ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2278-2566
Vol.02, Issue.03
August -2019
Pages: -362-368

AN EFFICIENT VLSI ARCHITECTURE FOR DATA ENCRYPTION STANDARDS

1. UPPALAPATI RAJU, 2. GOUTHAM KRISHNA REDDY
1. M.Tech, Dept. of ECE, A.K.R.G COLLEGE OF ENGINEERING & TECHNOLOGY, Nallajerla, A.P
2. Assistant Professor, Dept. of ECE, A.K.R.G COLLEGE OF ENGINEERING & TECHNOLOGY, Nallajerla, A.P

Abstract: To achieve the goal of secure communication, cryptography is an essential operation. Many applications, including health-monitoring and biometric data-based recognition system, need short-term data security. To design short-term security-based applications, there is an essential need of high-performance, low cost and area-efficient VLSI implementation of lightweight ciphers. Data encryption standard (DES) is well-suited for the implementation of low-cost lightweight cryptography applications. In this paper, we propose an efficient VLSI architecture for DES algorithm-based encryption engine. In the implementation of DES algorithm, a chain of multiplexer-based architecture is used to implement the substitution operations (S Boxes). The proposed architecture is modeled in the Verilog design language and synthesized in the Xilinx software.

Keywords- DES encryption block cipher, lightweight cryptography, VLSI architectures, FPGA.

CRYPTOGRAPHY

Information security has become a top priority in many applications such as smart cards, Radio Frequency Identification (RFID) tags and sensor nodes. All these applications require the access of private information. For example, in electronic transactions, customers provide credit card or debit card numbers when making payments online. If the connection is insecure, hackers can easily access this sensitive information. Cryptographic algorithms are used to ensure confidentiality and it comes within one of two categories: symmetric-key and asymmetric-key. Symmetric-key algorithms use the same key for both encryption and decryption. Asymmetric-key algorithms use a public key for encryption and a private key for decryption. Block ciphers and stream ciphers are two types of symmetric-key algorithms. Block ciphers encrypt a block of data whereas stream ciphers encrypt the data bit by bit. In cryptology, block ciphers are one of the most important primitives. A block cipher consists of two closely related algorithms, block encryption and block decryption algorithms. Block encryption algorithm takes an input block of fixed-size known as the plaintext and converts it into another block of the same size known as the cipher text. This block encryption and decryption takes place under the action of a fixed secret key that may not have the same size of the plaintext. A block cipher must be invertible i.e., by using the block decryption algorithm it should be possible to recover the original plaintext from the cipher text and the secret key. Once the plaintext has been encrypted using a given key, a successful decryption can be performed only by knowing the secret key. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality. Many have been thoroughly broken. Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher; see Category: Stream ciphers. Block ciphers can be used as stream ciphers; see Block cipher modes of operation. Cryptographic hash functions are a third type of cryptographic algorithm. They take a message of any length as input, and output a short, fixed length hash which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash.

DATA ENCRYPTION STANDARD: The Data Encryption Standard (DES) is an outdated symmetric-key method of data encryption. DES works by using the same key to encrypt and decrypt

a message, so both the sender and the receiver must know and use the same private key. Once the go-to, symmetric-key algorithm for the encryption of electronic data, DES has been superseded by the more secure Advanced Encryption Standard (AES) algorithm.

To encrypt a plaintext message, DES groups it into 64-bit blocks. Each block is enciphered using the secret key into a 64-bit ciphertext by means of permutation and substitution the process involves 16 rounds and can run in four different modes, encrypting blocks individually or making each cipher block dependent on all the previous blocks. Decryption is simply the inverse of encryption, following the same steps but reversing the order in which the keys are applied. The length of the key determines the number of possible keys -- and hence the feasibility of this type of attack. DES uses a 64-bit key, but eight of those bits are used for parity checks, effectively limiting the key to 56-bits. Hence, it would take a maximum of 2^56, or 72,057,594,037,927,936, attempts to find the correct key. To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper. Although the rounds are not aligned, the elements (mixer or swapper) are aligned. We proved that a mixer is a self-inverse; so is a swapper. The final and initial permutations are also inversing of each other.

The left section of the plaintext at the encryption site, L0, is enciphered as L16 at the encryption site; L16 at the decryption is deciphered as L0 at the decryption site. The situation is the same with R0 and R16. A very important point we need to remember about the ciphers is that the round keys (K1 to K16) should be applied in the reverse order. At the encryption site, round 1 uses K1 and round 16 uses K16; at the decryption site, round 1 uses K16 and round 16 uses K1. In the first approach, there is no swapper in the last round.

- Each of these permutations takes a 64-bit input and permutes them according to a predefined rule.
- These permutations are keyless straight permutations that are the inverse of each other. For example, in the initial permutation, the 58th bit in the input becomes the first bit in the output.
- Similarly, in the final permutation, the first bit in the input becomes the 58th bit in the output.
- In other words, if the rounds between these two permutations do not exist, the 58th bit entering the initial permutation is the same as the 58th bit leaving the final permutation.
- These two permutations have no cryptography significance in DES. Both permutations are keyless and predetermined.

