Area Optimized and Pipelined FPGA Implementation of AES Encryption and Decryption

[1,]Mg Suresh,[2,]Dr.Nataraj.K.R
[1,]Asst Professor Rgit, Bangalore, [2,]Professor
[1,2,]Department Of Electronics And Communication Sjb Institute Of Technology, Bangalore

Abstract

Advanced Encryption Standard (AES), a Federal Information Processing Standard (FIPS), and categorized as Computer Security Standard. The AES algorithm is a block cipher that can encrypt and decrypt digital information. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits. The Rijndael cipher has been selected as the official Advanced Encryption Standard (AES) and it is well suited for hardware. This paper talks of AES 128 bit block and 128 bit cipher key and is implemented on Spartan 3 FPGA using VHDL as the programing language.Here A new FPGA-based implementation scheme of the AES-128 (Advanced Encryption Standard, with 128-bit key) encryption and decryption algorithm is proposed in this paper. The mode of data transmission is modified in this design so that the chip size can be reduced. The 128-bit plaintext and the 128- bit initial key, as well as the 128-bit output of cipher text, are all divided into four 32-bit consecutive units respectively controlled by the clock. This system aims at reduced hardware structure and high throughput.ModelSim SE PLUS 6.3 g software is used for simulation and optimization of the synthesizable VHDL code. Synthesizing and implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx - Project Navigator, ISE 12.1i suite.

Keywords: AES,FPGA,FIPS,VHDL,Plaintext,Ciphertext

1. Introduction

With the rapid development and wide application of computer and communication networks, the information security has aroused high attention. Information security is not only applied to the political, military and diplomatic fields, but also applied to the common fields of people's daily lives. With the continuous development of cryptographic techniques, the long-serving DES algorithm with 56-bit key length has been broken because of the defect of short keys. So AES (Advanced Encryption Standard) substitutes DES and has already become the new standard. AES algorithm is already supported by a few international standards at present, and AES algorithm is widely applied in the financial field in domestic, such as realizing authenticated encryption in ATM, magnetism card and intelligence card. In 1997, an effort was initiated to develop a new American encryption standard to be commonly used well into the next century. This new standard was given a name AES, Advanced Encryption Standard. A new algorithm was selected through a contest organized by the National Institute of Standards and Technology (NIST). By June 1998, fifteen candidate algorithms have been submitted to NIST by research groups from all over the world. After the first round of analysis was concluded in August 1999, the number of candidates was reduced to final five. The five algorithms selected were MARS, RC6, RIJNDAEL, SERPENT and TWOFISH. The conclusion was that the five Competitors showed similar characteristics. On October 2nd 2000, NIST announced that the Rijndael Algorithm as the winner of the contest. The primary criteria used by NIST to evaluate AES candidates included security, efficiency in software and hardware, and flexibility. Rijndael Algorithm developed by Joan Daemen and Vincent Rijmen. Was chosen since it had the best overall scores in security, performance, efficiency, implementation ability and flexibility. Hence chosen as the standard AES (Advanced Encryption Standard) algorithm's 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 hardware implementation of the Rijndael algorithm can provide either high performance or low cost for specific applications.

AES algorithm is widely applied in the financial field in domestic, such as realizing authenticated encryption in ATM, magnetism card and intelligence card. On the current situation of researching at home and abroad, AES algorithm emphasizes its throughput using pipeline pattern. Its biggest advantage is to improve the system throughput, but there is a clear disadvantage that is at the cost of on-chip resources. And in accordance to that AES algorithm is used in the low requirements of the terminal throughput at present, the high safety and cost-effective reduced AES system is designed and validated on the xilinx spartan 3 chip aiming at reduced hardware structure. The advantages in this system are high speed, high reliability, a smaller chip area, and high cost-effective. These will effectively promote the AES algorithm to be used in the terminal equipments.Hardware security solution based on highly optimized programmable FPGA provides the parallel processing Capabilities and can achieve the required encryption performance benchmarks. The current area-optimized algorithms of AES are mainly based on the realization of S-box mode and the minimizing of the internal registers which could save the area of IP core significantly.

