Volume 2, No. 2, Mar-Apr 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

An Implementation of AES-128 Bit Encryption and Decryption Algorithm using Field Programmable Gate Array

S.G.Sindhuja*
Department of Electrical and Electronics Engineering,
Noorul Islam University,
Kumaracoil, Tamilnadu,India.
atthikka@yahoo.com

Mrs.M.P.Flower Queen
Department of Electrical and Electronics
Engineering,Noorul Islam University ,
Kumaracoil, Tamilnadu,India.
flower_aurther@rediffmail.com

Abstract::In today's world most of the communication is done by using electronic media. Data security plays a vital role in such communication. Hence, there is a need to protect data from many attacks. In this paper, to investigate hardware implementation of AES-128 cipher standard on FPGA technology. In many network applications software implementations of cryptographic algorithms are slow and inefficient. To solve that problems custom architecture in reconfigurable hardware was proposed to speed up the performance and flexibility of AES algorithm implementation. The investigations involved simulations and synthesis of VHDL code utilizing Xilinx's ISE 6 with the target device Spartan-II .The main aim is to simulate the AES using Field Programmable Gate Array (FPGA) to achieve low cost, ease of implementation, high flexibility including capability of frequent modifications of hardware, and low cost of the final product. The proposed design consumes less power and area which is suitable battery driven mobile phones. NIST has selected Rijndael as the new Advanced Encryption Standard (AES).The proposed architecture gives an reduction in area and increase in speed (throughput), reduces power consumption.

Keywords: Advanced Encryption Algorithm, FPGA, VHDL, encryption, decryption.

I. INTRODUCTION

At present a majority of computer and telecommunication systems requires data security when data is transmitted over network. Thus data encryption is performed to protect sensible data. Usually appropriate software algorithm is used for coding data at sender site and decode at receiver one. Such a solution is not adequate and too slow then high speed processing is necessary due to high transmission medium bandwidth and real time requirements [1] .In such situation increase of computational platform processor performance is necessary, though usage of faster general purpose processor is not effective. That is why hardware acceleration of cryptographic algorithms is necessary. The very good solutions of choice for such dedicated hardware are reconfigurable devices. For integer based data this technology guarantee better performance and lower power demand. Additionally because of the distinguished features of hardware solution it also provides better data security against crackers' attacks.

The proposed work is to develop a hardware architecture that can be reconfigured key with a 128-bit data input and 128-bit data output is developed [2] . In this paper the proposed work in which encryption and process is explained with the flow of steps along with their algorithms which are iteratively used for encrypting and decrypting and the results along with discussions are presented [3]. In this paper dedicated hardware is proposed for AES algorithm. The proposed design consumes less power and area which is suitable battery driven mobile phones.

II. THE ADVANCED ENCRYPTION STANDARD ALGORITHM (AES)

AES is a symmetric block cipher that can encrypt and decrypt information. The AES is capable of cryptographic keys of 128,192 or 256 bits. Other input, output, cipher key lengths are not permitted by this standard. For a long period of time, the Data Encryption Standard (DES) was considered a standard for the symmetric key encryption. DES has a key length of 56 bits [2]. For the time being, this key length is considered small and can easily be broken.

For this reason, the National Institute of Standards and Technology (NIST) announced as a result of computation among 15 algorithms that the Rijndael cipher will replace the DES cipher and will become a new AES [16]. The Rijndael cipher has three possible block and key lengths: 128, 192, or 256 bits. Therefore, the problem of breaking the key becomes more difficult.

In general, hardware implementations of encryption algorithms and their associated key schedules are physically secure, as they cannot easily be modified by an outside attacker. The basic block diagram of encryption module is shown in below fig. The decryption function is similar to that of the encryption function except that the keys have to be read in reverse order, they must be calculated prior to applying any input, therefore they are stored in a stack like buffer [9].

