Implementation and Analysis of Optimized AES on FPGA

Nikita Purohit
Research Scholar
Department of Electronics Engineering
RCOEM, Nagpur

Meghana A. Hasamnis
Professor
Department of Electronics Engineering
RCOEM, Nagpur

ABSTRACT
In today's world of digital transmission and reception of data and images high performance processing hardware is required. This paper presents an optimized AES algorithm for both software and hardware implementation through which the execution speed of the process is improved by reducing the cycle count. Optimized AES is implemented using soft-core processor on FPGA Spartan-6 kit and the results are obtained using timing analyzer tool of Xilinx design suite 14.5. The execution time for hardware implementation of optimized AES code is improved by 12.46% and 11.58% for encryption and decryption module respectively. Target device used for implementation of design is XILINX 14.5 platform studio xc6slx45-2csg324.

Keywords
FPGA-Spartan 6, Cryptography, AES.

1. INTRODUCTION
Cryptography is the process of securing information through various techniques. Thus, provide information security [1]. It has three types of functions: hash key, secret key and public key [3]. Secret key which is also known as symmetric key is popularly used in most of the algorithm [4]. The symmetric key include DES, 3-DES, Blowfish, RC2 and AES algorithm for encryption and decryption of data. It has been approved by National Institute of Standards and Technology (NIST) that Rijndeal AES is better than other methods as this algorithm supports significantly larger key sizes [5], [6]. In terms of cryptography encryption is the process of converting input plaintext into unreadable text known as ciphertext which hides the original text and thus protects the information from attacker. While decryption is the process of regaining the original plaintext from ciphertext [7]. Thus main objective of cryptography is to provide authencity to sender and receiver, message privacy and non-repudiation and is widely used in the field of Local area networks, digital communication, WLANs. Thus, in digital transmission cryptography provides confidentiality, data integrity and authencity [2].

2. AES ALGORITHM
In this paper we use Advanced Encryption Algorithm i.e. AES where input data in the form of plaintext which is encrypted to obtain cipher text. The plaintext of 128 bit is stored in the form of 4 X 4 matrix known as state having 16 elements. If the input data exceeds, it is then divided into 128-bit block of data and then processed [8], [9] AES is n round process, where n depends on key length as described in Table.1.

Table 1. Number of AES rounds

No. of rounds (N_r)	Cipher key (bit)
10	128
12	192
14	256

AES is divided into four stages and multiple rounds are performed to these stages for encrypting input data to cipher text. The four stage are as follows

i. Substitute byte
ii. Shift rows
iii. Mix columns
iv. Add round key

The process begins with Add round key followed by other stages, thus remaining nine rounds of four stages and tenth round of three stages is performed. The flow chart of AES algorithm is shown in Fig.1 [10].

i. Substitute byte

In SubByte, simply a look-up table of 16 × 16 matrix known as s-box is used. The value of state is replaced by the corresponding s-box value.

ii. Shift row transformation

It is a simple stage where permutation is done. Each byte of the state is rotated row-wise.

iii. Mix column transformation

This stage is basically substitution but make use of arithmetic of GF (2^8). Each column is operated individually.

iv. Add round key transformation

In this stage state values are bitwise XORed with the key.

The above process is followed inversely for Decryption of the ciphertext. In this paper we have used 128 bit AES algorithm.

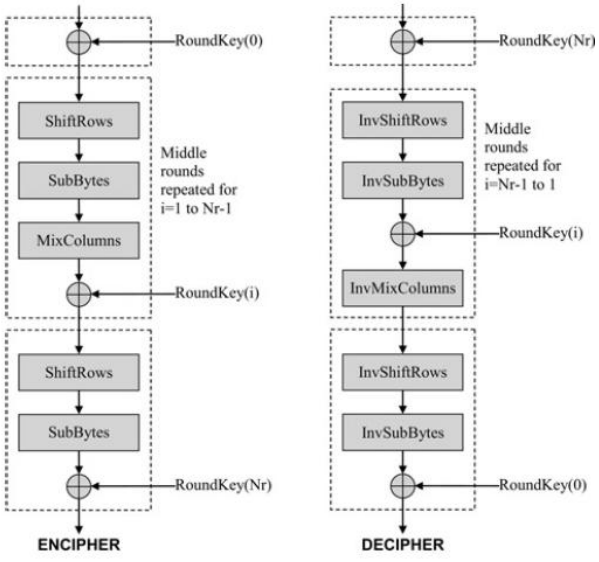

Fig 1: Flow chart for Encryption and Decryption [10]

3. DEVELOPMENT OF HARDWARE PLATFORM USING MICRO-BLAZE PROCESSOR

Softcore processor are implemented through FPGA, ASIC which can be customized according to the user requirement thus it provides better performance [11]. Different types of soft-core processor available are LEON 3, Microblaze, Nios II, OpenRISC.