Cryptographic algorithms are most efficiently implemented in custom hardware than in software running on general purpose processors. Hardware implementations are of extreme importance in case of high performance, security against system intruders and busy systems, where a cryptographic task consumes too much time. Traditional ASIC solutions have the well-known drawback of reduced flexibility compared to software solutions. Hence the implementation of the AES algorithm based on FPGA devices has certain advantages over the implementation based on ASICs.One new AES algorithm with 128-bit keys (AES-128) was described in this paper, which was realized in VHDL. The 128-bit plaintext and 128-bit key, as well as the 128-bit output data were all divided into four 32-bit consecutive units respectively. The pipelining technology was utilized in the intermediate nine round transformations so that the new algorithm achieved a balance between encryption speed and chip area, which met the requirements of practical application The data of each column (32 its) in the state matrix was used to be an operand of encryption, when the operation of ShiftRows and SubBytes were incorporated. And each round of the intermediate nine Round Transformations of encryption was processed by pipelining technology.The main problem in older algorithms is area utilization on chip is more. These are of low performance and low throughput architectures. Further, many of the architectures are not area efficient and can result in higher cost when implemented in silicon. In this proposed method, for maintaining the speed of encryption, the pipelining technology is applied and the mode of data transmission is modified in this design so that the chip size can be reduced. The 128-bit plaintext and the 128- bit initial key, as well as the 128-bit output of ciphertext, which are all divided into four 32-bit consecutive units respectively controlled by the clock.Hence we overcome the area utilization problem. Here area and throughput are carefully trading off to make it suitable for wireless military communication and mobile telephony where emphasis is on the speed as well as on area of implementation.

A.Objective

The main objective of the paper is to design and development of AES system using FPGA in computer communication networks such as internet and to perform the verification and validation of the developed system.

- First design the AES encryption algorithm using VHDL Language in Xilinx 12.2 and simulation is done by Modelsim 6.3f.
- Second design the AES decryption algorithm using VHDL Language in Xilinx 12.2 and simulation is done by Modelsim 6.3f. In the standard AES algorithm, there are four steps like SubByte, ShiftRow, MixColumn and Add Round Key in normal rounds. In Our design we highlight some following modifications:

1) Exclusion of Shift Row OR Merging of Sub Byte and Shift Row
2) Pipelining for high Throughput
3) Optimizing the design to keep handy balance between Throughput and Silicon Area .

B.Existing System:

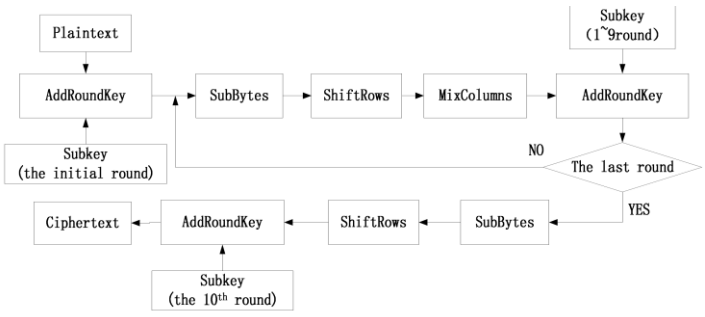

Figure1. The Structure of Rijndael Encryption Algorithm

Rijndael algorithm consists of encryption, decryption and key schedule algorithm. The main operations of the encryption algorithm among the three parts of Rijndael algorithm include: bytes substitution (SubBytes), the row shift (ShiftRows), column mixing (MixColumns), and the round key adding (AddRoundKey).Similarly Decryption includes:Inverse bytes substitution (InvSubBytes), the Inverse row shift (InvShiftRows), Inverse column mixing (InvMixColumns), and the round key adding (AddRoundKey). Exclusive-OR operation is an inverse of itself, so there is no need for specific InvAddRoundKey transformation the same AddRoundKey operation is used during both encryption and decryption rounds.. Encryption algorithm processes Nr+1 rounds of AddRoundKey transformation of the plaintext for the ciphertext. Decryption algorithm processes Nr+1 rounds of InvAddRoundKey transformation of the ciphertext for the plaintext .The value of Nr in AES algorithm whose packet length is 128 bits should be 10 for the key length of 128.