The VHDL implementation of AES encryption and Decryption module is shown in fig.1 and fig.2

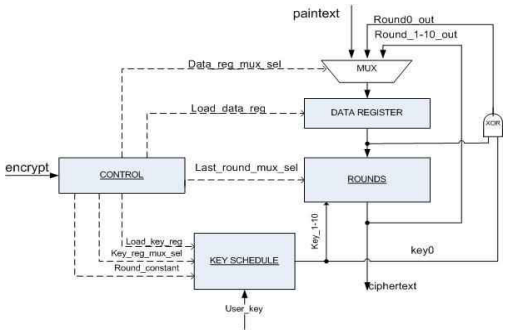

Figure.1: Basic block Diagram of Encryption Module

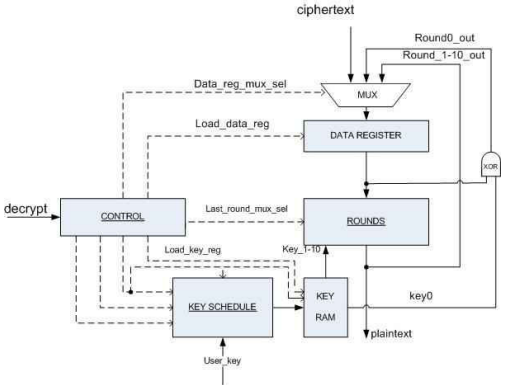

Figure.2: Block Diagram of decryption Module

III TRANSFORMATIONS USED IN AES

The Advanced Encryption Standard (AES) is a block cipher. The algorithm may be used with the three different "flavors" may by referred to as "AES-128", "AES-192", and "AES-256". The input to each round consists of a block of message called the state and the round key [2] . It has to be noted that the round key changes in every round. The state can be represented 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 same could be applied to the cipher key. The number of columns of the cipher key is denoted by Nk and is equal to the key length divided by 32. The cipher consists of a number of rounds - that is denoted by Nr - which depends on both block and key lengths [14].

Each round of AES encryption function consists mainly of four different transformations:

A. ByteSub Transformation:

The ByteSub transformation is a non-linear byte substitution, operating on each of the state bytes independently. The ByteSub transformation is done using a once-pre-calculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values.

B. Shift Row Transformation:

In ShiftRow transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted over one byte; row 2 is shifted over two bytes and row 3 is shifted over three bytes.

C. Mix-Column Transformation:

It operates on each column individually. It takes all the columns of the state and mixes their data to produce new column.

D. Add Key Transformation :

The round key is applied to the state - resulted from the operation of the Mix- Column transformation - by a simple bitwise X-OR. The round key length is equal to the block length. Each Round Key consists of *Nb* words from the key schedule. Those *Nb* words are each added into a column of the state.

The output of the above transformations is called the 'state'. The state consists of the same byte length as each block of the message.

Decryption process is performed according reversed encryption scheme although mentioned earlier Trans formations have different definition to be complementary to encryption transformations. Only Add RoundKey is identical for encryption and decryption. Also the same keys are used for rounds in decryption as they were use in encryption. Such for decryption round the following transformations are in use: InvByteSub Transformation, InvShiftRow Transformation and InvMix-Column Transformation.

The below fig. shows the modified implementation of AES Encryption and Decryption Module.

