ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.5

Implementation of Advanced Encryption Standard (AES) on FPGA

Ankita Jadhav[1], Prajakta Choudhari[2], Tejaswini Gherade[3], S. C. Wagaj[4]
BE Studen[1, 2, 3], Professor[4]
Department of E&TC
Rajarshi Shahu College of Engineering, Savitribai Phule Pune University, Maharahtra, India

Abstract:
A high speed security algorithm is always necessary and important for wired/wireless communication. One of the best existing symmetric security algorithms to give information security is advanced encryption standard (AES). The significance of cryptography applied to security in electronic data transactions has obtained an fundamental importance amid during the last few years. A proposed FPGA-based implementation of the Advanced Encryption Standard (AES) algorithm is presented in this paper. The design has been coded by Very high speed integrated circuit Hardware Descriptive Language. All the results are synthesized and simulated using Xilinx ISE and ModelSim software respectively. This implementation is compared with other works to show the efficiency. The design uses an iterative looping approach with block and key size of 128 bits, lookup table implementation of S-box. This gives low complexity architecture and effortlessly accomplishes low latency and in addition high throughput. Simulation results, performance results are given and compared with previous reported designs.

Keywords: AES, FPGA, VHDL, encryption, decryption and block cipher.

I. INTRODUCTION

Cryptography is the study of Mathematical methods for secured communication within the presence of adversaries and furthermore it manages the parts of data security, for example, confidentiality, data integrity, entity authentication and data origin authentication. The advanced encryption standard (AES), standardized by NIST, National Institute of Standards and Technology, is a cryptographic algorithm replacement to DES (Data Encryption Standard) algorithm as the federal standard to protect sensitive information. AES has effectively received wide spread use because of its high security, high performance in both hardware and software. Many implementations are done in software but it seems to be too slow for fast applications for example, routers and some wireless communication systems. The few of AES hardware implementation architectures and optimizations have been recommended for various applications. AES algorithm can oppose any kinds of password attacks with a strong practicability in information security and reliability. AES can be actualized in software or hardware but, hardware implementation is more suitable for high speed applications in real time. The common goal of cryptographic algorithms is providing security. From last several years, Data Encryption Standard (DES) had been utilized as a cryptographic algorithm. Because of the short key length of DES it is replaced by the Rijndael algorithm which has became as a standard in the cryptography domain, known as Advanced Encryption Standard (AES). Everyday millions of users produce and interchange huge volumes of data in different fields for example, medical reports, and bank services via Internet. Every one of these applications deserve a special treatment from the security point of view, not just in the transport of such information. Here, cryptography procedures are particularly applicable. DES considered being insecure for many applications. This is due to the 56-bit key size being too small, in January, 1999, distributed and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes. Furthermore this is the reason, why the National Institute of Standards and Technology (NIST) opened a formal call for algorithms in September 1997. So a group of fifteen AES candidate algorithms were announced in August 1998. This paper is organized as follows. Section III presents proposed Methodology and Section IV gives the overall Implementation. Finaly, Section V provides the Conclusion.

II. LITERATURE SURVEY

The design utilizes an iterative looping approach with block and key size of 128 bits (Hoang Trang and Nguyen Van, 2012). The design has been coded by Verilog HDL. Every one of the results are synthesized and simulated basing on the Quatus 9.0, the Model Sim. So the latency of encryption is 51 clock cycles on Xilinx platform. Similarly, the latency of decryption is 51 clock cycles. The algorithm achieves a low latency and the throughput reaches the value of 1054Mbit/sec for encryption and 615 Mbit/sec for decryption[1]. An implementation of high speed AES algorithm based on FPGA is presented, in order to enhance the safety of information in transmission (WANG Wei, CHEN Jie & XU Fei, 2012). Mathematic principle, encryption process and logic structure of AES algorithm are introduced. In order to achieve the purpose of enhancing the system computing speed, the pipelining and parallel processing methods were utilized. The simulation results show that the high-speed AES encryption algorithm implemented correctly. Design was tested on Xilinx Virtex-5 FPGA. All the processes including Implementation and Simulation are finished in Model Sim ISE 13.3 development platform. Results show that the system could complete the whole process correctly in a 200MHz clock rate[2]. This system aims at diminished hardware structure (Yang Jun Ding Jun Li Na Guo Yixiong, 2010). And this framework has high security and dependability. The advantage of this design is the fact that we do