2. Implementation Of Area Optimized And Pipelined AES Encryption And Decryption

A. The Design of Improved AES Algorithm

2. Two main processes of improved AES Encryption and Decryption algorithm:

The AES Encryption algorithm can be divided into two parts, the key schedule and round transformation. similarly in decryption also. Key schedule consists of two modules: key expansion and round key selection. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selectes Nb bits of round key from the expanded key module.Round Transformation involves four modules by ByteSubstitution, ByteRotation, MixColumn and AddRoundKey in the Encryption .where as in Decryption the Round Transformation involves four modules Inv ByteSubstitution, InvByteRotation,InvMixColumn and AddRoundKey

2) Key points for the design:

In the AES-128, the data in the main process mentioned above is mapped to a 4×4 two-dimensional matrix. The matrix is also called state matrix, state matrix in at the begin of Encryption process will have the plain text. Where as at the start of Decryption process will contain cipher text which is shown as Figure.2.

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Figure 2: The state matrix

In the four transformation modules of round transformation of the Encryption process, the Byte Rotation, Mix Column and Add Round Key are all linear transformations except the Byte Sub.Similary in Decryption process the modules of round transformation the Inv Byte Rotation, Inv Mix Column and AddRoundKey are all linear transformations except the Inv Byte Sub.

Take analysis of the AES Encryption and Decryption algorithm principle and we can find:

- Byte Substitution(Sub Bytes) operation simply replaces the element of 128-bit input plaintext with the element corresponding to the Galois field GF (2^8), whose smallest unit of operation is 8 bits/ group. Inv Byte Substitution Transformation in the Decryption(inverse cipher) is the inverse of Byte Substitution (Sub Bytes).
- Byte Rotation(Shift Rows) operation takes cyclic shift of the 128-bit state matrix, in which one row (32 bits) is taken as the smallest operand. . Inv Byte Rotation Transformation in the Decryption(inverse cipher) is the inverse of Byte Rotation (Shift Rows).
- Mix Column operation takes multiplication and addition operations of the results of Byte Rotation with the corresponding irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ in GF(2^8), whose minimum operating unit is 32 bits. . Inv Mix Column Transformation in the Decryption(inverse cipher) is the inverse of Mix Column.
- Add round key operation takes a simple XOR operation with 8-bit units. Since Exclusive-OR operation is an inverse of itself, so there is no need for specific Inv Add Round Key transformation the same Add Round Key operation is used during both Encryption and Decryption rounds.

The inputs of plaintext and initial key, intermediate inputs and outputs of round transformation, as well as the output of cipher text in the AES Encryption algorithm are all stored in the state matrixes. Similarly the inputs of ciphertext and initial key, intermediate inputs and outputs of round transformation, as well as the output of plaintext in the AES Decryption algorithm are all stored in the state matrixes which are processed in one byte or one word. Thus, in order to take operations at least bits, the original 128-bit data should be segmented.We design some external controllers in the new algorithm, so that the data transmission and processing can be implemented on each column of the state matrix (32bit). That means the data should be packed and put into further operations.Take the independent and reversible bytes substitution operation of S-box as example. Firstly, the state matrix is divided into four columns. And then byte replacement is achieved by the operation of look-up table shown as Figure 3.

