FPGA IMPLEMENTATION OF AN AES PROCESSOR

Kazi Shabbir Ahmed, Md. Liakot Ali, Mohammad Bozlul Karim and S.M. Tofayel Ahmad
Institute of Information and Communication Technology
Bangladesh University of Engineering and Technology, Bangladesh
E-mail: shabbir_528@yahoo.com,liakot@iict.buet.ac.bd, hira9505040@gmail.com and ntofayel@yahoo.com

ABSTRACT

Advanced Encryption Standard(AES) implementing in a faster and secured way is expected. AES can be implemented in software/hardware. In hardware implementation ASIC solution requires high cost and much design time while FPGA based implementation offers lower cost, quicker and more customizable solution. This paper represents implementing AES in FPGA with minimum latency and speedy throughput where Verilog HDL is used to simulate the operations. Both Encryption and Decryption are carried out and simulated in an iterative design approach to minimize the hardware consumption. Here the operation of AES like substitution bytes, mix column are simplified to reduce complexity and to produce a simple design which would deliver a high throughput and minimum latency with EP2C35F672C6 device from Altera provided Cyclone II family.

Key Words—AES, FPGA, Verilog HDL, cryptography, encryption, decryption.

1. INTRODUCTION

With the growth of information and communication technology, the processing of data and transferring the same through different media involves security [1]. A number of crypto algorithms have been developed [2-4]. Keeping pace with maturity of the security technology the hackers, the electronic eavesdroppers, virus and the electronic frauds have been coming into the field with new sophisticated techniques to attack the security mechanism. So to protect any unusual attack to the valuable information source and their transmission, the algorithm Advanced Encryption Standard(AES), a Federal Information Processing Standard (FIPS) is approved by National Institute of Standards and Technology(NIST)[4,7,8,11].But AES has 10(Ten) round of complex algebraic and matrix operation which involve high processing power and introduce delay in encryption and decryption process. For this reason at the start of this work the speed is treated as a major issue and concentration is given on hardware based implementation. There are also two types of hardware based implementation. FPGA (Field Programmable Gate Array) based implementation is chosen in this work as FPGA offers lower cost, flexibility and reasonable performance than ASIC (Application Specific Integrated Circuit) implementation. Previously researcher proposed implementation of AES processor on FPGA hardware dropping many security features since earlier version of the FPGA available in the market was low capacity. Now high capacity FPGA from different vendor is coming in the market. Recently design of an AES processor using VHDL and its implementation on Xillinx FPGA without sacrificing any security feature of the algorithm is reported [6]. Altera's FPGA is another famous FPGA to the customers. It offers a lot of high capacity FPGAs under different families. Literatures [10],[12],[13],[18],[21-23] describe design and implementation of AES processor in the FPGA platform where maximum throughput achieved is 21.54 Gbps with latency 71 clock cycle. However reduced latency is essential for developing real time applications. So a research work conducted to implement the AES processor on this FPGA to achieve minimum latency with suitable speed performance.

2. AES STRUCTURE

AES, also known as Rijndael, is a block cipher adopted as an encryption standard by the US government, which specifies an encryption algorithm [4-8]. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits sequence. In this paper 128 bits key is used for 128 bit data block. The input, output and cipher key bit sequences are processed as arrays of bytes that are formed by dividing these sequences into groups of eight contiguous bits to form arrays of bytes. The different transformations operate on the intermediate result, called the state, which is the intermediate cipher result. The state can be pictured as a rectangular array of bytes. This array has four rows; the number of columns is denoted by Nb and is equal to the

block length divided by 32. The cipher key is similarly pictured as a rectangular array with four rows. The number of columns of the cipher key is denoted by Nk and is equal to the key length divided by 32. The number of rounds is denoted by Nr and depends on the values Nb and Nk. It is given in Table 1.

Table 1: Number of rounds depending on key size and block length

	Key Length N_k	Block Size N_b	Number of Rounds N_k
AES 128	4	4	10
AES 192	6	4	12
AES 256	8	4	14

For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by *Nr*, where Nr = 10 when $N_k = 4$, Nr = 12 when $N_k = 6$, and $N_r = 14$ when $N_k = 8$. For both its cipher and inverse cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations:
1) Byte substitution using a substitution table (S-box), 2) Shifting rows of the state array by different offsets, 3) Mixing the data within each column of the state array, and 4) Adding a round key to the state.

