Design and Implementation of Encryption Module for AES Core Using Verilog.

Sirasala Tejaswi[1], K. Kishore Kumar[2]
[1]PG Scholar, Dr.K.V.S.R. Institute of technology, AP, India,
[2]Assistant Professor, Dr.K.V.S.R. Institute of technology, AP, India,

Abstract: The National Institute of Standards and Technology (NIST) announced Rijndael as the new Advanced Encryption Standard (AES). The Predecessor to the AES was Data Encryption Standard (DES) which was considered to be insecure because of its vulnerability to brute force attacks. DES was a standard from 1977 and stayed until the mid-1990. However, by the mid-1990s, it was clear that the DES's 56- bit key was no longer big enough to prevent attacks mounted on contemporary computers, which were thousands of times more powerful than those available when the DES was standardized. High throughput architecture is proposed for an efficient implementation of Advanced Encryption Standard (AES) algorithm. The AES is a 128-bit Symmetric Block Cipher. This thesis includes the complete step by step implementation of Advanced Encryption Technique, i.e, encrypting and decrypting 128-bit data using the AES and its modification for enhanced reliability and security. The encryption process consists of the combination of various classical techniques such as substitution, rearrangement and transformation encoding techniques. The encryption and decryption modules include the Key Expansion module which generates Key for all iterations. The modifications include the addition of an arithmetic operation and a route transposition cipher in the attacks iterative rounds. The Key expansion module is extended to double the number of iterative processing rounds in order to increase its immunity against unauthorized attacks.

Key words- Throughput, Encryptor, Decryptor.

I. INTRODUCTION

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. It was ratified as a federal standard in May 2002. AES is the most recent of the four current algorithms approved for federal us in the United States. One should not compare AES with RSA, another standard algorithm, as RSA is a different category of algorithm. Bulk encryption of information itself is seldom performed with RSA.RSA is used to transfer other encryption keys for use by AES for example, and for digital signatures. AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. 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.

II. METHODOLOGY

A. Two main processes of AES encryption algorithm

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. 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 selectesNb bits of round key from the expanded key module.Round Transformation involves four modules by ByteSubstitution, ByteRotation,

B. MixColumn and AddRoundKey.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, which is shown as Fig1.

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}

Fig1. The state matrix.

In the four transformation modules of round transformation, the ByteRotation, MixColumn and AddRoundKey are all linear transformations except the ByteSub. Take analysis of the AES algorithm principle and we can find, ByteSubstitution operation simply replaces the element of 128-bit input plaintext with the inverse element corresponding to the Galois field GF (28), whose smallest unit of operation is 8 bits/ group.ByteRotation operation takes cyclic shift of the 128-bit state matrix, in which one row (32 bits) is taken as the smallest operand. MixColumns operation takes multiplication and addition operations of the results of ByteRotation with the corresponding irreducible polynomial x8 + x4 + x3 + x + 1 in GF(28), whose minimum operating unit is 32 bits.Addroundkey operation takes a simple XOR operation with 8-bit units.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 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 Fig2

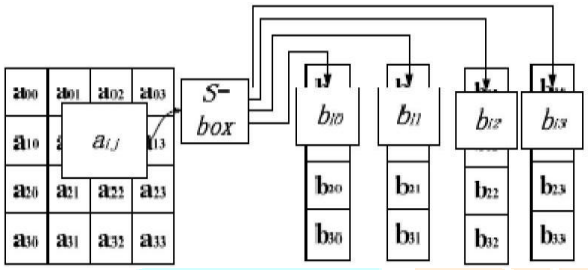

Fig3. Bytes segmentation and replacement processing.

Therefore, the original 128-bit input of plaintext and key will be replaced with four consecutive 32-bit input sequences respectively. In order to decrease the output ports, four continuous 32-bit cipher text 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.