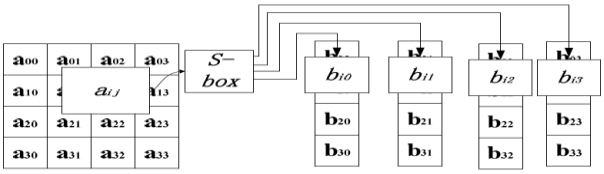

Figure 3: Bytes Segmentation and replacement processing

Therefore, the original 128-bit inputs and key will be replaced with four consecutive 32-bit input sequences respectively. In order to decrease the output ports, four continuous 32-bit output sequences have taken place of the original 128-bit output by adding a clock controller. The 128- bit data in the round transformation is also split into four groups of 32-bit data before the operation of pipelining.

B.PROPOSED MODEL :

In the proposed method architectural optimization has been incorporated which includes pipelining techniques. Speed is increased by processing multiple rounds simultaneously but at the cost of increased area. The corresponding hardware realization is optimal in terms of area and offers high data throughout. An optimized code for the implementation of this algorithm for 128 bits has been developed and experimentally tested using Xilinx device. So the focal approach of our design on hardware platform is to attain speed up (i.e. high throughput No. of block processed per second) at the same time, silicon area optimization.From the above analysis, we can find that the process of AES encryption and Decryption can be mainly divided into two parts: key schedule and round transformation. The improved structure is also divided into these two major processes. The initial key will be sent to the two modules: Key expansion and Keyselection,While the plaintext is to be sent to the round transformation during Encryption and cipher textis sent to round transformation after the roundkey is selected. But the operand of data transmission is turned into a 32-bit unit.

The process of new algorithm is shown as Figure 4(a) & (b).

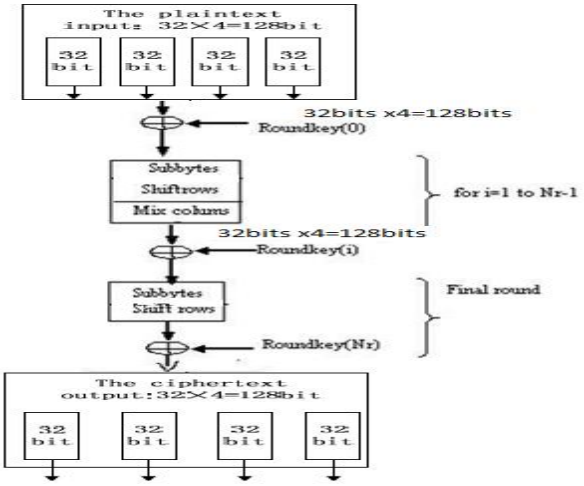

Figure 4(a): Proposed AES algorithm for Encryption

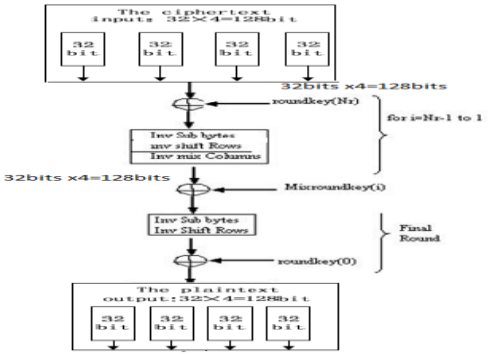

Figure 4(b) Proposed AES algorithm for Decryption Flow

In AES Encryption and Decryption process, first of all, a initial round of transformation has to be performed

3. The initial round of encryption

Which XORs the original 128 bits of input data with the original 128 bits of key i.e. four packets of consecutive 32-bit plaintext (Encryption) or four packets of consecutive 32-bit Ciphertext (Decryption) have been put into the corresponding registers. Meanwhile, another four packets of consecutive 32-bit initial key (128 bits) have been put into other registers by the control of the enable clock signal. Furthermore, this module should combine the plaintext and initial key by using the XOR operator during Encryption where as in the Decryption this module should combine the ciphertext and initial key by using the XOR operator.

