Efficient VLSI Implementation of DES and Triple DES Algorithm with Cipher Block Chaining concept using Verilog and FPGA

Aqib Al Azad
Department of EECS
North South University (NSU)
Dhaka, Bangladesh

ABSTRACT

In this paper, Data Encryption Standard (DES) and Triple Data Encryption Standard (TDES) algorithm and their efficient hardware implementation in cyclone II Field Programmable Gate Array (FPGA) is analyzed with the help of Cipher Block Chaining (CBC) concept. The Data Encryption Standard (DES) has been the most extensively used encryption algorithm in recent times. Triple DES is the common name for the Triple Data Encryption Algorithm (TDEA or Triple DEA) block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. The paper covers DES and Triple DES algorithm with Cipher Block Chaining concept, simulation results, basic FPGA technology and the implementation details of the proposed DES and Triple DES architecture. Register transfer level (RTL) of DES and Triple DES algorithm is designed, simulated and implemented separately using Verilog in different FPGA devices including Cyclone II, Spartan 3E, Vertex 5 and Vertex E series FPGAs. The results from the comparison with existing implementations show that the proposed design was efficient in all aspects.

General Terms

Encryption algorithm, Simulation, Hardwire implementation

Keywords

CBC, FPGA, DES, TDES, RTL, Verilog

1. INTRODUCTION

In today’s uncertain and increasingly wired world cryptology plays an important and significant role in protecting and securing communication channels, databases, and software from unwanted intruders.Modern block ciphers are widely used to grant encryption of quantities of information, and/or a cryptographic checksum to make sure the contents have not been revised. Among others the most widely used private key block cipher, is the Data Encryption Standard (DES). It was first adopted in the year 1977 by the National Bureau of Standards as Federal Information Processing Standard 46 (FIPS PUB 46) [1].DES encrypts data in 64-bit blocks using a 56-bit key. Although the original DES cipher's key size of 56 bits was sufficient and serving the purpose well when that algorithm was designed, the availability of increasing computational power made brute-force attacks feasible and predictable. Triple DES provides a relatively simple method of increasing the key size of DES, the main feature of which is to protect against such attacks. The advantage is that there was no need to design a completely new block cipher algorithm. A much more secure version of DES called Triple-DES (TDES), which is essentially equivalent to using DES three times on plaintext with three different keys. Though it is three times slower than the original form of DES, it is comparatively more secure. DES and Triple DES implementations can be found on reconfigurable hardware using FPGA devices [3-8].

The results constitute simulation of Verilog codes of different modules of the DES and Triple DES algorithm in Quartus II. The design was successfully implemented in the cyclone II FPGA using the Altera DE1 board. The design can also be synthesized to other FPGA architectures.

This paper presents an efficient design and implementation of DES and Triple DES algorithm in FPGAs. The following chronology is being followed to present the paper. In section 2, basics of cryptography and Cipher Block Chaining concept is discussed. Then in section 3, .the operation and architecture of DES algorithm and Triple DES algorithm is given briefly. Section 4 deals with the design architecture. Design hierarchies and block diagrams for both DES and Triple DES algorithms are shown and the basic blocks are described. After that in section 5, I discussed implementation strategies where details of design architecture and hardware blocks are shown. Here also implementation results are presented and in section 6, conclusion is drawn based on my results.

2. CRYPTOGRAPHY BASICS

Cryptography describes a process of encrypting information so that its meaning is hidden and thus secured from those who do not know how to decrypt the information. It beggars description to mention the immense importance of cryptography, both in the past and in the context of today's high tech world. A cryptographic algorithm (also known as a cipher) is a step by step sequence of mathematical calculations used to encrypt and decrypt information. There are currently three different types of cryptographic algorithms: hashing algorithms, symmetric-key algorithms and asymmetric key algorithms. Hashing algorithm creates a unique fixed length signature of a block of data. Hashes are created with an algorithm, or hash function, and are used to compare sets of data. A symmetric key encryption algorithm is one that both sender and receiver within the transmission channel share the same key. The asymmetric key algorithm, also known as the public-key algorithm, uses two different keys for encryption and decryption: Public key and private key. Symmetric key encryption is performed using two methods, block cipher and stream cipher [2].

2.1 Block cipher and Feistal structure

A block cipher is an encryption/decryption method in which a block of plaintext is treated as a whole and used to generate a ciphertext block of equal length. Block ciphers process messages into blocks, each of which is then encrypted/decrypted.

Usually many block ciphers have a Feistel structure and this type of structure consists of a number of identical rounds of processing. In each round, a substitution is carried out on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.