not need to store the round key since they are currently calculated in accordance that AES algorithm is utilized in the low requirements of the terminal throughput at present, the high wellbeing and cost-effective diminished AES system is designed and approved on the Altera Cyclone, aiming at reduced hardware structure. Moreover, this system can be broadly utilized in the terminal equipments which less demand on the throughput. Throughput found for encryption and decryption process is 593.45Mbps and 267.63Mbps respectively[3]. The usage of FPGA based AES algorithm is cryptography. The cryptography is the science of secret codes empowering the confidentiality of communication through insecure channel. Cryptosystems can give confidentiality, authenticity, integrity, and non repudiation services. It secures against unauthorized parties by preventing unauthorized alteration of utilization. As a rule, it utilizes a cryptographic system to transform a plaintext into a cipher text, using most of the time a key. To increase the computational speed parallelism and pipelining architecture have been implemented. The simulation is done utilizing Xilinx 13.2 version. It doesn't give accessibility of information or frameworks. The salient feature of AES encryption and decryption are high throughput, parameter flexibility, implantation flexibility, no known security attack Confidentiality means that unauthorized parties cannot get to information. Authenticity refers to validating the source of the message to ensure the sender is properly identified. Integrity provides assurance that the message was not modified during transmission, accidentally or intentionally. Non repudiation means that a sender cannot deny sending the message at a later date, and the receiver can't deny receiving it. Cipher is any method of encrypting text (concealing its readability and meaning). It is also sometimes used to refer to the encrypted text message itself. A block cipher is one that breaks a message up into chunks and combines a key with each chunk (for example, 64-bits of text) [4]. This depicts a high effective AES core hardware architecture for implementing it to encrypt/decrypt the data in portable hard disk drive system that apply to successfully in the terms of speed, scale size and power utilization to comply with minimum speed of 5 Gbps (USB3.0). We proposed the 128 bits data path of two different AES architectures design, Basic Iterative AES, which reuses the same hardware for all the ten iterations and, One Stage Sub Pipelined AES, with one stage of outer pipelining in the data blocks that both of them are purely 128 bits data path architecture that different from the previous public paper. The implementation result on the targeted FPGA, the fundamental iterative AES encryption can offer the throughput of 3.85 Gbps at 300 MHz and one stage sub pipelined AES can offer the throughput to expand the efficiency of 6.2 Gbps at 481 MHz clock speed [5].

III. PROPOSED METHODOLOGY

The Advanced Encryption Standard (AES) is a standard for the encryption of electronic information. The AES-128 Algorithm incorporates the following functions i.e. 128-bit key size, Automatic Round key calculation and Encryption or decryption functions. In this paper, we design the 128 bit AES algorithm in encryption and decryption process. We conduct a fault attack against the unprotected AES by using VHDL code.