4. Round Transformation in the intermediate steps:

A round transformation mainly realizes the function of SubBytes and MixColumns with 32-bit columns during Encryption and InvSubBytes and InvMixColumns with 32-bit columns during Decryption. Here **Exclusion of Shift Row is** performed by calling required shifted element from the data matrix, instead of calling element one by one sequentially from the data matrix. Thereby SUB-BYTE and SHIFT ROW for encryption (which is known as BYTE ROTATE when the shift is done on register or 32 bit word) operations are carried out in one-step instead of two similarly for Decryption**.** Four packets of round transformation are processed independently. Then the results of MixColumns and InvMix Column and the 32-bit keys sourced from Key expansion are combined by using XOR operators respectively. Here, the round transformation is a module with 64 input ports (32-bit plaintext+32-bit key) and 32 output ports. The function of SubByte and Inv SubByte is realized by Look-Up Table (LUT). It means that the operation is completed by the Find and Replace after all replacement units are stored in a memory (256×8bit = 1024 bit). The implementation of MixColumn and InvMixColumn is mainly based on the mathematical analysis in the Galois field GF(2^8) and Inv Galois field GF(2^{-8}). Only the multiplication module and the 32-bit XOR module of each processing unit(one column) are needed to design, because the elements of the multiplication and addition in Galois field are commutative and associative. Then the function of MixColumn and InvMixColumn can be achieved.

Figure.5 is a block diagram for the introduction of pipelining technology used in the round transformation.

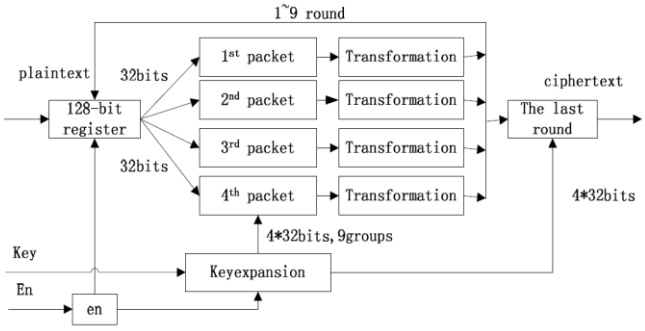

Figure5: The Process of Pipelining

In the process of pipelining, the 128-bit data is divided into four consecutive 32-bit packets that take round transformation independently. The operation of the above four groups of data can be realized in pipelining technology. In brief, it can be described as follow: store the unprocessed data in the 128-bit register, and control the clock for restarting the 128-bit register to read the new data when the four groups' operations have been overcome. Thus the 128-bit round-operating unit has been transformed into four 32-bit round-operating elements. The internal pipelining processing should be implemented during the whole nine intermediate Round Transformations of the four packets before achieving the 128-bit ciphertext for Encryption and 128-bit plaintext for Decryption.

5. The process of the last round:

The final round is a 128-bit processor. After nine rounds of operations included Shiftrows, SubByte and Mixclumns during Encryption and After nine rounds of operations included InvShiftrows, InvSubByte and InvMixclumns during Decryption. The 128-bit intermediate resultant data will be used in XOR operation with the final expanded key (4*32bit), which is provided by the key expansion module. The output of final round of Encryption in the processor is the desired 128-bit ciphertext. Similarly, the ciphertext is divided into four packets of 32- bit data by an external enable signal. . The output of final round of Decryption in the processor is the desired 128-bit plaintext. Similarly, the plaintext is divided into four packets of 32- bit data by an external enable signal. The proper selection of the module and data path for a particular round is done by the control signals. These signals also control the Key Scheduling module so that valid keys are called for the particular round.

6. Key expansion and Key extraction

This module is implemented basically the same with the traditional way as another part of the AES encryption algorithm.

The only difference lies on the mode of data transmission. The initial key and expanded keys are divided into four 32-bit data before being extracted.

All of the above modules can be decomposed into basic operations of seeking and XOR if the AES algorithm is Implemented on FPGA. So the basic processing unit (look-uptable) of FPGA can be used. The operation of AddRoundKey is taken first in each round. When the plaintext and initial key are input, the encryption module starts running, and the expanded keys are stored into the registers at the same time. This implementation method is independent on a specific FPGA.