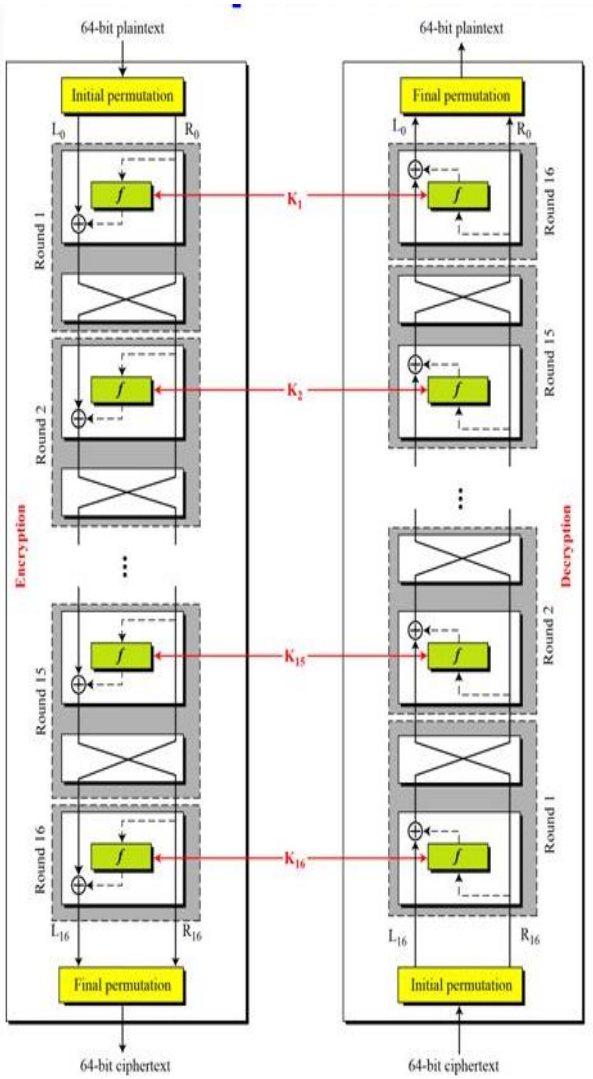


Fig. DES cipher and Reverse cipher for first approach

Key Generation:

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.

DES uses sixteen rounds of Feistel ciphers. It has been Rounds: proved that after eight rounds, each ciphertext is a function of every plaintext bit and every key bit; the ciphertext is thoroughly a random function of plaintext and ciphertext. Therefore, it looks like eight rounds should be enough. However, experiments have found that DES versions with less than sixteen rounds are even more vulnerable to known -plaintext attacks than brute-force attack, which justifies the use of sixteen rounds by the designers of DES

