International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

Design an Advanced Encryption Standard (Aes) Algorithm

Gaddam Amulya & Y.R.K. Paramahasa.
[1]M.tech-Scholar, Dept of ECE, Pace Institute Of Technology And Sciences, Ongole, A.P, India
[2]Associate Professor, Dept of ECE, Pace Institute Of Technology And Sciences, Ongole, A.P, India

ABSTRACT: *Now a days, VLSI application speed and area reduction is very important one. In this paper AES algorithm is implemented. AES represents an algorithm for advance encryption standard of different operation required in the steps of encryption and decryption. Advanced Encryption Standard (AES) is the most efficient public key encryption system which is based on Rijndael Algorithm that can be used to create faster and efficient cryptographic keys. The proposed architecture is based on optimizing area in terms of reducing and improve throughput for design of AES algorithm in VHDL. This paper presents AES-128 bit algorithm design consist of 128 bit symmetric key and XILINX ISE 14.1 project used for synthesis and simulation of this proposed design.*

Keyword: advance encryption algorithm (AES); VHDL; FPGA; encryption and decryption

I.INTRODUCTION

The internet plays a key role in day-to-day life. The people can transfer important data through the internet such as Email, banking transaction and online purchase. In order to acquire secured transaction, network security is most essential. Network security is mostly achieved through the cryptography. Cryptography refers to the art and science of transforming the message to provide them with secure and immune to attacks. Different algorithms and protocols are utilized to protect the data. In this paper, AES algorithm is implemented. AES is a cryptographic algorithm that is utilized for protecting electronic data or information. AES is a symmetric algorithm which process 128 bit stream in 10 rounds. It uses same key for encryption information. The AES algorithm input is applied, to perform number 10 rounds transformation and finally cipher is generated.

Millions of users interchange their information in different fields, like medical reports and bank services, financial and legal files via Internet. A cryptography technique is especially applicable and plays a major role for secure the data. This implementation will be useful in wireless security such as military communication and mobile telephony where there is a greater emphasis on the speed of communication. AES can be implemented in software or hardware. The hardware implementation is more suitable for high speed applications in real time. From last several years, Data Encryption Standard (DES) had been utilized as a cryptographic algorithm. DES is replaced by the Rijndael algorithm due to its short key length. A standard algorithm in the cryptography domain is Advanced Encryption Standard (AES).

II.EXISTED SYSTEM

AES Advanced Encryption algorithm has excellent cryptographic properties, which employs symmetrical structure to resist all known attacks. The algorithm has fast speed of encryption and decryption and strong anti-attack capability. As the only one non-linear element of high decryption algorithm, the S box determines the encryption strength and decryption speed of the algorithm. In addition to a good cryptographic properties, a good S-box also has little hardware resources consumption.

The number of S box used in the block ciphers is more, the nonlinearity of the cryptographic algorithms is higher, and the confusion of the cipher algorithm is stronger. In fact, the bigger S box, which is used in the hardware structure, the calculation, check list and storage are also

required more time and space. In this case, the algorithm becomes very low efficient. So how choose a good S-box, we need to consider the security of algorithm and the work efficiency of implementation. The traditional method generates the S box, which has a good index of cryptography. However, it is difficult to use pure logic hardware implementation consuming a large number of logic units. What's worse, the use of look-up table costs too much memory resource. Ideally, the method needs 16 clock cycles to complete the transformation, which is not conductive to the high-speed implementation of encryption system.

III.PROPOSED SYSTEM

The Advanced Encryption Standard (AES) is a symmetric block cipher used by the U.S. government to protect classified information and is implemented in software and hardware throughout the world for sensitive data encryption.

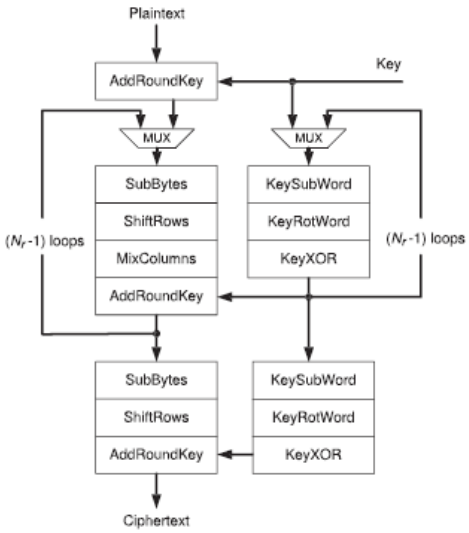

FIG. 1 PROPOSED BLOCK DIAGRAM

AES is actually, three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128 bits, 192 bits and 256 bits, respectively. In Advanced encryption standard there are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys

Each round in encryption process further follows some steps to complete each round till n. Each round possess four rounds.

1. Subbytes
2. Shiftrows
3. Mixcolumn
4. Addroundkey

A. Substitution round

In this step, Sub-Bytes are byte-by-byte substituted during the forward encryption process.

B. Shift Rows

In this step, shifting the rows of the state array during the forward process(S-Box process)

C. Mix Column

Mix Columns for mixing up of the bytes in each column separately during the forward process.

D. Add Round Key

In this step, round key is added to the output of the previous step during the forward process. This step differs from others because of key size difference.

AES Encryption

To implement the AES encryption algorithm, we proceed exactly the same way as for the key expansion, that is, we first implement the basic helper functions and then move up to the main loop. The functions take as parameter a *state*, which is, as already explained, a rectangular 4x4 array of bytes. The shiftRows function iterates over all the rows and then call shiftRow with the correct offset. shiftRow does nothing but to shift a 4-byte array by the given offset.

