Paper ID: VESCOMM11

AN ADVANCED ENCRYPTION STANDARD WITH RTL SCHEMATIC DESIGN

Ms. Sarika N. Wagaj
Dept of Electronics & Telecommunication
VVPIET Solapur, India.
wagajsarika9@gmail.com

Mr. Sajid Shaikh
Dept of Electronics & Telecommunication
VVPIET Solapur, India.
sajid0077@gmail.com

Abstract—The Advanced Encryption Standard (AES) algorithm isdefault choice for various security services in various applications. This encryption implementation will do through VLSI platform. In this architecture we are deal with ROM module in FPGA. AES are presents a low area and low power hardware architecture for the data transmission. In this algorithm there are four stages, in that four stages for first experimental parameter we are select merging of two stages i. e. sub byte transformation and shift row and second experimental parameter is mix column stage. Designing of S-box is more important in AES algorithm. This architecture can be used in many military, industrial, and commercial applications that require compactness and low cost.

KEYWORDS - AES ENCRYPTION, ROM SUBMODULES,LOW POWER CONSUMPTION.

II. INTRODUCTION

Messages or information in such a way that only authorized parties can read it. Encryption doesn't prevent hacking but it reduces the likelihood that the hacker will be able to read the data that is encrypted. In an encryption scheme, the message or information, referred to as plaintext, is encrypted using an encryption algorithm turning into an unreadable text. This is usually with the help of an encryption key, which specifies how the message is to be encoded. Any adversary that can see the ciphertext should not be able to determine anything about the original message. An authorized party, however, is able to decode the cipher text using a decryption algorithm, that usually requires a secret decryption key, that adversaries do not have access to. An encryption scheme usually needs a key-generation algorithm to randomly produce keys. With the increasing proliferation

of images, videos and other multimedia data over the unsecured network, such as Internet, there is a serious need to encrypt those, so as to provide the security and privacy. So the importance of encryption arises in this scenario. Encryption transforms information such that its true meaning is hidden and requires special knowledge to retrieve the information after decryption. Although, AES can be implemented in software, its hardware implementation provides high speed with added physical security [1]. Hardware implementation can have different approaches like very high, medium and low throughput architectures trading-off area for speed [2]. AES is a private key encryption algorithm consisting of following transformations-SubBytes, ShiftRows, MixColumns and AddRoundKeys [3]. AES encrypts data in blocks of 128 bits. It can accept three key sizes, 128-bit, 192-bit and 256- bit, but generates 128-bit round key for XORing with 128- bit data in the AddRoundKeys step. The number of rounds for 128-bit keys, 192-bit keys and 256-bit keys are 10, 12 and 14 respectively[4]. Fig.1 shows the encryption steps and number of rounds involved for 128-bit key size. In this paper, an efficient VLSI architecture for AES encryption of key size 128-bit for medium throughput applications, like simultaneous image compression and encryption is proposed. The proposed architecture has a medium throughput and low power consumption as compared to existing AES architecture. The remaining content of the paper is organized as follows. Section II describes the AES architecture. Section III describes the proposed architecture of AES Encryption. Section IV dis-cusses the hardware implementation results and comparison. Conclusion has been shown in Section V.

III. AES ENCRYPTION

AES is a private key encryption algorithm consisting of following transformations-SubBytes, ShiftRows, MixColumns ,andRoundkeys.

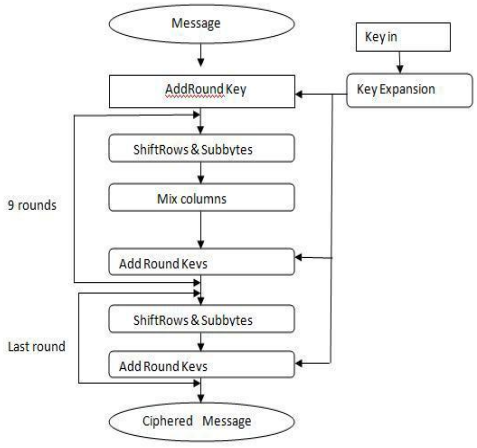

Fig. 1. Proposed Steps of AES Encryption

AES encrypts data in blocks of 128 bits. It can accept three key sizes, 128-bit, 192-bit and 256-bit, but generates 128-bit round key for XORing with 128-bit data in the AddRoundKeys step.