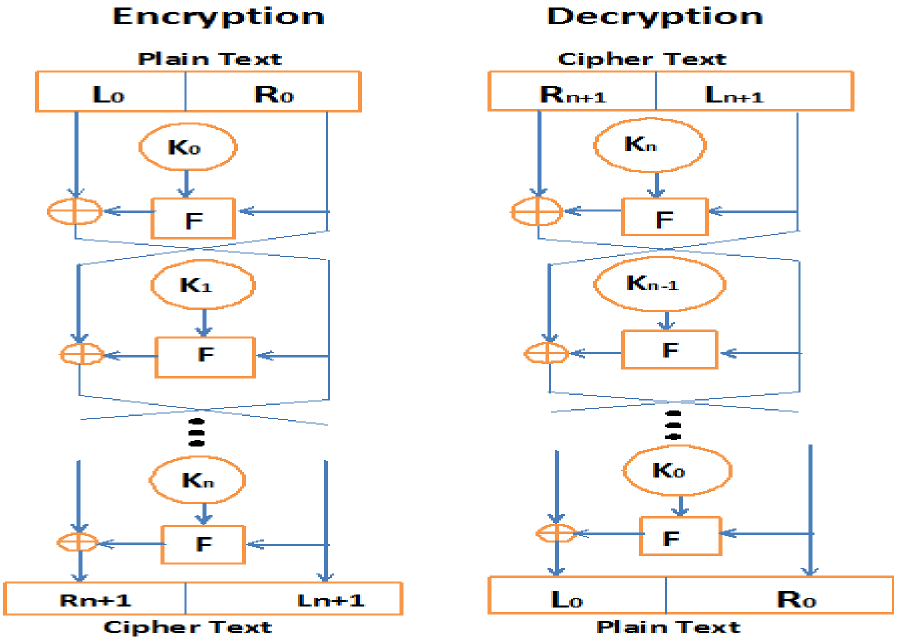


Fig. 1 Feistal structure

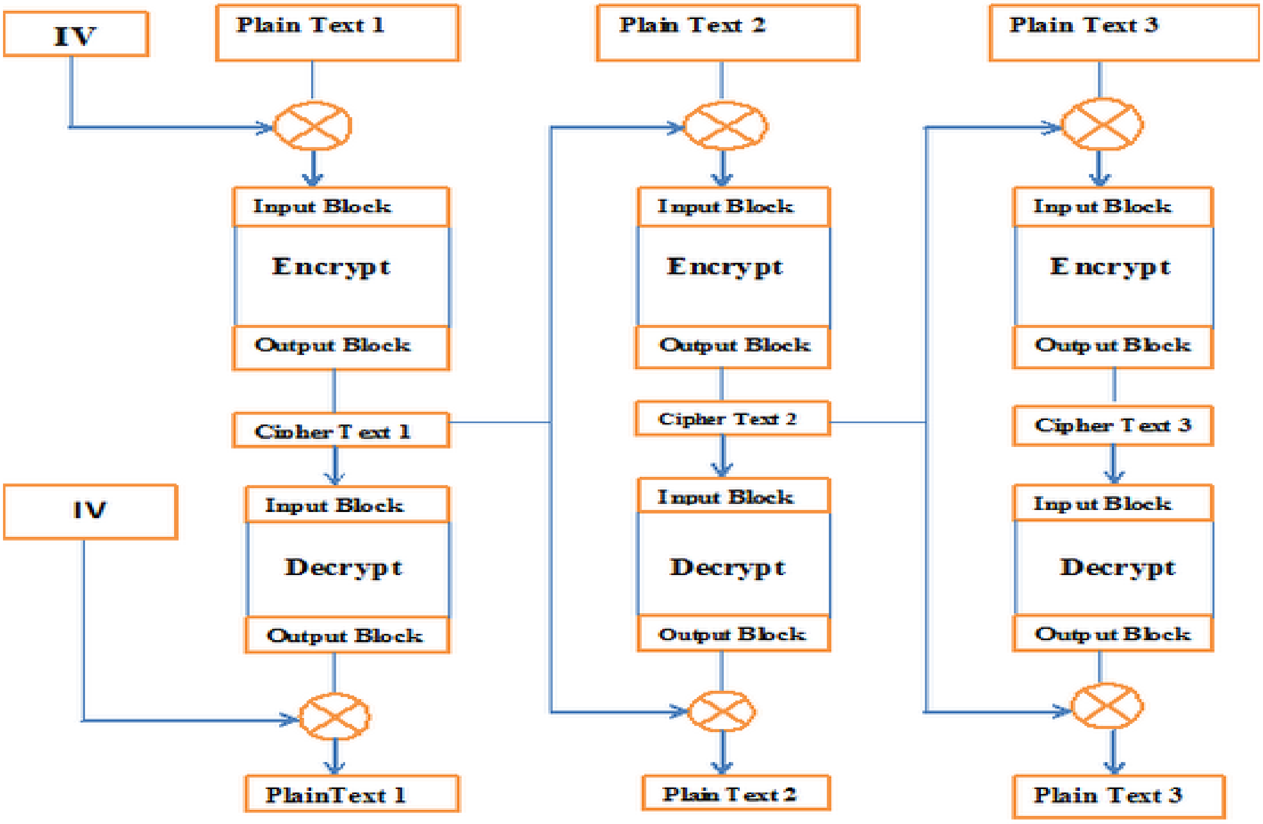


Fig. 2 Cipher Block Chaining (CBC) mode

Fig. 1 shows the general design of a Feistel cipher, a scheme used by almost all modern block ciphers. The input is broken into two equal size blocks, generally called left (L) and right (R). After that the blocks are repeatedly cycled through the algorithm.

At each cycle, a hash function (F) is applied to the right block and the key, and the result of the hash is XOR-ed into the left block. The blocks are then interchanged. As a result the XOR-ed result becomes the new right block and the unaltered right block becomes the left block. The process is then repeated a number of times further [12].