C. The Process of New algorithms

From the above analysis, we can find that the process of AES encryption 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: Keyexpansion and Keyselection, while the plaintext is to be sent to the 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 Fig. 4. The functions of various parts of the structure shown above are described as follow:

The initial round of encryption: The four packets of consecutive 32-bit plaintext (128 bits) 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 operators.

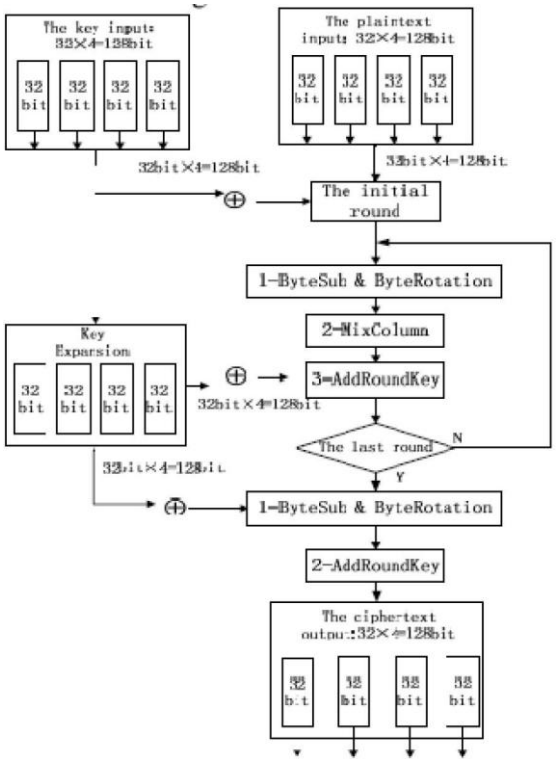

Fig4. The new improved structure of AES algorithm

Round Transformation in the intermediate steps: A round transformation mainly realizes the function of SubBytes and MixColumns with 32-bit columns. Four packets of round transformation are processed independently. Then the results of MixColumns and the 32-bit keys sourced from Keyexpansion are combined by using XOR operators. 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 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=1024bit). The implementation of MixColumn is mainly based on the mathematical analysis in the Galois field GF (2). 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 can be achieved. Fig.5 is a block diagram for the introduction of pipelining technology used in the round transformation.

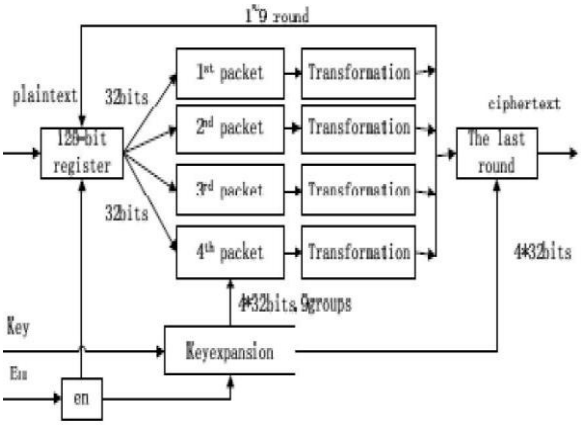

Fig5. The round processing with pipeline technology.

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, andcontrol the clock for re-starting 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 cipher text.

The process of the last round: The final round is a 128-bit processor. After nine rounds of operations included Shift rows, SubByte and Mixcolumns, the 128-bit intermediate encrypted data will be used in XORoperation with the final expanded key(4*32bit), which is provided by the key expansion module. The output of final round in the processor is the desired 128-bit cipher text. Similarly, the ciphertext is divided into four packets of 32- bit data by an external enable signal.

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. Theinitial 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 expandedkeys are stored into the registers at the same time. This implementation method is independent on a specific FPGA.