7. SIMULATION RESULTS

ModelSim SE PLUS 6.3 g software is used for simulation and optimization of the synthesizable VHDL code. Synthesizing and implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx - Project Navigator, ISE 12.1i suite.Initially we designed AES module in such a way that it should be implemented on a single chip, but when we implemented on FPGA kit (SPARTAN-III), our design exceeded more than 4 lakhs gates. This made impossible to implement the AES design module to dump on a single chip. As a result we implemented the AES Encryption module on a FPGA for the verification of the hardware implementation of the design.When we feed 16 bytes(128 bit four 32 bits packets) of data in case of AES encryption into Modelsim simulator, In Figure6 we can observe that all the 16 bytes of encrypted data for AES, can be observed on the output of FPGA also In terms of 8 bits of chunk.For verification of the working of Decryption we have feed the output of encryption to the decryption module and observed that the output has regenerated the original text at the output in Figure7. We have forced HELLO WORLD as an input to the design and obtained the original data back after the decryption of the encrypted message with the use of same key as seen in Figure8. We have also monitored that if the single bit of the decryption key is change we are not in a position to retrieve the original data back.

Figure6: Simulation of 128-bit AES Encryption

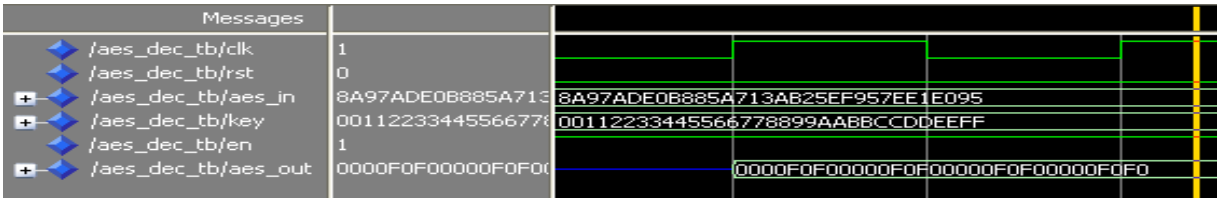

Figure7: Simulation of 128-bit AES decryption

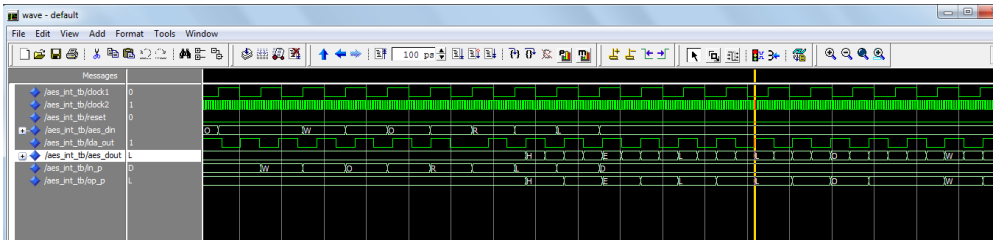

Figure8: HELLO WORLD is forced as input to the design and obtained the original data back

8. Conclusion And Future Work

From our work we have concluded that the concept of Pipelined AES architecture can be practically implemented. It has been observed that the implementation of AES Encryption on the FPGA is successful and several data input. The cipher key can be changed with respect to the user requirements.The result shows that the design with the pipelining technology and special data transmission mode can optimize the chip area effectively. Meanwhile, this design reduces power consumption to some extent, for the power consumption is directly related to the chip area. Therefore the encryption device implemented in this method can meet some practical applications.As the S-box is implemented by look-up-table in this design, the chip area and power can still be optimized. So the future work should focus on the implementation mode of S-box. Mathematics in Galois field (2^8) can accomplish the bytes substitution of the AES algorithm, which could be another idea of further research.While implementing the AES Algorithm, the critical aspect was the area utilization. Which was done using implementation of functions for different sub modules in the algorithm the work has approximately reduced around 10% utilization on chip as compared to basic available modules.