The number of rounds for 128-bit keys, 192-bit keys and 256-bit keys are 10, 12 and 14 respectively. Fig.1 shows the encryption steps and number of rounds involved for 128-bit key size. For encryption, each round consists of the following four steps:

1.SubBytes (SB): a bytewise transformation that applies on each byte of the current block an 8-bit to 8-bit nonlinear S-box

2. ShiftRows (SR): a linear operation that rotates on the left all the rows of the current matrix (0 for the first row, 1 for the second, 2 for the third and 3 for the fourth).

3.MixColumns (MC): another linear operation represented by a 4 x 4 matrix. Each column of the input matrix is multiplied by the MixColumns matrix in GF(28).

4. AddRoundKey (AK): a simple XOR operation between the input matrix and the subkey of the current round.

The above steps are related to standard AES. In this paper we are introduce the merging of subbyte transformation and shift row. These are the take minimum space for performing operation of AES and those implementing stages don't compromise for security.

The MixColumns operation is omitted in the last round and an initial key addition is performed before the first round for whitening. Also the MixColumns operation is omitted in the last round of the reduced-round variants. The number of rounds is variable depending on the key length; 10 rounds for 128-bit key, 12 for 192-bit key and 14 for 256-bit key. The last step consists of XORing the output of the previous three steps with four words from the key schedule. For decryption, each round consists of the following four steps: (a)Inverse shift rows

(b) Inverse substitute bytes (c) Add round keys (d) Inverse mix columns.

The third step i.e,Addround keys consists of XORing the output of the previous two steps with four words from the key schedule. Note the differences between the order in which substitution and shifting operations are carried out in a de-cryption round and also the order in which similar operations are carried out in an encryption round.

IV. PROPOSED ARCHITECTURE

In this paper, some architectural modifications are introduced which enhances the total efficiency of the cryptosystem, and reduces area and power consumption. The proposed AES architecture for encryption is implemented in device using Xilinx Virtex-V FPGA board.

In this paper, another method introduced is the exclusion of Shift row operation. Exclusion of Shift Row is performed through calling required shifted element from the data ma-trix,(instead of calling element one by one sequential order from the data matrix); Thus merging of the two steps Sub-Byte and Shift Row reduces to one step. Since the step in the AES algorithm is reduced, it can be a reason and way to reduce the area and power consumption of the total hardware. Hence the architecture changes into another level which enhances the total efficiency of the system. The design flow chart is given in Fig.1.

In this work, the round keys generated are stored in ROM modules, rather than in registers. Key expansion procedure generates 11 round keys (each of 128 bits) including the initial input key. It can be stored in registers. In the VLSI design, a single bit register takes higher area as compared to single bit ROM. So, the proposedarchitecture employs ROM to store the 10x128-bit Round keys. Fig..3 shows the round key storage module in ROM. There are 40 ROM sub-modules used for generation of 10 round keys. Each ROM consists of 8-bits in four locations. The RoundKey generator generates 4 bytes of key in a single clock cycle. A total of 16 bytes of round keys are required for the AddRoundKeys module, which are provided in four clock cycles.

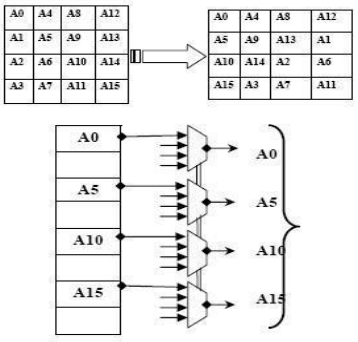

Fig. 2. Shifted elements calling through MUX

Table I: Comparison in available and proposed architecture

Hardware consumption	Available Architecture	ROM based Proposed Architecture
Slices	4478	1586
LUT's	8565	1287
Minimum clock	4.838ns	3.782ns
Throughput	1.202Gbps	1Gbps
Power Consumption(m) Table	478 mW	448 mW

Using above comparison we can say that propose architecture is more beneficial to us. It can be observed that the proposedarchitecture has high hardware efficiency in terms of throughput per slice.The power consumption results have been obtained fromthe XPower analyzer tool integrated in Xilinx ISE 14.5. It can be concluded that the proposed architecture consumes low power even at higher clock frequency than the one used.Baseline AES: It includes the baseline steps. The main steps as key Expansion and round Transformations. The Round Transformation includes AddRound Key, ShiftRows ,Simple SubBytes Transformation and Mixcolumn Transformation.

Composite AES: It includes all the steps same as above but it includes composite SubBytes Transformation. The following figures shows the device utilization summary of both Composite AES Encryption and Decryption.