The input and output used by Rijndael at its external interface are considered to be one dimensional arrays of 8-bit bytes numbered upwards from 0 to the 4*Nb-1. These blocks hence have lengths of 16 bytes array indices in the ranges 0..15. The cipher key is considered to be a one-dimensional arrays of 8-bit bytes numbered upwards from 0 to the 4*Nk-1. These blocks hence have lengths of 16, 24 or 32 bytes and array indices in the ranges 0..15, 0..23 or 0..31 for 128 bit,192 bit and 256 bit key respectively. The cipher input bytes are mapped onto the state bytes in the order $a_{0,0}$, $a_{1,0}$, $a_{2,0}$, $a_{3,0}$, $a_{0,1}$, $a_{1,1}$, $a_{2,1}$, $a_{3,1}$, $a_{4,1}$ ... and the bytes of the cipher key are mapped onto the array in the order $k_{0,0}$, $k_{1,0}$, $k_{2,0}$, $k_{3,0}$, $k_{0,1}$, $k_{1,1}$, $k_{2,1}$, $k_{3,1}$, $k_{4,1}$ ... At the end of the cipher operation, the cipher output is extracted from the state by taking the state bytes in the same order as shown in Figure1.

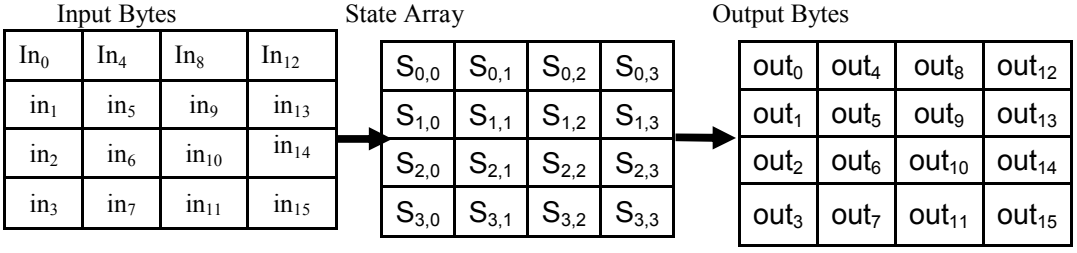

Figure 1: States of the AES

Encryption Process of AES: Four different stages are used, one for permutation and three for substitution. The stages together provide confusion, diffusion and nonlinearity [4]. The stages are as follows:

Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block. For encryption and decryption, this function is indicated by SubBytes () and InvSubBytes () respectively.

Shift rows: A simple permutation. For encryption and decryption, this function is indicated by ShiftRows () and InvShiftRows () respectively.

Mix Columns: A substitution that makes use of arithmetic over $GF(2^8)$, with the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. For encryption and decryption, this function is indicated by MixColumns () and InvMixColumns () respectively [1].

Add round key: A simple bitwise XOR operation of the current block with a portion of the expanded key. For both encryption and decryption this function is indicated by AddRoundKey ().

3 FPGA IMPLIMENTATION

Hardware implementation can be both by ASIC Solution or by FPGA.ASIC(Application Specific Integrated Circuit) solution can be better for mass production but which incurs more design time and also a costly solution[6,11,12].

But Field programmable gate array (FPGA) is an integrated circuit that can be reconfigured by the designer to produce different design and test a lot of circuits with a minimal time which is also be a customizable solution. It means that the circuit may be usable for different application. With each configuration, which takes only fraction of a second, an integrated circuit can perform a completely different function which is said to be a customizable solution [19].

For this reason this project is intended to develop FPGA based customized high speed AES processor to get low latency and high throughput for encryption and decryption. In this project a FPGA solution is developed and tested using Altera provided FPGA and Verilog HDLwith the help of Quartus II software.

4 DESIGN PARTITIONING AND MODULES

The project is partitioned in to main four basic modules and key expansion operation in the main module for encryption and same for decryption cycle.