2.2 Cipher Block Chaining (CBC) Mode

CBC is the most widely used method of operation for a block cipher. Prior to encryption, each block of plaintext is XOR-ed with the prior block of ciphertext. After decryption, the output of the cipher must then be XOR-ed with the previous ciphertext to regain the original plaintext. The first block of plaintext is XOR-ed with an initialization vector (IV), which is usually a block of random bits transmitted in the clear. Between CBC and ECB, CBC is more secure than ECB as it effectively scrambles the plaintext prior to each encryption step. Since the ciphertext is constantly changing, two identical blocks of plaintext will encrypt to two different blocks of ciphertext. Fig. 2 shows the CBC mode operation.

CBC can be used to convert a block cipher into a hash algorithm. To perform this action, CBC is run repeatedly on the input data, and all the ciphertext is discarded except for the last block, which will depend on all the data blocks in the message. This last block becomes the output of the hash function [9].

2.3 Triple CBC (Cipher Block Chaining) Mode

This method is very similar to the standard DES CBC mode. As with Triple ECB, the effective key length is 168 bits and keys are used in the same manner, as described above, but the chaining features of CBC mode are also employed. The first 64-bit key acts as the Initialization Vector to DES. Triple ECB is then executed for a single 64-bit block of plaintext. The resulting ciphertext is then XORed with the next plaintext block to be encrypted, and the procedure is repeated. This method adds an extra layer of security to Triple DES. Hence it is more secure than Triple ECB, although it is not used as widely as Triple ECB [13].

3. DES ALGOROTHIM

```
The Input:
  T: 64 bits of clear text
   k1, k2, ..., k16: 16 round keys
   IP: Initial permutation
   FP: Final permutation
   f(): Round function
Output:
   C: 64 bits of cipher text
Algorithm:
   T' = IP(T), applying initial permutation
   (L0, R0) = T', dividing T' into two 32-bit parts
   (L1, R1) = (R0, L0 ^ f(R0, k1))
   (L2, R2) = (R1, L1 ^ f(R1, k2))
   ......
   C' = (R16, L16), swapping the two parts
   C = FP(C'), applying final permutation
```

3.1 Overall structure

The algorithm's overall structure is shown in Fig. 3. There are 16 identical stages of processing, which are termed as rounds. There is also an initial and final permutation, known as *IP* and *FP*, which are inverses (IP "undoes" the action of FP, and vice versa).

Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the Feistel scheme which is already been discussed in the earlier sections.

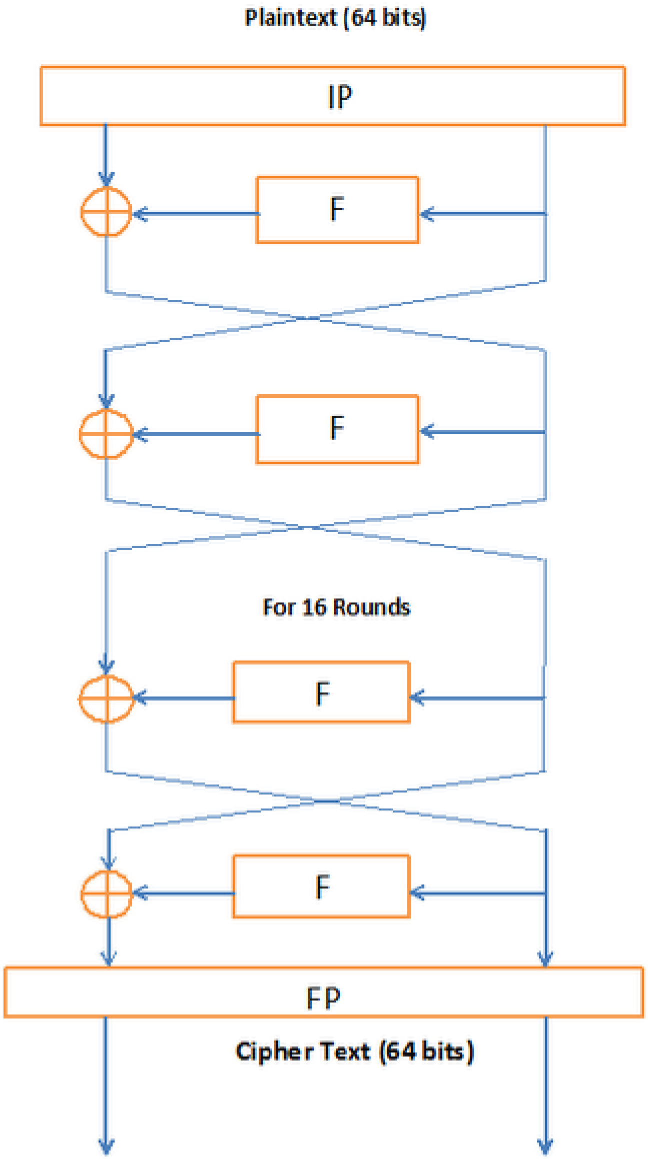


Fig. 3 DES Overall Structure

3.2 DES Round Structure- the Feistel (F) function

The F-function, depicted in Fig. 4, operates on half a block (32 bits) at a time and consists of four stages:

- Expansion
- Key mixing
- Substitution
- Permutation

Fig. 4 illustrates the internal structure of the DES round function F. The 32-bit half-block is expanded to 48 bits by using expansion table E that defines a permutation plus an expansion. This gives an output which consists of eight 6-bit(8*6=48bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side. The result is then XOR-ed with the subkey. 16 48-bit subkeys — one for each round — are derived from the main key using the key schedule algorithm. This 48-bit result passes through a substitution function comprising 8 S-boxes which each map 6 input bits to 4 output bits, producing a 32-bit output, which is then permuted by permutation P. This is designed in such a way that, after expansion, each S-box's output bits are spread across 6 different S boxes in the next round.