Microblaze soft-core processor is a reduced instruction set computer (RISC) used for implementations of Xilinx FPGA [12]. It is highly configurable in terms of Cache size, pipelining, peripherals, bus interface, etc. Thus due to these customization one can make appropriate design for specific host hardware. Implementation of soft-core microblaze processor has been done on hardware[13].

Hardware platform for module is generated by embedded development kit (EDK), in the generated design interfaces like processor local bus (PLB), BRAM, RS232 UART, Digilent USB, controllers, XPS timer are included as shown in Fig.2.

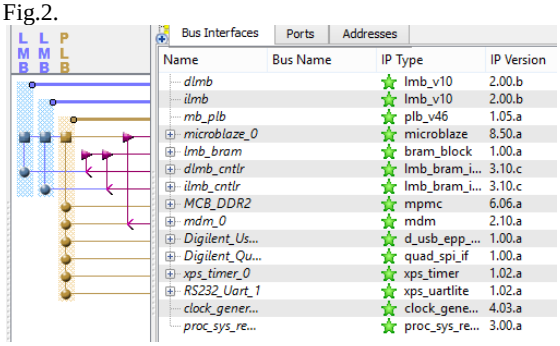

Fig 2: Peripherals of module generated on Hardware Platform of Microblaze

The block diagram of the design with all the peripherals and module including master microblaze processor mb_plb in which all the BRAM, data local memory bus(dlmb) controller and instruction local memory bus is included while in slave processor peripherals like xps timer, xps_UART, quad_SPI are included as shown in Fig.3

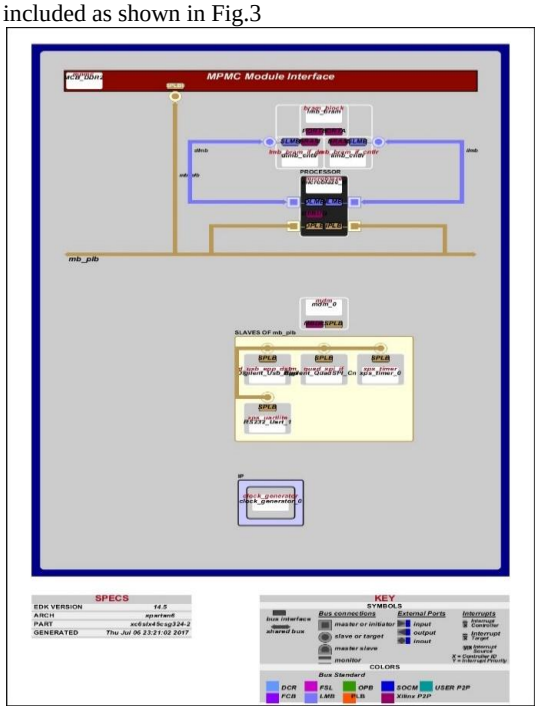

Fig 3: Generated Block diagram on Microblaze Processor

4. SOFTWARE IMPLEMENTATION OF AES ALGORITHM

AES algorithm is implemented on XILINX ISE design suite 14.5 platform studio xc6slx45-2csg324 Microblaze processor. Design is made by using EDK platform of microblaze processor and then exported and launched through SDK. In SDK platform the algorithm is written in High level language C, the code is then implemented on target device. The results for encryption and decryption is shown in fig.4 and fig.5 respectively.

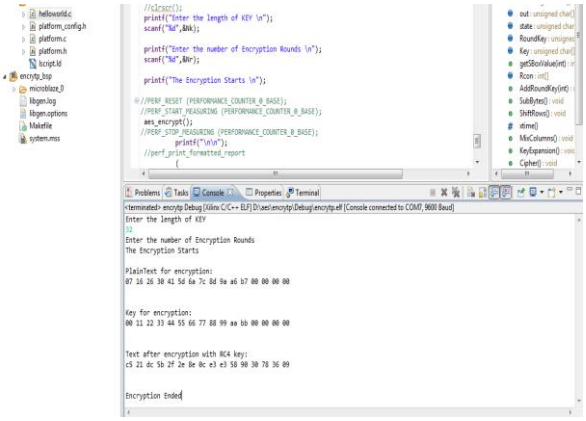

Fig 4: AES Encryption implementation using developed platform

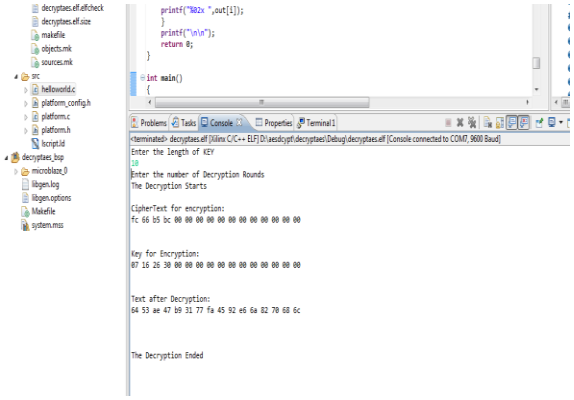

Fig 5: AES Decryption implementation using developed platform

5. OPTIMIZED AES ALGORITHM

5.1 Software Implementation

The speed of AES algorithm generated on developed platform is checked through cycle count [14]. Individual blocks of AES is analyzed by counting the cycle which implies that the two block out of four stage consumes more time, they are Subbyte and Add-round Key. The two blocks have two rolling loop which increases the overhead time. Which can be optimized by unrolling the loops thus decrease overhead time [15]. Thus the execution time of encryption and decryption is improved by decreasing the cycle count. The results for the AES and optimized AES software implementation is shown in table.2