We have successfully implemented AES encryption on FPGA. We have achieved the data encryption as per 100% accuracy as compared to data encryption module available online. The next step is to study the behavior of the AES Model in extreme conditions and to implement as an ASIC. There is a provision and flexibility to remove or add any other cryptographic standards.

References

1. Oded Goodrich, *Foundations of Cryptography, Volume 1: Basic Tools*, Cambridge University Press, 2001.
2. "Cryptology (definition)". *Merriam-Webster's Collegiate Dictionary* (11th edition.). Merriam-Webster.http://www.merriam-webster.com/dictionary/cryptology/. Retrieved on 2008-02-01.
3. http://Advanced Encryption Standard - Wikipedia, the free encyclopedia.html
4. J.Yang, J.Ding, N.Li and Y.X.Guo,"FPGA-based design and implementation of reduced AES algorithm" IEEE Inter.Conf. Chal Envir Sci Com Engin(CESCE)., Vol.02, Issue.5-6, pp.67-70, Jun 2010.
5. A.M.Deshpande, M.S.Deshpande and D.N.Kayatanavar,"FPGA Implementation of AES Encryption and Decryption"IEEE Inter.Conf.Cont,Auto,Com,and Ener., vol.01,issue04, pp.1-6,Jun.2009.
6. Hiremath.S. and Suma.M.S.,"Advanced Encryption Standard Implemented on FPGA" IEEE Inter.Conf. Comp Elec Engin.(IECEE),vol.02,issue.28,pp.656-660,Dec.2009.
7. Abdel-hafeez.S.,Sawalmeh.A. and Bataineh.S.,"High Performance AES Design using Pipelining Structure over GF(28)" IEEE Inter Conf.Signal Proc and Com.,vol.24-27, pp.716-719,Nov. 2007.
8. Rizk.M.R.M. and Morsy, M., "Optimized Area and Optimized Speed Hardware Implementations of AES on FPGA", IEEE Inter Conf. Desig Tes Wor.,vol.1,issue.16,pp.207-217, Dec. 2007.
9. Liberatori.M.,Otero.F.,Bonadero.J.C. and Castineira.J. "AES-128 Cipher. High Speed, Low Cost FPGA Implementation", IEEE Conf. Southern Programmable Logic(SPL),vol.04,issue.07,pp.195-198,Jun. 2007.
10. Abdelhalim.M.B., Aslan.H.K. and Farouk.H. "A design for an FPGAbased implementation of Rijndael cipher",ITICT. Ena Techn N Kn Soc.(ETNKS), vol.5,issue.6,pp.897-912,Dec.2005.
11. Pong P. Chu, RTL Hardware Design Using VHDL, John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.
12. National Institute of Standards and Technology, "Specification for the ADVANCED ENCRYPTION STANDARD (AES)", Federal Information Processing Standards Publication 197, November 26, 2001 (http://www.csrc.nist.gov).
13. Łukasz Krukowski, "Realizacja algorytmów szyfrowania w układach FPGA", dissertation for M.Sc. degree, Wrocław University of Technology, Faculty of Electronics, 2007 (in Polish).
14. A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography,CRC Press, New York, 1997, p. 81-83.
15. Joseph Zambreno, Student member, IEEE, Dan Honbo, student Member, IEEE, Alok Choudhary, Fellow,IEEE, Rahul Simha, Member, IEEE, and Bhagirath Narahari, "High performance Software Protection Using Reconfigurable Architectures", Proceedings of the IEEE, volume 94, No. 2, February 2006.
16. Thomas Wollinger, Jorge Guajardo and Christ of Paar, "Cryptography on FPGAs: State of the Art Implementations and Attacks", Special Issue on Embedded Systems and Security of the ACM Transactions in Embedded Computing Systems (TECS).