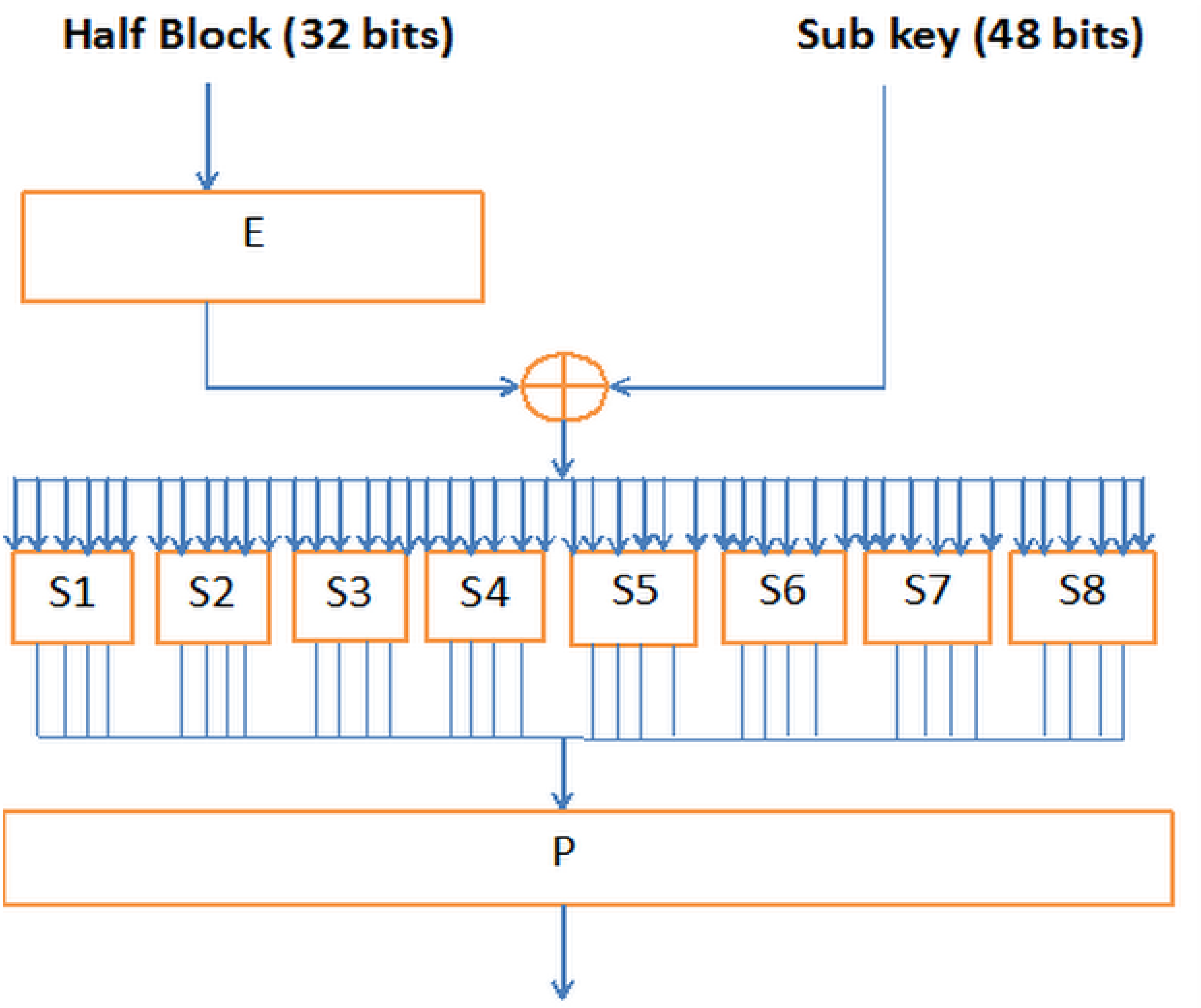


Fig. 4 DES Round Structure- the F function

The alternation of substitution from the S-boxes, and permutation of bits from the P-box and E-expansion provides so-called "confusion and diffusion" respectively, a concept identified by Claude Shannon in the 1940s as a necessary condition for a secure yet practical cipher [10].

3.3 DES DECRYPTION ALGORITHM

The decryption algorithm of a block cipher should be identical to the encryption algorithm, step by step but, in a reverse order. But in case of DES cipher, the encryption algorithm is so well designed, that the decryption algorithm is identical to the encryption algorithm step by step in the same order, only with the subkeys applied in the reverse order. Feistel structure makes encryption and decryption similar processes.