AES_SUB_BYTE Module: This module performs the Substitution byte operation of AES algorithm. The substitution byte transformation of input state to the output state involves basically two algebraic calculations for each byte which is responsible for high processing time. So for this reason a 16 x 16 byte lookup table is used for substitution to eliminate complex algebraic operation which will increase throughput.

AES_SHIFT_ROW Module: This module is used for performing shift row operation of the AES. The operation is very simple just to alter the position of the bytes on the state matrix.

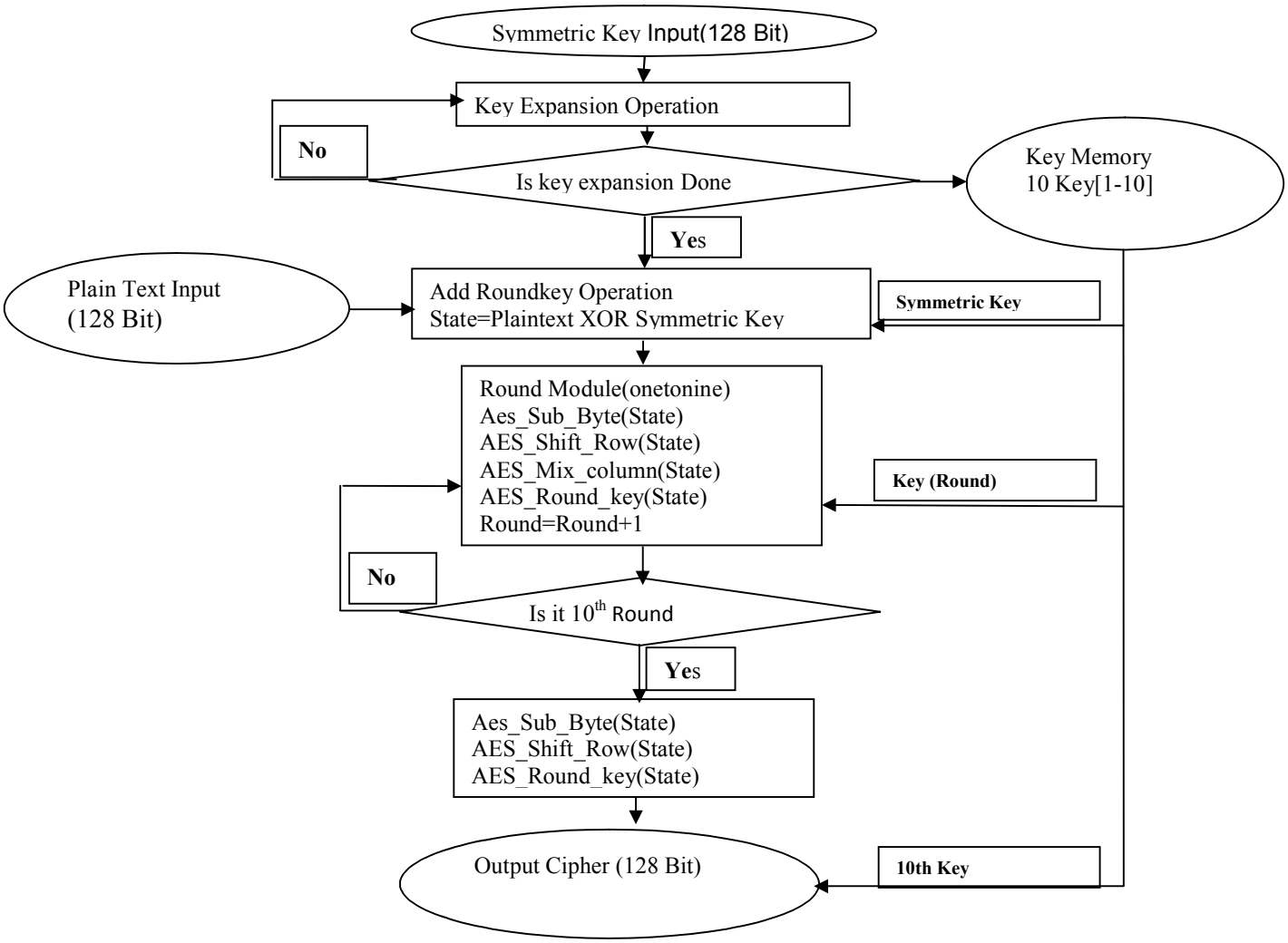

Figure 2: Operational Diagram of modules of AES

AES_MIX_COLUMN Module: This is a operation in AES to multiply the present state of AES to a constant matrix by the multiplication rules used in GF(2^8) Field[1,8].