IV. IMPLEMENTATIONOF AES ENCRYPTION AND DECRYPTION MODULE

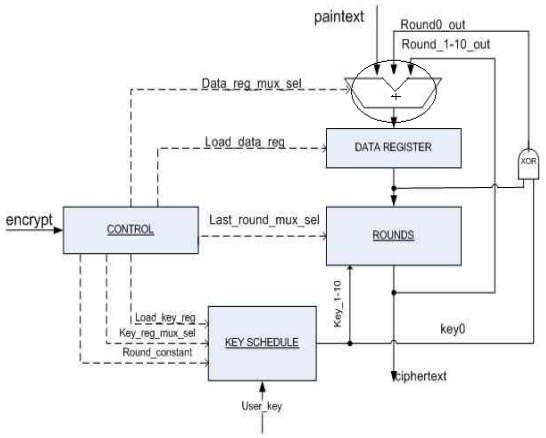

Figure.3: Modified block Diagram of Encryption Module

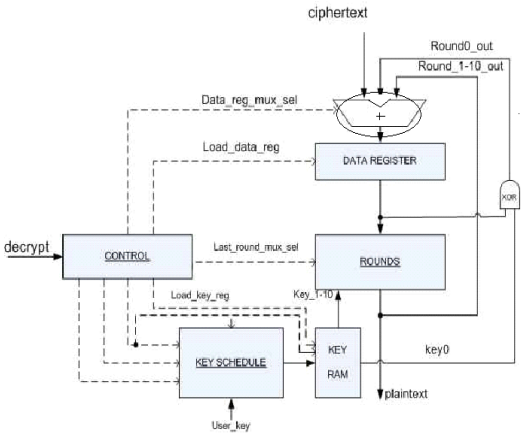

Figure.4: Modified block Diagram of Decryption Module

In the modified block diagram instead of multiplier we use shift-and-add multiplier. So it is mainly used to reduce the power consumption nearly 75% and increase the speed [10].

A. The Controller:

This block controls the sequencing operations of the rounds. It generates the round constant associated with each round. It also generates the control signal at the appropriate time, those control signals are:

Key_reg_mux_sel: The key schedule input selector which selects the input key before the first round, and then it reads the output key as the next input.
Load_key_reg: The key schedule output enable.
load_data_reg: the input registers enable.

data_reg_mux_sel: The data register selector which selects input register data from the plaintext input, last_mux_sel: it goes to the round layer to indicate the last round. *round_constant:* it goes to the key schedule layer to indicate which round is running at the moment and it applies a constant value according to the running round.

B. Key Generator:

This block is responsible for creating the keys for each round. It contains four S boxes.

C. *The Round Layers*:

It contains the row shifter followed by two blocks running in parallel. The first one is Mix-Column block combined with an S-Box block, and the second one is S-Box block. The first block calls four look-up tables to calculate its output. The second block is the normal S-Box block. The final output is taken from S-Box combined with Mix-Column block .

V. OUTPUT OF THE ENCRYPTION FUNCTION

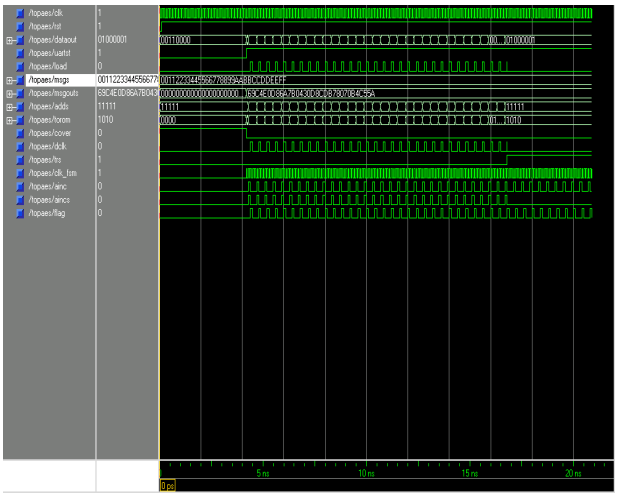

Figure: 5

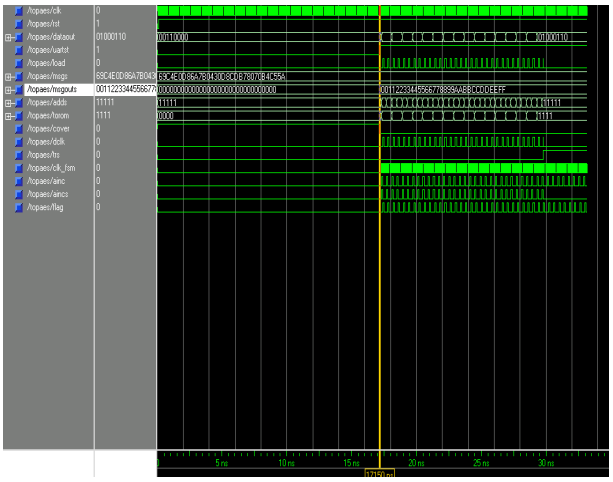

Figure: 6

VI. CONCLUSION

In this paper we approved a modified implementation of AES encryption and Decryption standard based on the fact that any FPGA includes built in memory block; therefore we store all the results of the fixed operations within the memory modules. The modification gives an reduction in area and increase in speed (Throughput) compared with the original design. This design gives the highest throughput and area utilization over all the Iterative Looping based FPGA implementations.The combination of a simple, portable and efficient AES cryptographic algorithm implemented in VHDL source code provides an excellent platform for high security applications.A synthesizable VHDL code is developed for the implementation of both encryption and decryption process. So, overall the proposed architecture can reduces the power consumption nearly 75% as compared to the conventional algorithm and architecture.