3.4 Triple DES algorithm

The main purpose behind the development of Triple DES was to address the obvious flaws in DES without making an effort to design a whole new cryptosystem. Triple DES simply extends the key size of DES by applying the algorithm three times in succession with three different keys. The combined key size is thus 168 bits (3 times 56), beyond the reach of brute-force techniques such as those used by the EFF DES Cracker. Triple DES has always been under some suspicion, as the original algorithm was never designed to be used in this way, however, no serious flaws have been discovered in its design, and it is today a viable and widely used cryptosystem with usage in a number of Internet protocols [9].

The standards define three keying options:

- Keying option 1: All three keys are independent. Keying option 1 is the strongest, with 3 × 56 = 168 independent key bits.
- Keying option 2: K_1 and K_2 are independent, and K_3 = K_1. Keying option 2 provides less security, with 2 × 56 = 112 key bits. This option is stronger than simply

DES encrypting twice, e.g. with K_1 and K_2, because it protects against meet-in-the-middle attacks.

- Keying option 3: All three keys are identical, i.e. $K_1 = K_2 = K_3$. Keying option 3 is equivalent to DES, with only 56 key bits. This option provides backward compatibility with DES, because the first and second DES operations cancel out. It is no longer recommended by the National Institute of Standards and Technology (NIST) and is not supported by ISO/IEC 18033-3 [11].