AES_KEY_EXPANSION: This is very simple. It takes the input key which is supplied for the algorithm which is called symmetric key and generate 10 additional key for next 10 rounds by a complex algebraic operation. Key expansion is performed at the starting of encryption process in the encryption main module.

There is another AES operation which is AddRoundKey where key of each round is XORED with the state during round operation. There is no separate module for AddRoundKey but this is done in main module of encryption and standard modules named as onetonine module where other AES operation are done simultaneously.

Onetonine module: This module is treated as standard module in AES which perform the operation of a standard round in AES. Nine standard rounds includes all four operation of AES such as substitution byte, shift row, AddRoundKey and Mix column operation .In the encryption part at first key expansion is performed to generate ten additional key from the supplied symmetric key and then AddRoundKey which is done by the supplied symmetric key in the main encryption module called AES_Encryption module. After then other nine standard rounds are performed in AES_Encryption module each containing 4 basic operations of AES. The standard module onetonine is called from the main module AES_ENCRYPTION for each round from one to nine.10^{th} Round is different from standard round where mix column operation is skipped. For this reason it is done from AES main module AES_ENCRYPTION. Operational diagram of AES is shown in Figure 2.

In Figure 3 the processing and memory unit used in AES processor is shown where memory S-Box used by Sub Byte module, Round key is generated by key expansion processor using Round constants and supplied key in memory and the key produced are used by add_round_key processor in each round. Inv S-Box memory is used by Inv_sub_byte processor in each round of decryption module.

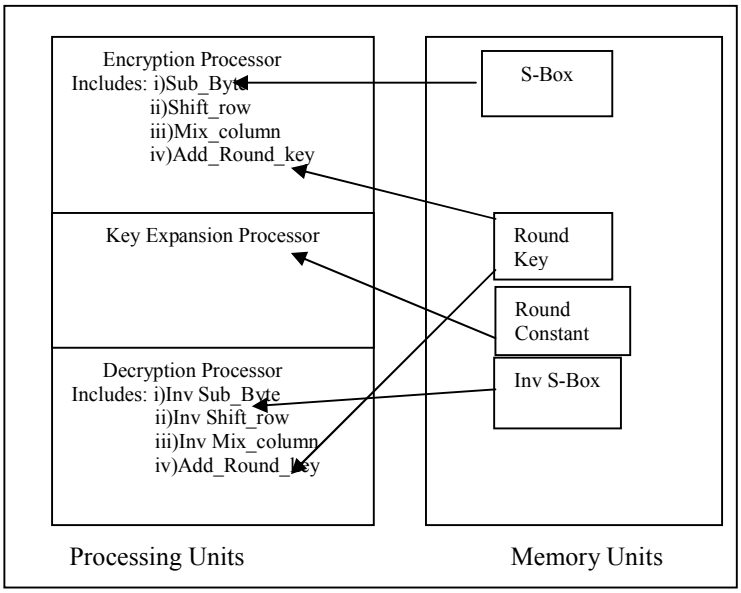

Figure 3: Design Components of the total implementation

5.RESULTS & SIMULATIONS

The design of the AES processor described in the previous section has been coded by Verilog HDL. The Quartus II development software is used for coding and simulation. The design is implemented on Altera provided Cyclone II FPGA which is based on a high performance architecture with sufficient memory. Here in this project the device used is EP2C35F672C6.

Compilation results of Encryption are as follows:

Total Logic Elements	:3405/33216
Total Combinational functions	:3405/33216
Dedicated logic registers:	:1

Total registers	:1
Total Pins	:388/475
Total memory bits	:327680/483,840

Simulation Results: At first each operation like substitution byte, shift row, mix column and key expansion operation are simulated using verilog code and design files. Output is realized using input vectors and key from NIST publication [4].