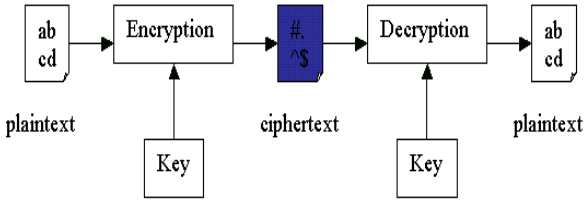

Figure.1. Block Diagram of AES

Number of people and associations utilizing wide computer networks for personal and professional activities has recently increased a lot. A cryptographic algorithm is an fundamental part in network security. A well-known crypto-graphic algorithm is the Data Encryption Standard (DES) which has been broadly received in security products. However, serious considerations arise for long-term security due to the generally short key word length of only 56 bits and from the very effective cryptanalysis attacks. In November 2001, the National Institute of Standards and Technology (NIST) of the United States chose the AES algorithm as the reasonable Advanced Encryption Standard (AES) to supplant the DES algorithm. Since then, many hardware implementations have been proposed in literature some of them utilize field programmable gate arrays (FPGA) and some use application-specific integrated circuits (ASIC). The advantages of a software implementation include convenience, ease of upgrade and versatility. There are some restrictions of software implementation as it offers only limited physical security, particularly with respect to key storage. Alternately, cryptographic algorithms (and their associated keys) implemented in hardware are more physically secure since they cannot easily be read or modified by an outside attacker. The downside of traditional (ASIC) hardware implementations is the absence of adaptability with respect to algorithm and parameter switching. Reconfigurable hardware devices such as FPGAs are a promising alternative for the implementation of block ciphers. FPGAs are hardware devices whose function are not fixed and can be programmed in-system. In this paper, we present an implementation of the AES block cipher with Virtex II Pro FPGA using 0.13 mm and 90 nm process technologies. We have exploited the temporal parallelism available in the AES algorithm.

A. AES Algorithm

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192. AES is the most recent of the four current algorithms approved for federal us in the United States. AES is a symmetric encryption algorithm processing data in block of 128 bits. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption . The only secret necessary to keep for security is the key. AES may configured to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. The older standard, DES or Data Encryption Standard. DES is upto 56bits only [4]. To overcome the disadvantages of des algorithm, the new standard is AES algorithm. The AES algorithm is a symmetric block cipher that processes data blocks of 128 bits using a cipher key of 128, 192, or 256 bits length. Here each data block consists of a 4!4 array of bytes known as the state, on which the basic operations of the

AES algorithm are performed. Fig. 1 shows the AES encryption and decryption procedures.

The encryption procedure is as follows:-

After an initial round key addition, a round function consisting of four different transformations—byte-sub, shift-row, mix-column, and add-round-key—is applied to the data block in the encryption procedure. The round function is performed 10, 12, or 14 times, depending on the key length. The mix-column operation is not applied to the last round. The byte-sub operation is a nonlinear byte substitution that operates independently on each byte of the state using a substitution table (S-Box). The shift-row operation is a circular shifting on the rows of the state with different numbers of bytes (offsets). The mix-column operation mixes the bytes in each column by the multiplication of the state with a fixed polynomial modulo x4C1. Add-round-key operation is an XOR that adds a round key to the state in each iteration, where the round keys are generated during the key expansion phase. The byte-sub transformation (S-Box operation), which consists of a multiplicative inverse over GF(28) and an affine transform, is the most critical part of the AES algorithm in terms of computational complexity. However, the S-Box operation is required for both encryption and key expansion. Conventionally, the coefficients of the S-Box and inverse S-Box are stored in the lookup tables, or a hard-wired multiplicative inverter over GF (28) can be used, together with an affine transformation circuit. The decryption procedure of the AES is basically the inverse of each transformation. However, the standard decryption procedure is not identical to the encryption procedure. That is, the sequence of transformations for decryption is different from that for encryption though the form of the key schedules for encryption and decryption the same. There is, however, an equivalent version of the decryption procedure that has the same structure as the encryption procedure. The equivalent version has the same sequence of transformations as the encryption procedure (with transformations replaced by their inverses). standard explicitly defines the allowed values for the key length (*Nk*), block size (*Nb*), and number of rounds (*Nr*).