This is the part that involves the roundKey was generated during each iteration. Here simply XOR each byte of the key to the respective byte

of the state. The MixColumns implementation was carried out by first one would generate a column and then call mixColumn, which would then apply the matrix multiplication.

One AES round is the one which has to apply all four operations on the state consecutively. All we have to do is take the state, the Expanded Key and the number of rounds as parameters and then call the operations one after the other. Finally, all we have to do is put it all together. Our parameters are the input plaintext, the key of size keySize and the output. First, we calculate the number of rounds based on they keySize and then the expanded KeySize based on the number of rounds. Then we have to map the 16 byte input plaintext in the correct order to the 4x4 byte state (as explained above), expand the key using our key schedule, encrypt the state using our main AES body and finally un-map the state again in the correct order in order to get the 16 byte output cipher text.

AES Decryption

For the AES Decryption, the key schedule stays the same, the only operations we need to implement are the inversed sub Bytes, shift Rows and mix Columns, while addRoundKey stays the same.

As you can see, they are nearly identical to their encryption except that the rotation this time is to the right and that we use the inversed S-Box for the substitution. As for the inversed mixColumns operation, the only difference is the multiplication matrix is different. Finally, the only thing left to do is putting it all together in one inversed main algorithm. Please note that we use our expanded key backwards, starting with the last 16 bytes and then moving towards the start.

The separate modules were written for the Last Round and other Rounds. From first round to ninth round the same module can be instantiated and for the last round, a separate module was used since it doesn't have the MixColumns operation. The functional verification was carried out for all the test cases and hence the RTL modeling is taken to the synthesis process using the Xilinx tool.

IV.RESULTS

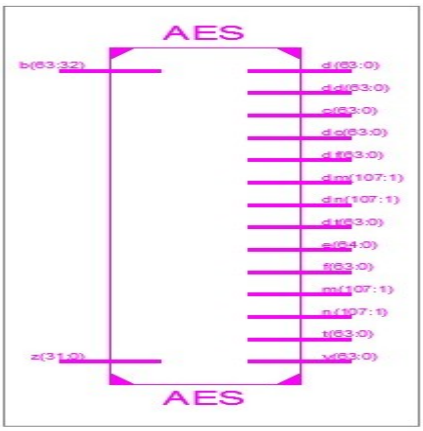

FIG. 2 RTL SCHEMATIC

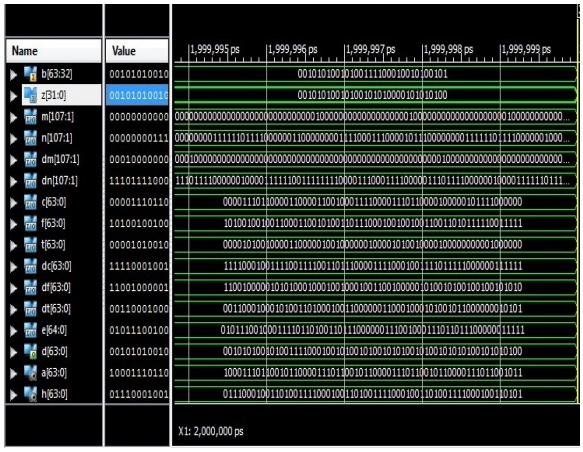

FIG. 3 OUTPUT

V.CONCLUSION

We have presented a VLSI architecture for the Rijndael AES algorithm that performs both the encryption and decryption. S-boxes are used for the implementation of the multiplicative inverses and shared between encryption and decryption. The round keys needed for each round of the implementation are generated in real-time. The forward and reverse key scheduling is implemented on the same device, thus allowing efficient area minimization. Although The implementation of the key unit in the proposed architecture, can be scaled for the keys of length 192 and 256 bits easily. 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. In this direction, we have presented a pipeline version of AES algorithm that can encrypt data.

VI.REFERENCES

[1] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, "A DCT Domain Visible Watermarking Technique for Images," in Proc of the IEEE International Conf on Multimedia and Expo, 2000, pp. 1029–1032.

[2] M. S. Kankanhalli and T. T. Guan, "Compressed-Domain Scrambler Descrambler for Digital Video," IEEE Transactions on Consumer Electronics, vol. 48, no. 2, pp. 356–365, May 2002.

[3] B. M. Macq and J. J. Quisquater, "Cryptography for Digital TV Broadcasting," Proceedings of the IEEE, vol. 83, no. 6, pp. 944–957, Jun 1995.

[4] H. Kuo and I. Verbauwhede, "Architectural Optimization for a 1.82 Gbits/sec VLSI Implementation of the AES Rijndael Algorithm," in Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, 2001, vol. 2162, pp. 51–64.

[5] M. McLoone and J. V. McCanny, "Rijndael FPGA Implementation Utilizing Look-up Tables," in Proceedings of the IEEE Workshop on Signal Processing Systems, 2001, pp. 349–360.

[6] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," in Proceedings of Advances in Cryptology - ASIACRYPT 2001, 2001, pp. 171–184.

[7] S. Mangard, M. Aigner, and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture," IEEE Transactions on Computers, vol. 52, no. 4, pp. 483–491, April 2003.

[8] T. Sodon O. J. Hernandez and M. Adel, "Low-Cost Advanced Encryption Standard (AES) VLSI Architecture: A Minimalist Bit-Serial Approach," in Proc of IEEE Southeast Conference, 2005, pp. 121–125.

GADDAM AMULYA completed her B.Tech at Qis College of Engineering and Technology, Ongole and pursuing M.Tech at Pace institute of technology and sciences, Ongole. Her area of interest is VLSI.

Y.R.K. PARAMAHASA completed his B.E at S.V.H college of Engineering, Machilipatnam and M.Tech at JNTUA. He has 29 years of teaching experience and at present he is working as Associate professor at Pace institute of technology and sciences, Ongole.