III. RESULTS

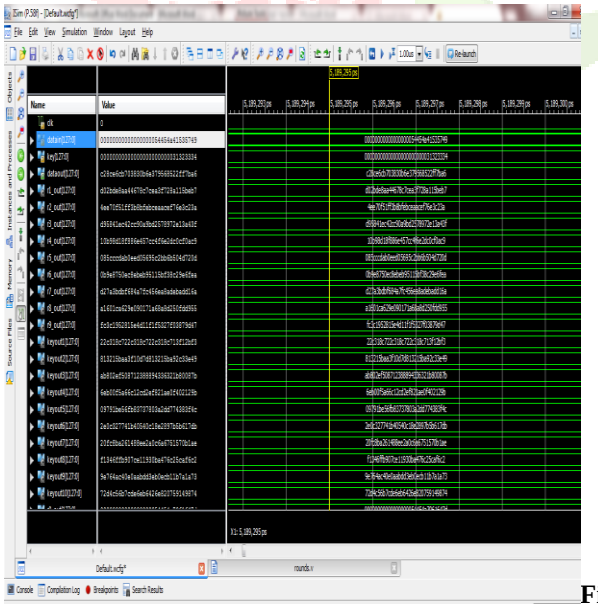

Fig6. Simulation waveform for encryption

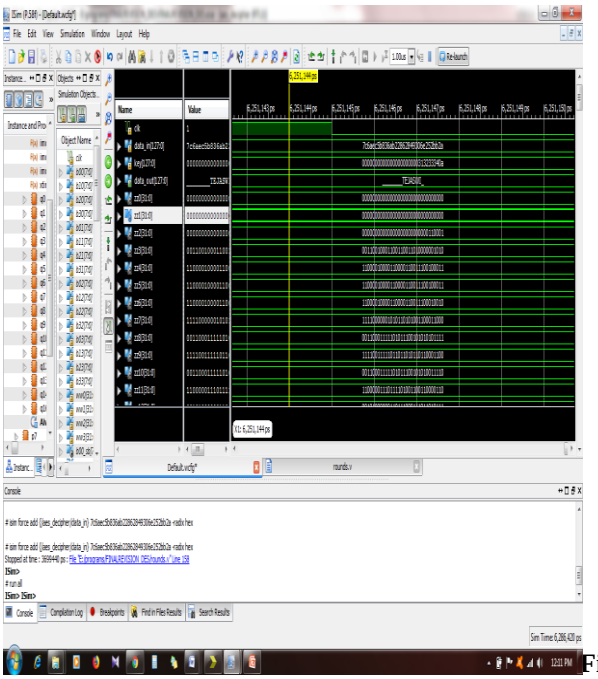

Fig7. Simulation waveform for Decryption.

Area report for Encryption

Selected Device : 3s500efg320-4

Number of Slices:13758 out of 4656 295% (*)
Number of Slice Flip Flops:128 out of 9312 1%
Number of 4 input LUTs:26999 out of 9312 289% (*)
Number of IOs: 385
Number of bonded IOBs: 385 out of 232 165% (*)
Number of BRAMs: 10 out of 20 50%
Number of GCLKs: 1 out of 24 4%

Area report for Decryption

Selected Device : 3s500efg320-4
Number of Slices: 16243 out of 4656 348% (*)
Number of Slice Flip Flops: 144 out of 9312 1%
Number of 4 input LUTs: 31226 out of 9312 335% (*)
Number of IOs: 385
Number of bonded IOBs: 385 out of 232 165% (*)
Number of BRAMs: 10 out of 20 50%
Number of GCLKs:1 out of 24 4%

RTL Diagram:

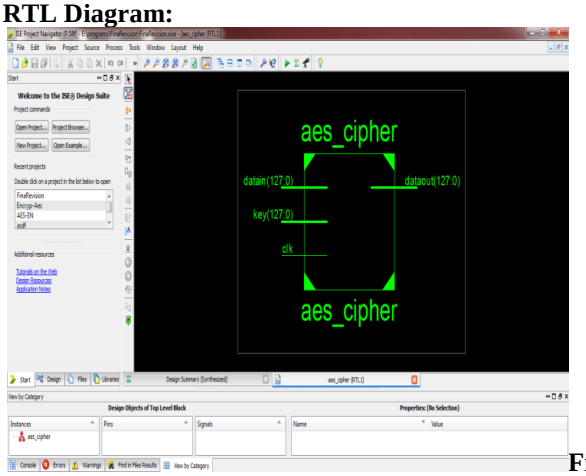

Fig8.