1. AES Algorithm Specification:

For the AES algorithm, the length of the input block, the output block and the State is 128 bits. This is represented by *Nb* = 4, which reflects the number of 32-bit words (number of columns). An implementation of the AES algorithm shall support *at least one* of the three key lengths: 128, 192, or 256 bits (i.e., *Nk* = 4, 6, or 8, respectively). Implementations may optionally support two or three key lengths, which may promote the interoperability of algorithm implementations. For the AES algorithm, the length of the Cipher Key, *K*, is 128, 192 or 256 bits. The key length is represented by *Nk* = 4, 6, or 8which reflects the number of 32-bit words (number of columns) in the Cipher Key. 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 *Nk* = 4, *Nr* = 12 when International Journal of Scientific & Engineering Research Volume 3, Issue 3, March -2012 2 ISSN 2229-5518 IJSER © 2012 http://www .ijser.org *Nk* = 6, and *Nr* = 14 when *Nk* = 8. The only Key-Block-Round combinations that conform to this standard are given in Table 1.

Table.1. Key-Block-Round Combinations.

Bit pattern	Key Length (N_K Words)	Block Size (N_B Words)	No Of Rounds (N_R Words)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

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.

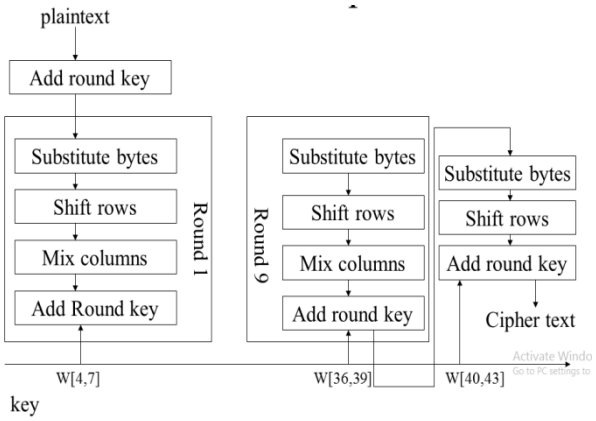

Figure.2. AES Encryption/Decryption process

2. AES Cipher

For 128-bit key size, there are 10 rounds substitutions and permutations that have to be executed in AES cipher (see Table 3.1). The input 128-bit plaintext is presented in a 4x4 matrix of bytes. Thus, there are 32 bits each row and each column in the matrix. This matrix is also called State array which is illustrated in table 3.1. I, Si,j indicates a byte, where $0 \leq i,j \leq 3$. The state array is altered in each round. The input key is expanded into an array of forty four 32-bit words, and each 4 words of the expanded key will be used in each round. The key expansion should be done before the cipher operation. Each round transformation consists of four phases as follows:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

Sub Bytes the function Sub Bytes is the only non-linear function in AES. It substitutes all bytes of the state array using a LUT which is a 16x16 matrix of bytes, often called S-box. The S-box is used for SubBytes operation that contains the results of substitution and permutation of all possible 8-bit values. The content of the LUT can be computed by a finite-field inversion followed by an affine transformation over GF(28). Each byte of state is mapped into a byte from the S-box; The 4 leftmost bits are used as the row index while the 4 rightmost bits are used as the column index. Figure 3.1 illustrates the effect of the SubBytes transformation on the State array. The S-box is

designed to be resistance to known cryptanalytic attacks. SubBytes function has a property that the output cannot be described as a simple mathematical function of the input.

IV. IMPLEMENTATION

The AES algorithm is a symmetric block cipher that cans encryption and decryption information. Encryption converts data to a unintelligible form called ciphertext. Decryption of the cipher-text converts the data back into it's a original, which is called plain-text.

A. AES encryption

The AES algorithm operates on a 128-bit block of data. The key length is 128, 192 or 256 bits in length respectively. The pre-rpound and last rounds differ from other rounds, there is an AddRoundKey transformation in pre-round and no MixCoulmns transformation is performed in the last round as shown in fig. 1. In this paper, we use the key length of 128 bits as a model for general explanation.

