Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000,at [4]

[4] AES page available via http://www.nist.gov/CryptoToolkit.

[5] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, September 3, 1999, available at [1].

[6] Computer Security Objects Register (CSOR): http://csrc.nist.gov/csor/.

[7] G. Rouvroy, F. X. Standaert, J. J. Quisquater, J. D. Legat, Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications, Procedings of the international conference on Information Technology: Coding and Computing 2004 (ITCC 2004), pp. 583 – 587, Vol. 2, April 2004

[8] K. Chapman, PicoBlaze 8-bit Microcontroller, Xilinx, 2002 http://www.xilinx.com/products/design_resources/proc_central/grouping/picoblaze.html

[9] N. Pramstaller and J. Wolkerstorfer, A Universal and efficient AES co-processor for Field Programmable Logic Arrays, FPL 2004, LNCS Vol. 3203, pp. 565-574, SpringerVerlag, 2004

[10] P. Chodowiec, K. Gaj, Very Compact FPGA Implementation of the AES Algorithm, Cryptographic Hardware and Embedded Systems (CHES 2003), LNCS Vol. 2779, pp. 319 – 333, Springer-Verlag, October 2003