Figure 4: Simulation of one Standard round (one to nine round)

In encryption module each standard round is simulated and output is verified. Figure 4 shows the simulation of one standard round where all basic operations (Substitution bytes shift rows mix columns and add round key) are performed and latency is observed.

The simulation results of full encryption module is shown in Figure 5 where input vectors and keys are given from NIST standard publication [4] and output was verified.
Input Plain text: 3243f6a8885a308d313198a2e0370734
Input Cipher Key :2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

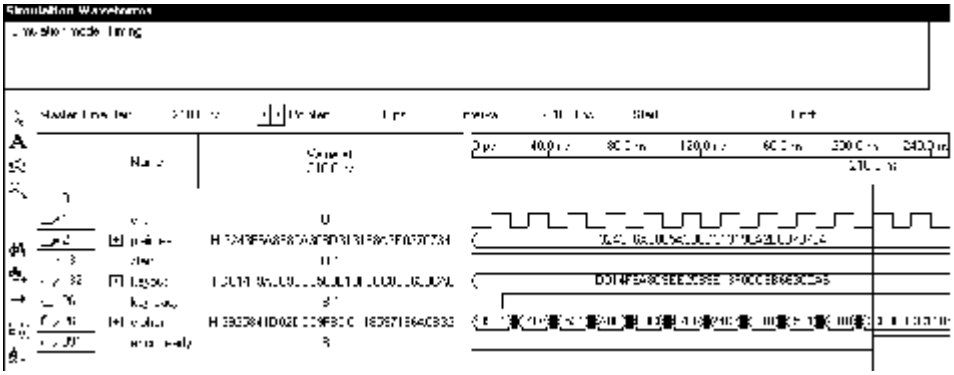

Figure 5: Simulation of full Encryption Module

The output result of the encryption was found accurately after 11 clock cycle from the starting of encryption process. So the latency of encryption is only 11 clock cycle. In the Figure 5 the generated last key(10^{th}) is shown as keyout and latency is observed by keyready function. As the device used is Altera EP2C35F672C6 from Cyclone II family has maximum clock frequency of 50MHz, so the encryption through put will be 6.4Gbps as per clock cycle encrypt 128 bits data samples. If other device having more clock frequency is used then throughput can be increased linearly.

The output of the algorithm are visualized by the 8 seven segment display of the FPGA board where 128 bit cipher produced from 128bits plaintext which is the implementation of simulation results in Figure 5.Inputs are given by toggle switch or by input data to the program. 128 bit encryption key are also given to the code directly.

Overall the simulation of quartus II software and implementation results on the FPGA board found accurate with reduced latency of 11 clock cycle.

6. CONCLUSION

This is the work to implementing a faster cryptosystem in hardware to ensure speedy IT security. A synthesizable Verilog design is developed for each of the encryption and decryption module and has been tested with FPGA cyclone II device EP2C35F672C6.The performance of the each of the sub module with entire Encryption and decryption module is found satisfactory with proper accuracy with minimum latency and speedy throughput which can be described as a simple, portable and efficient AES implementation in a reconfigurable hardware with reduced latency. In this implementation total throughput gained is 6.4Gbps and latency of 11 clock cycle/210ns with max clock frequency of 50Mhz.

AES is a strong algorithm due to its large rounds and algebraic complexity inside the rounds [4]. For this reason this project is conducted to implement the AES in hardware to speed up the AES enabled processing system where minimum latency with required throughput is gained which would required for real time application. Portable electronic system is the vision of this day where power is an important issue. So power analysis of the processor can be carried out. The proposed processor can be implemented on ASIC to improve its performance.

7. ACKNOWLEDGEMENT

I am grateful to IICT, BUET for allowing me to conduct the research using its all kind of faculties.

8. REFERENCES

[1] Stallings W. "Cryptography and Network Security: Principles and Practices. "4th ed., Pearson Education, Inc. pp. 63-173. 2006.

[2] Pfleeger C. " Security in Computing. " Upper Saddle River, NJ: Prentice Hall, 1997.