Steps in AES Encryption

- Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
- Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
- Add Round Key—each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule

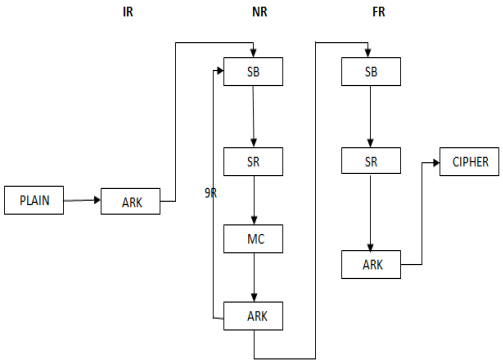

Figure. 3. General structure of Encryption.

1. SUBBYTE TRANSFORMATION

The SubBytes transformation includes non-linear byte substitution, operating on each of the state bytes independently. This is done by using a once-precalculated substitution table called S-box. S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. Advantage of performing the S-box computation in a single clock cycle, reducing the latency and avoids complexity of hardware implementation. In SubBytes transformation there is a non-linear byte substitution, operating on each of the state bytes independently as shown in fig The SubBytes transformation is done using a once-precalculated substitution table called S-box.

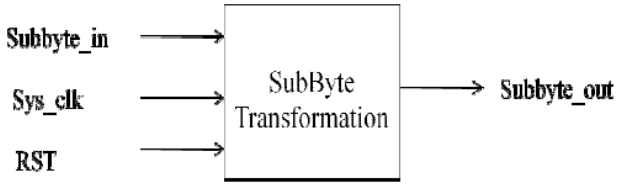

Figure.4. Subbyte Transformation

2. SHIFTROWS TRANSFORMATION

ShiftRows transformation includes, the rows of the state are cyclically left shifted. Row 0 remain unchange; row 1 does shift of one byte to the left; row 2 does shift of two bytes to the left and row 3 does shift of three bytes to the left. In ShiftRows transformation, the rows of the state are cyclically left shifted. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left as shown infig

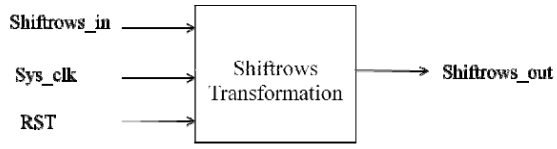

Figure.5. Shiftrow Transformation

3. MIXCOLUMNS TRANSFORMATION

MixColumns transformation includes, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by: c(x)={03}x3 + {01}x2 + {01}x + {02}. In MixColumns transformation, the columns of the state are considered as polynomials over GF (28). MixColumn operation performs column by column state, treating each column as a four-term polynomial over GF (28) .As a result of this multiplication, the new four bytes in a column are generated as shown in fig

Figure. 6. Mixcolumn Transformation

4. ADD ROUND KEY

Add RoundKey transformation includes, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation. The RoundKey for each round is derived from the main key using the Key Expansion algorithm. The encryption/ decryption algorithm needs eleven 128-bit RoundKey, which are denoted by RoundKey[0] to RoundKey[10].In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation - by a simple bitwise XOR operation shown in figure 5. The RoundKey of each roundis derived from the main key using the Key Expansion algorithm. The encryption/ decryption algorithm

needs eleven128-bit RoundKey, which are denoted RoundKey [0] RoundKey[10].

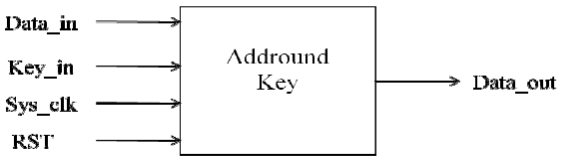

Figure.7. Addround Transformation

B. AES decryption

In decryption mode, the operations are in reverse order compared to their order in encryption mode. Thus it starts with an initial round, followed by 9 iterations of an inverse normal round and ends with an AddRoundKey. An inverse normal round consists of the following operations in this order: AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes. An initial round is an inverse normal round without the InvMixColumns.