RTL Diagram for Encryption.

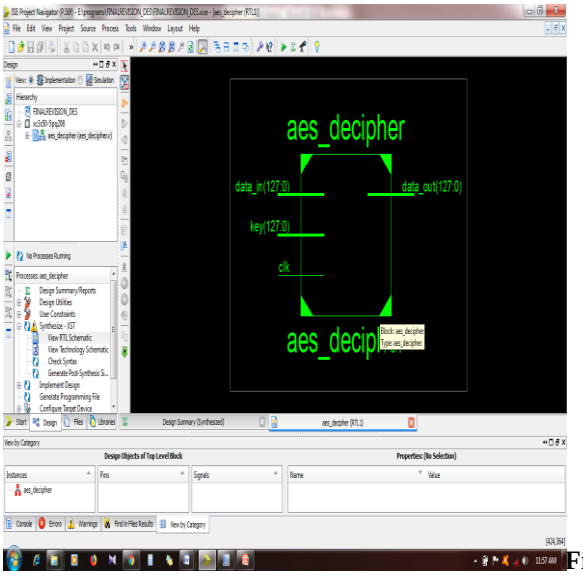

Fig9. RTL Diagram for Decryption

IV. CONCLUSION

The project "Design And Implementation of Encryption Module for AES Core Using Verilog" is successfully designed and simulated which consumes very low time and occupies less memory compared to other encryption algorithms. It is also highly secure. The AES core implementation using Rijandael algorithm for 128-bit data block with its far better results than its other opponents encourages the cryptographic standards to focus on to even larger data blocks like 192 bits, 256 bits of plain text with approximately selected key lengths, inorder to achieve the best cryptographic algorithm so that adversaries, intruders are kept away from sensitiveinformation.

V. REFERENCES

[1] Mr. Atul M. Broker, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare "FPGA Implementation of AES Algorithm" –2011 IEEE, 401-405

[2] Marko Mali, Franc Novak and Anton Biasizzo "Hardware Implementation of AES Algorithm" –Journal of ELECTRICALENGINEERING, Vol. 56, No. 9-10, 2005,265-269.

[3] ShraddhaSoni, HimaniAgrawal and Dr. (Mrs.) Monish Sharma, "Analysis and Comparison between AES and DES Cryptographic Algorithm", International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 6, December 2012

[4] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[5] L. Thulasimani,"A Single Chip Design and Implementation of AES- 128/192/256Encryption Algorithms"-International Journal of Engineering Science and Technology, Vol.2 (5), 2010, 1052-1059.

[6]Hiremath.S. And Suma.M.S.,"Advanced Encryption Standard Implemented on FPGA" IEEE Inter.Conf. Comp ElecEngin.(IECEE),vol.02,issue.28,pp.656-660,Dec.2009.

[7]Abdel-hafeez.S.,Sawalmeh.A.andBataineh.S.,"High Performance AES Design using Pipelining Structure over GF(28)" IEEE Inter Conf.SignalProc and Com.,vol.24-27, pp.716-719,Nov. 2007.

[8] Alan Kaminsky, Michael Kurdziel, Stanisław Radziszowski, "An overview of cryptanalysis research for the advanced encryption standard", IEEE, the 2010 military communications conference - unclassified program - cyber security and network management, pp. 1310, 2010.

[9] F.X.Standaert, "A Methodology to implement block ciphers in reconfigurable hardware and as application to fast and compact AES Rijndael. "The field programmable logic array conference, Monterey, California, pp.216- 224. 2003.