[3] Schneier B. "Applied Cryptography," 2nd Edition, Wiley, New York, 1996.

[4] "Advanced encryption standard (AES)", Federal Information Processing Standards Publication (FIPS PUB) 197, National Institute of Standards and Technology (NIST), November, 2001.Available at: http://csrc.nist.gov/publication/drafts/dfips-AES.pdf

[5] Daemen J. and Rijmen V., "AES Proposal: Rijndael, " Version 2. Submission to NIST, March 1999. Available at: http://csrc.nist.gov/encryption/aes

[6] Ashwini M. D, Mangesh S. D and Devendra N. K ",FPGA Implementation of AES Encryption and Decryption".

[7] Daemen J. and Rijmen V., "Rijndael: The Advanced Encryption Standard". Dr. Dobb's Journal, March 2001.

[8] NIST, "DRAFT NIST Special Publication 800-131, Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes", Federal Information Processing Standards Publication (FIPS PUB) 197, National Institute of Standards and Technology (NIST), January, 2010.

[9] Leopld G., "U.S. unveils advanced encryption standard," EE Times December 10, 2001. Available at: http://www.eetimes.com/story/OEG20011205S0060.

[10] Qin H., Nonmember, SASAO T. and IGUCHI Y.,Members ,"A Design of AES Encryption Circuit with 128 bit keys using Look-UP Table Ring on FPGA",IEICE TRANS. INF. & SYST.,VOL.E89-D,NO.3 MARCH 2006.

[11] Rahman T., Pan S. and Zhang Q., "Design of a High Throughput 128-bit (Rijndael Block Cipher)",Proceeding of International Multiconferrence of Engineers and computer scientists 2010 Vol II IMECS 2010, March 17- 19,2010, Hongkong.

[12] Hodjat A. and Varbauwhede I.,"A 21.54 Gbits Fully Pipelined AES Processor on FPGA", IEEE Symposim on Field-Programmable Custom Computing Machines,April 2004.

[13] Jarvinen et al, "A fully pipelined memoryless 17.8 Gbps AES-128 encrypter",International Symposium on Field Programmable Gate arrays,pp.207-215.2003.

[14] Cheng K., Chang T. and Lo J., "Cryptanalysis of Security Enhancement for a Modified Authenticated Key Agreement Protocol", International Journal of Network Security, Vol.11, No.1, PP.55- 57, July 2010.

[15] Salama D.A.M, Hatem M. A.K and Hadhoud M.M, " Evaluating the effects of symmetric Cryptography Algorithms on Power Consumption for Different Data Types.", International Journal of Network Security,Vol.11,No.2,PP.78-87,Sept.2010.

[16] Ngo H. H, Wu X., Le D. P, Wilson C., and Srinivasan B., "Dynamic Key Cryptography and Applications", International Journal of Network Security, Vol.10, No.3, PP.161-174, May 2010.

[17] Selvaraju N. and Sekar G., "A Method to Improve the Security Level of ATM Banking Systems Using AES Algorithm", International Journal of Computer Applications (0975 – 8887), Volume 3 – No.6, June 2010.

[18] Zambreno J., Nguyen D. and Choudhary A., "Exploring Area/Delay Tradeoffs in an AES FPGA Implementation",FPL 2004, LNCS 3203, pp. 575–585, 2004.

[19] Mroczkowski P., "Implementation of the block cipher Rijndael using Altera FPGA", May 2000. [Online]. Available WWW: http://csrc.nist.gov/archive/aes/round2/.../20000510-pmroczkowski.pdf

[20] Altera Corp. (2007, February). "Cyclone II device family data sheet [Online]". Available: http://www.altera.com/literature/hb/cyc2/cyc2_cii51001.pdf

[21] Kenny D., "Energy Efficiency Analysis and Implementation of AES on an FPGA", University of Waterloo,2008.

[22] Xiao S.,Chen y. and Luo P., "The Optimized Design of Rijndael Algorithm Based on SOPC", International Conference on Information and Multimedia Technology,2009

[23] Helion Technology Limited, "High performance AES cores for Altera FPGA", Available at: http://www.helion