Table.2. Software Implementation of AES algorithm

AES Implementation	AES (cycle count)	Optimized AES (cycle count)
AES Encryption	16735149	16734000
AES Decryption	16596580	16595980

5.2 Hardware Implementation

The software implemented C code for AES algorithm is converted into hardware descriptive language i.e into VHDL code and then implemented on Xilinx ISE design suite 14.5. The optimized AES C code is also converted into VHDL code and implemented on the same platform. The execution time for both the code is generated using Timing Analyzer tool of Xilinx. The results in terms of execution time for hardware implementation is shown in Table.3.

Table.3. Execution Time for AES algorithm

AES Implementation	AES (Execution time)	Optimized AES (Execution time)
Encryption	10.253ns	8.975ns
Decryption	12.096ns	10.695ns

6. RESULT AND CONCLUSION

It has been analyzed that after implementing the AES and optimized AES code on hardware as well the execution time have been decreased. The analysis is shown in table.2 and table.3 where all the results obtained after implementing design in software as well as hardware is mentioned. The simulation result as performed on Xilinx ISE 14.5 for implementing encryption and decryption module.

Graphical representation for Hardware implementation of AES and optimized AES is shown in Fig.6.

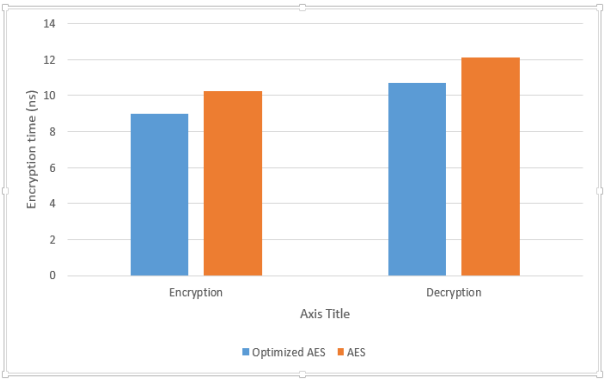

Fig.6. Graphical representation for Hardware implementation of AES.

Thus the execution time for hardware implementation optimized AES encryption is improved by 12.46% and for Decryption it is improved by 11.58%. Thus overall execution time is reduced for optimized AES**.**

7. REFERENCES

[1] G. Julius Caesar. Cryptography, Security Engineering: A Guide to Building Dependable Distributed System, Chapter 5.

[2] Tutorials Point www.tutorialspoint.com.

[3] Guru Ghasidas Vishwavidyalaya http://www.ggu.ac.in

[4] Sourabh Chandra and Smita Paira. A comparative study of symmetric and asymmetric key cryptography, ICECCE 2014.

[5] The Rijndeal algorithm T.Jamil, IEEE potential 2004 volume: 23.

[6] J.Daeman , V.Rijmen. the block cipher Rijndeal springer- verlag ,2002.

[7] Panda, Madhumita, and Atul Nag. "Plain Text Encryption Using AES, DES and SALSA20 by Java Based Bouncy Castle API on Windows and Linux."ICACCE, 2015.

[8] A.M.Deshpande, M.S.Deshpande and D.N.kayatanayar, "FPGA Implementation of AES encryption and decryption". IEEE inter conference, Vol.01, issue 04, pp.1-6, june 2009.

[9] N. S. SAI SRINIVAS and MD. AKRAMUDDIN. "FPGA Based Hardware Implementation of AES Rijndael Algorithm for Encryption and Decryption" International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016.

[10] T.Good, M.Benaissa. Pipelined AES on FPGA with support for feedback. IET Inf. Secur 2007.

[11] Franjo Plavec, "SOFT-CORE PROCESSOR DESIGN" Master of Applied Science Graduate Department of Electrical and Computer Engineering University of Toronto 2004.

[12] MicroBlaze Processor Reference Guide by Xilinx, 2008

[13] Parallel and Flexible Multiprocessor System-On-Chip for Adaptive Automotive Applications based on Xilinx MicroBlaze Soft-Cores by Michael Hübner, Katarina Paulsson, Jürgen Becker Universitaet Karlsruhe (TH), Germany.

[14] A Compact 8-bit AES Crypto-Processor F Haghighizadeh .H. Attarzadeh, M. Sharifkhani, Second International Conference on Computer and Network Technology,IEEE 2010.

[15] AES-CBC Software Execution OptimizationRazvi Doomun*, Jayramsingh Doma, Sundeep Tengur Computer Science and Engineering, University of Mauritius.