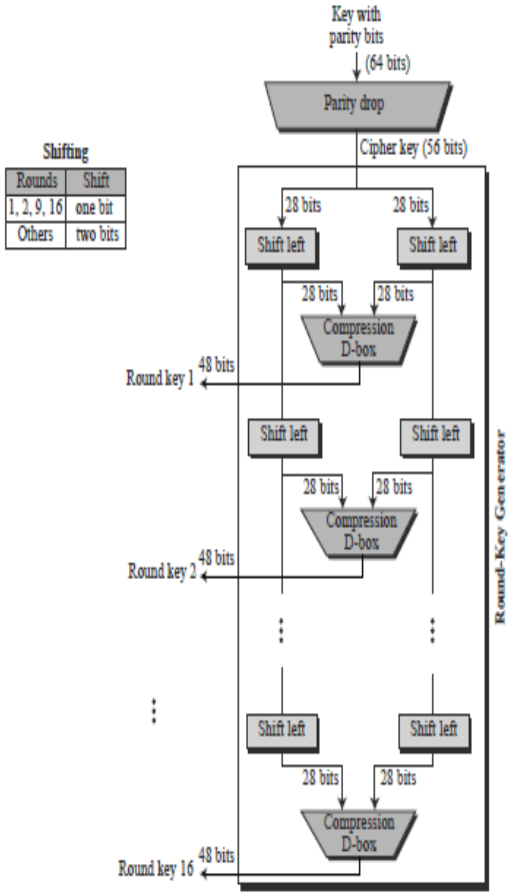
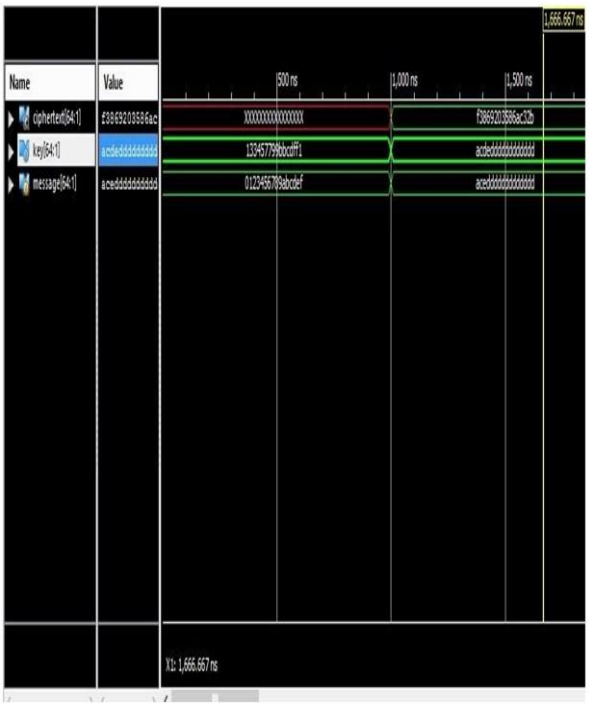
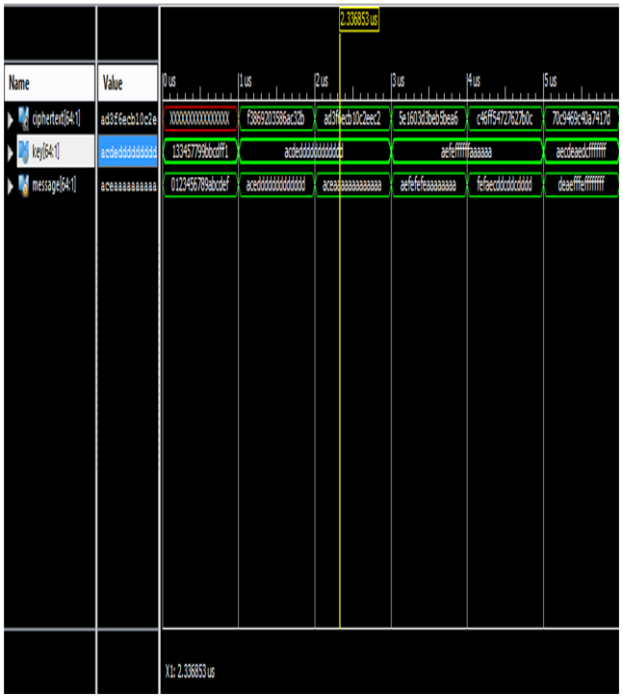


Fig. Key Generation

RESULT:





CONCLUSION

In this project, we have proposed an efficient VLSI Architecture for Data Encryption Standards (DES) Algorithm Based Encryption. As per the Requirements of Encryption The substitution operation (S-BOX) needed in the DES Algorithm has been implemented by Multiplexer Based Architecture the proposed architecture is very regular and it requires very low amount of hardware resources, therefore it can be efficiently utilized in light weight cryptography applications the design has been modeled in the XILINX software version 14.5.

REFRENCES

[1] W. Stallings, Cryptography and Practice, 5th ed. Prentice Hall, 2011.

[2] S. Vaudenay, A Classical Introduction to Cryptography Application for Communications Security. Springer science and Business media, 2006.

[3] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann and L. Uhsadel, A "survey of lightweight-cryptography Implementations" IEEE Design & Test of Computers, vol. 24, no. 6, 2007, pp. 522-533.

[4] G. Leander, C. Paar, A. Poschmann, and K. Schramm, "New Lightweight DES Variants," in Fast Software Encryption (FSE 2007), A. Biryukov, Ed. Springer Berlin Heidelberg: LNCS 4593, 2007, pp. 196-210.

[5] E. Biham and A. Shamir, Differential Cryptanalysis of the Data Encryption Standard. Springer Science & Business Media, 2012

[6] S. Kelly. (2006, Dec.) Security implications of using the data encryption (DES). [Online]. https://tools.ietf.org/html/rfc4772
[7] J. Daemen and V. Rijmen, The design of Rijndael: AES-the advanced encryption standard. New York USA: Springer Science & Business Media, 2013.
[8] C. Patterson, "High performance DES encryption in Virtex FPGAs using JBits," in IEEE Symposium on Field Programmable Custom Computing Machines, Napa Valley, CA, 2000 pp. 113-121.
[9] O. P. Verma, R. Agarwal, D. Dafouti, and S. Tyagi "Performance analysis of data encryption algorithms "in 3rd Int'l Conf. on Electronics Computer Technology (ICECT), vol. 5, Kanyakumari 8-10 Apr. 2011, pp. 399-403.
[10] M. E. Smid and D. K. Branstad "Data encryption standard: past and future," Proc. of the IEEE, vol. 76 no. 5, pp. 550-559, 1988.
[11] S. Landau, "Standing the test of time: The data encryption standard," Notices of the AMS, vol. 47, no. 3 Mar. 2000, pp. 341-349.