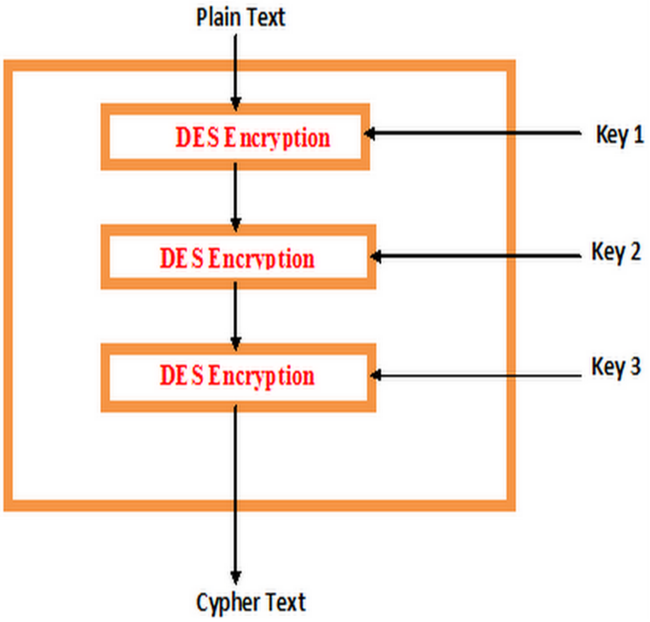


Fig. 5 Working of Triple DES Algorithm

4. IMPLEMENTATION DETAILS OF BASIC BLOCKS

4.1 Design hierarchy

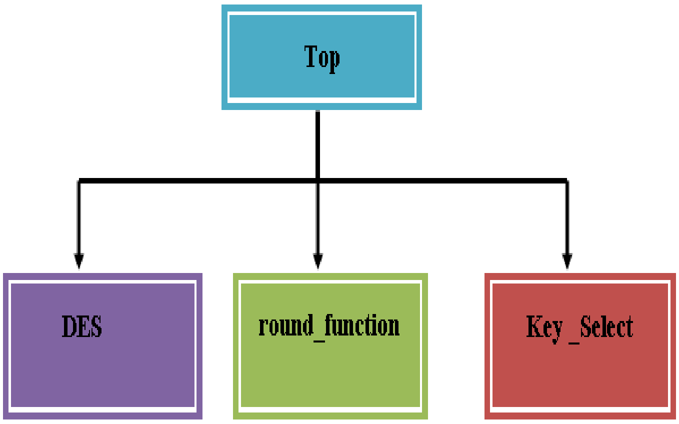


Fig. 6 Design hierarchy for DES

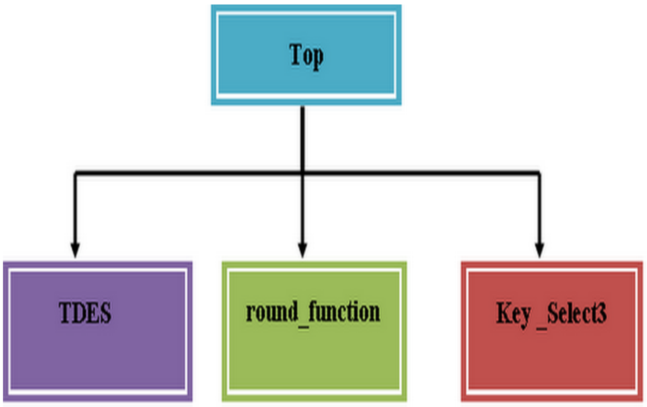


Fig. 7 Design hierarchy for TDES

4.2 Block diagram

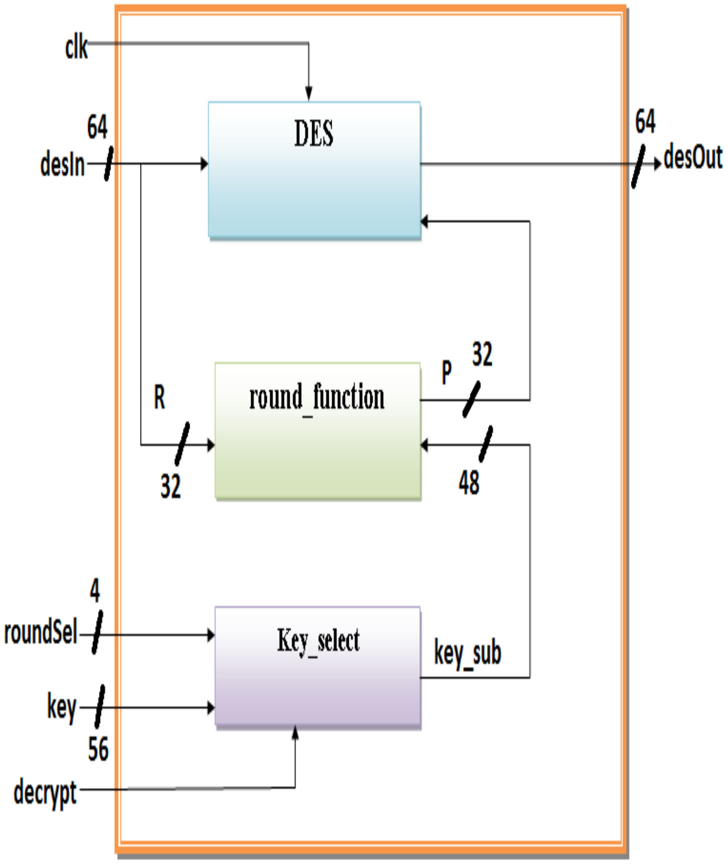


Fig. 8 Block diagram for DES

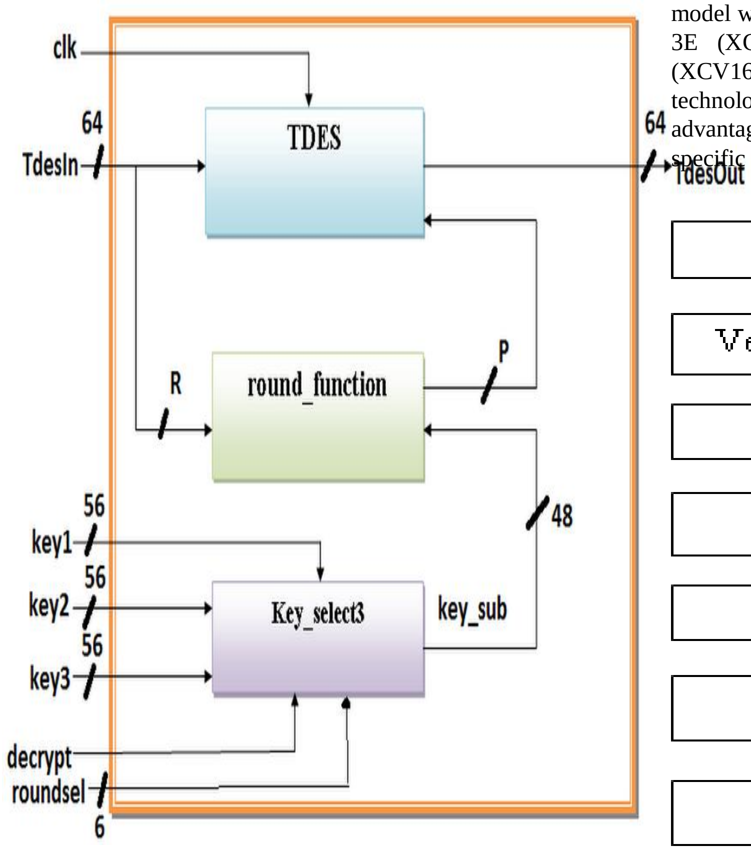


Fig. 9 Block diagram for TDES

4.3 Block description of DES

4.3.1 Round_function

Applies expansion permutation and then XOR with the round key. After that applies S boxes function and returns 32-bit data,R. This block Implements CBC DES encryption algorithm.

4.3.2 DES

Applies initial permutation and final permutation and gives 64 bit output.

4.3.3 *Key_select*

Select one of 16 sub-keys for round, returns 48-bit data key_sub.

4.4 Block description of TDES

4.4.1 Round_function

Applies expansion permutation and then XOR with the round key. After that applies S boxes function and returns 32-bit data,R. Implements outer triple CBC DES encryption algorithm with three keys.

4.4.2 TDES

Applies initial permutation and final permutation and gives 64 bit output.

4.4.3 *Key_select3*

Select three of 16 sub-keys for round, returns 48-bit data key_sub.

5. EXPERIMENTAL FRAMEWORK AND RESULTS

In this section we describe the design procedure and the architecture of DES and Triple DES. Fig. 10 shows the different stages of my design. The Verilog model was synthesized with Quartus II Software targeted for Cyclone II (EP2C20F484C7) device and simulated with Modelsim. Then also the Verilog model was synthesized with Xilinx Software targeted for Spartan 3E (XC3S1600E), Vertex 5 (XC5VLX50) and Vertex E (XCV1600E) device and simulated with Modelsim. FPGA technology was chosen because it provides some important advantages over general purpose processors and application specific integrated circuits (ASICs).