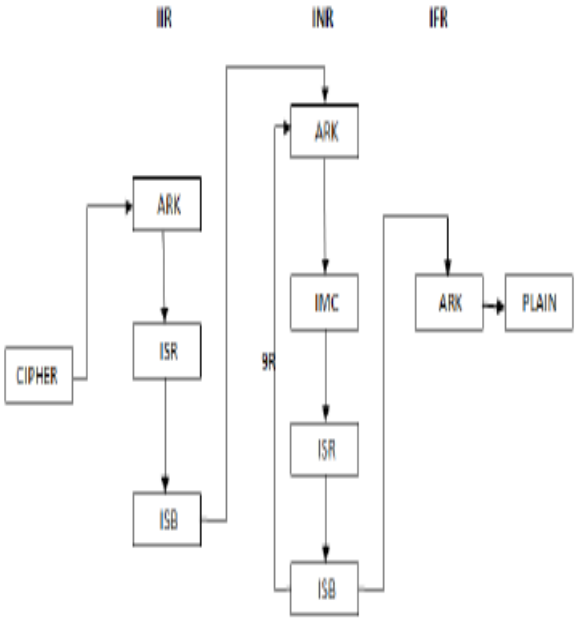

Figure. 8. General structure of Decryption.

Reverse of encryption which inverses round transformations to compute the original plaintext from cipher-text in reverse order called as decryption. The rounds of transformation of decryption use the functions AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes successively as shown in fig. 1.

Result:

Input data:- Two One Nine Two

Input data in Hex:-54 77 6F 20 4F 6E 65 20 4E 69 6E 65 20 54 77 6F

Key:- Thats my Kung Fu

Key in Hex:-54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

Simulation:-

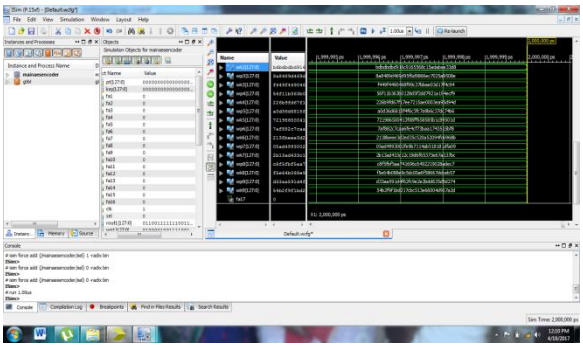

Figure. 9. encryption simulation

V. CONCLUSION

Efficient implementation of AES algorithm is exhibited in this paper. High throughput is accomplished in this design. Results are compared with previous reported designs result to show efficiency. Simulation of AES algorithm is done on ModelSim software and implemented on Xilinx XC3S500E Spartan-3E FPGA kit. Encryption algorithm is being used by military and government over a last couple of decades for secure communication. The main purpose of encryption is to hide data from unauthorized usage. In this paper, we purposed a method to employ the crypto processor run in integration with a General Purpose Processor.

VI. REFERENCES

[1].Hoang Trang and Nguyen Van (2012), An efficient FPGA implementation of the Advanced Encryption Standard algorithm IEEE 978-1-4673-0309-5/12.

[2].WANG Wei, CHEN Jie & XU Fei (2012), An Implementation of AES Algorithm Based on FPGA, IEEE 978-1-4673-0024-7/10.

[3].Yang Jun Ding Jun Li Na Guo Yixiong (2010), FPGA based design and implementation of reduced AES algorithm, IEEE 978-0-7695-3972-0/10.

[4].Mr. Atul M. Borkar, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare, "FPGA Implementation of AES Algorithm", International Conference on Electronics Computer Technology (ICECT), pp. 401-405, 2011 3rd.

[5].Chalermwat Thanavijitpun, Khanob Thongkhome, and Somsak Choomchuay, "FPGA Implementation of FOE-Portable hard disk System", "The Int. Conf on Information and Communication Technology for Embedded Systems, Pattaya, Thailand, January2011