VII. REFERENCES

[1] Alina Olteanu, Yang Xiao, Yan Zhang:Optimization Between AES Security and Performance for IEEE 802.15.3 WPAN, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS,DECEMBER 2009,VOL. 8, NO. 12

[2] J. Daemen, V.Rijmen: The Rijndael Block Cipher: AES Proposal : First AES Candidate Conference (AES1) : August 20-22, 1998

[3] A. Dandalis, V.K. Prasanna, J.D.P. Rolim: A Comparative Study of Performance of AES Candidates Using FPGAs: The Third Advanced Encryption Standard (AES3) Candidate Conference, 13-14 April 2000, New York, USA.

[4] N. Sklavos; G. Selimis and O. Koufopavlou; "FPGA Implementation Cost and Performance Evaluation of IEEE 802.11 Protocol Encryption Security Schemes", Journal of Physics: Conference Series 10 (2005), pp. 361- 364.

[5] X Zhang, and K.K. Parhi, "High-speed VLSI architectures for the AES algorithm". IEEE Trans. VLSI Syst., 2004, 12, (9), pp. 957,967.

[6] H. Satyanarayana, AES128, OpenCores, Online, AvailablefromInternet, <opencores.com/projects.cgi/ web/ aes_crypto_core/overview>, 2004, accessed 1.10.07.

[7] Chih-Chung Lu and Shau-Yin Tseng, "Integrated Design of AES (Advanced Encryption Standard) Encrypter and Decrypter," Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, Processors, (ASAP'02), pp. 277-285, 2002

[8] CP. Su, TF. Lin, CT Huang and Cw. Wu, "A high-throughput low-cost AES processor", IEEE Commun Mag. 42 (2003) (12), pp. 86--91.

[9] Xinmiao Zhang and Keshab K. Parhi, "Implementation Approaches for the Advanced Encryption Standard Algorithm," IEEE Circuits and systems Magazine, vol. 2, no. 4, pp. 24–46, 2003

[10] Xinmiao Zhang, Keshab K. Parhi. High-Speed VLSI Architectures for the AES Algorithm. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, 2004 (9): 957-967.

[11] G. Rouvroy, F. Standaert, J. Quisquater and J. Legat, "Compact and Efficient Encryption/Decryption Module for FPGA Implementation of the AES Rijndael VeryWell Suited for Small Embedded Applications", Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), IEEE 2004.

[12] Dennis Ka Yau Tong, Pui Sze Lo, Kin Hong Lee, Philip H.W. Leong, "A System Level Implementation of Rijndael on a Memory-slot based FPGA Card", Proceedings of the 2002 IEEE International Conference on Field Programmable Technology (FPT), Hong Kong, pp. 102-109, 2002

[13] Xilinx. Virtex-E 1.8 V Field Programmable *Gate Arrays*. Product Specification, Sep 2002.

[14] A. Elbirt, W. Yip, B. Chetwynd, and C. Paar.An FPGA-based performance evaluation of theAES block cipher candidate algorithm finalists.IEEE Transactions on Very Large Scale Integration(VLSI) Systems, pages 545–557, Aug 2001.

[15] A. Daly and W. Marnane. Efficient Architectures for implementing Montgomery Modular Multiplication and RSA Modular Exponentiation on Reconfigurable Logic. International Symposium on Field Programmable Gate Arrays, pages 40–49, Feb 2002.

[16] K. Gaj, P. Chodowiec: Comparison of the Hardware Performance of the AES Candidates using Reconfigurable Hardware: The Third Advanced Encryption Standard Candidate Conference, April 13-14, 2000, New York, USA.