V. IMPLEMENTATI AND SIMULATION

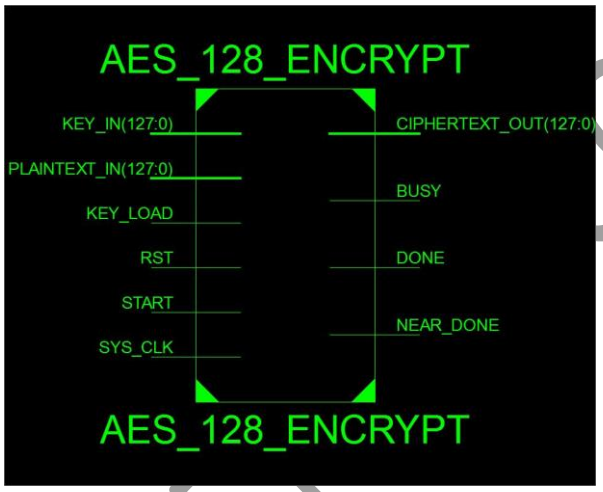

Fig. 3. The RTL schematic symbol Of AES

The proposed architecture is implementing in Virtex5 FPGA with Xilinx VHDL coding. The overall performance can be compared with the basic ROM based key generation method. The AES architecture is implemented in pipelining mode. Eachpipeline require clock cycles in the completion clock cycles are required in Sub Bytes transformation whereas a single byte is accessed from the S-box implemented in ROM. clock cycles are required in the mixing module where 4 bytes are XORed with the key obtained either directly from the key generator (in case of first block of data) or from the ROM in a single clock cycle. Round keys generated by the key generator are stored in 40 ROM modules. For the encryption of next block of data, 128-bit keys are provided from the ROMs in clock cycles.

Implements AES (Rijndael) to latest Designed specifically for ultra low resource applications – this is the very smallest hardware AES solution available. Data throughput up to 75Mbps .Full dynamic support for all AES key sizes (128, 192 and 256-bits) .Single core handles encryption, decryption, and hardware roundkey expansion. All AES operating modes easily implemented (eg. ECB, CBC, OFB, CFB, CTR) Simple external interface Highly optimised for use in each individual FPGA technology.

VI. CONCLUSION

The proposed architecture is efficient in terms of throughput per slice (area) and consumes less power when compared with existing architecture of medium throughput. The area reduction is done by storing the round keys in ROM instead of registers and hence area and power reduction is achieved through this method. The key expansion algorithm ensures that AES has no weak keys. A weak key is a key that reduces the security of a cipher in a predictable manner.

REFERENCES

[1] Data Encryption Standard in cryptography system, Panu Hämäläinen, TimoAlho, Marko Hännikäinen, and Timo D. Hämäläinen Tampere University of Technology / Institute of Digital and Computer Systems, Tampere Finland; May 5 2007.

[2] "Area and power optimization for AES encryption module implementation on FPGA" by Pham, Tuan Anh Fac. Of Comput., Eng. & Technol., Staffordshire Univ, Stafford,UKHasan, Mohammad Shahidul; Yu, Hongnain N.; September 2012.

[3] "Efficient implementation of AES algorithm on reconfigurable FPGA"; by Hrushikesh S Deshpande, Rady, Ahmed El Sehely, E.; International Conference on Microelectronics, 2014.

[4] "Design and implementation of area optimized AES algorithm on reconfigurable FPGA"paper at International journal of Computer Science and its Applications by M.SirinKumari ,D.Mahesh Kumar Y. Assoc.Prof, Rama

Devi at JITS Kanpur.
[5]National Institute of Standards Technology (NIST),
Recommendation for Block Cipher Modes of Operation -
Methods and Technologies, Dec. 2001.
[6] B. Jyrwa and R. Paily, "An Area-Throughput Efficient
FPGA imple-mentation of Block Cipher AES algorithm", IEEE International Confer-ence on
Advances in Computing, Control,nand Telecommunication Tech-nologies, pp. 328-332, 2009.
[7] S. Morioka and A. Satoh, "An Optimized S-Box Circuit
Architecture for Low Power AES
Design," in CHES '02: Revised Papers from the 4th International Workshop on Cryptographic
Hardware and Embedded Systems, pp. 172–186, Springer-Verlag, 2003.
[8] Y. Zhang and X. Wang, "Pipelined Implementation of
AES Encryp-tionBased on FPGA",
IEEE International Conference on Information Theory