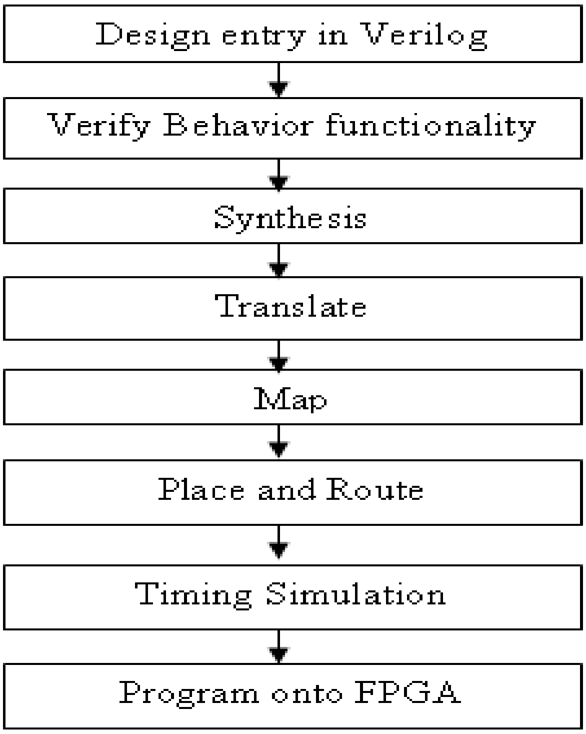


Fig. 10 Implementation flow

The RTL architecture of DES and Triple DES is shown in Fig. 11 and Fig. 12 respectively.

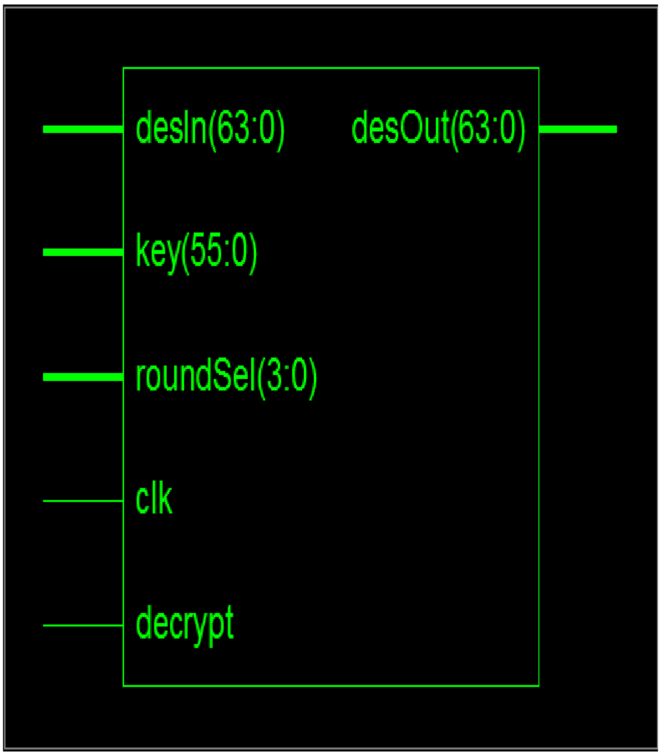


Fig. 11 RTL schematic of DES

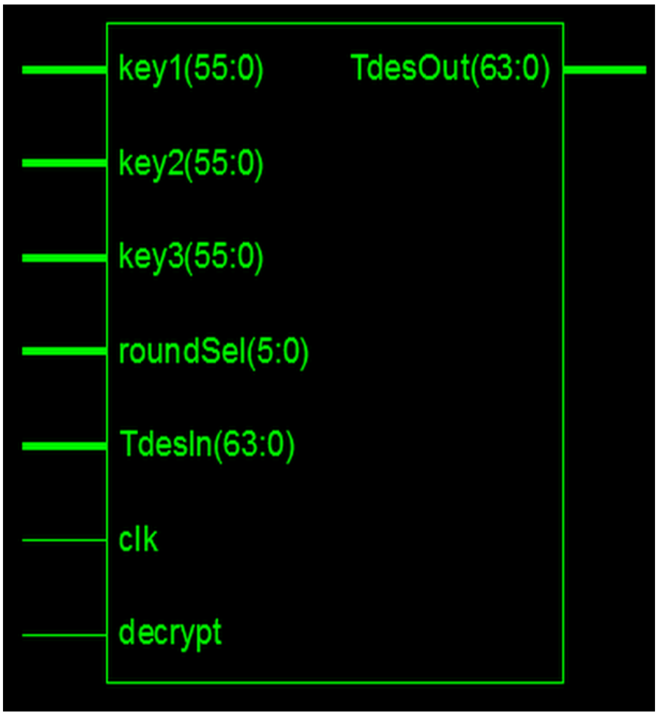


Fig. 12 RTL schematic of TDES

Fig. 13 and Fig. 14 illustrate the implemented components inside the chip. Additionally, the interconnections of the components are shown in Fig. 15 and Fig. 16.

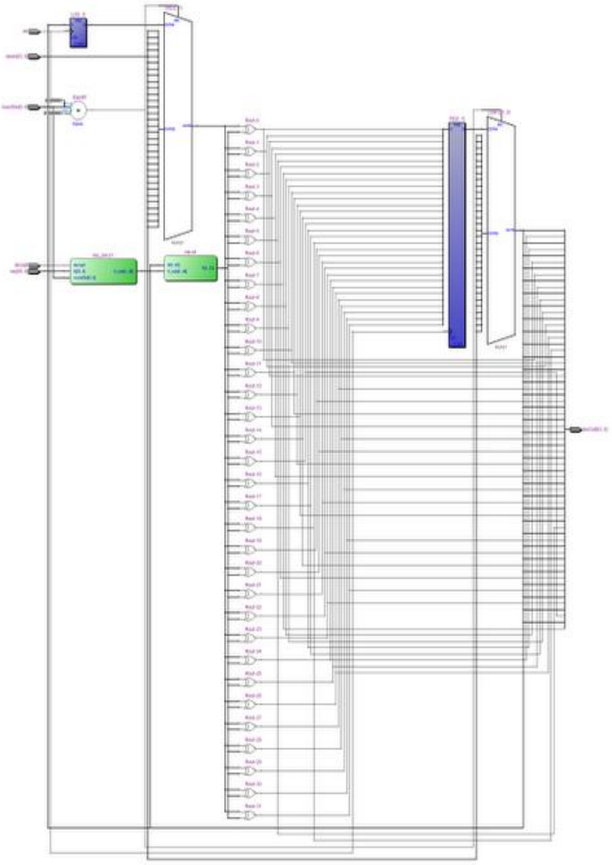


Fig. 13 Internal block diagram of DES

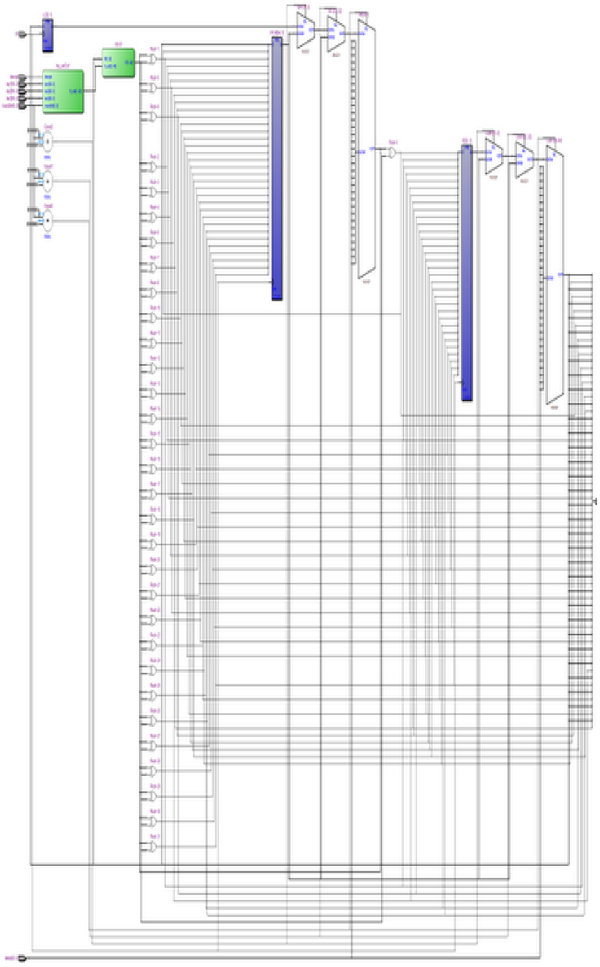


Fig. 14 Internal block diagram of TDES

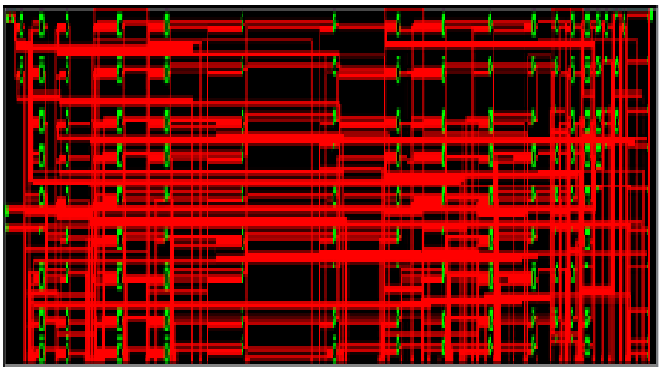


Fig. 15 Technology Schematic of DES

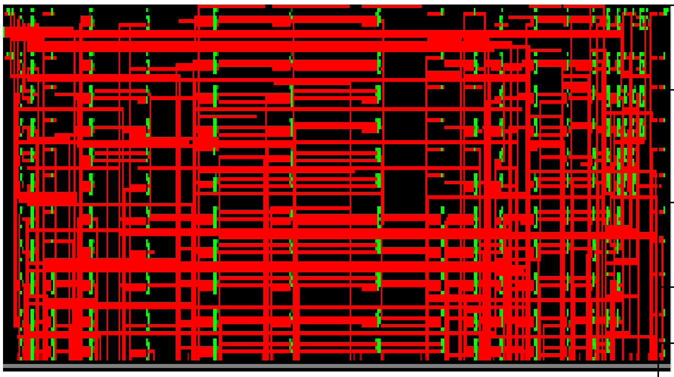
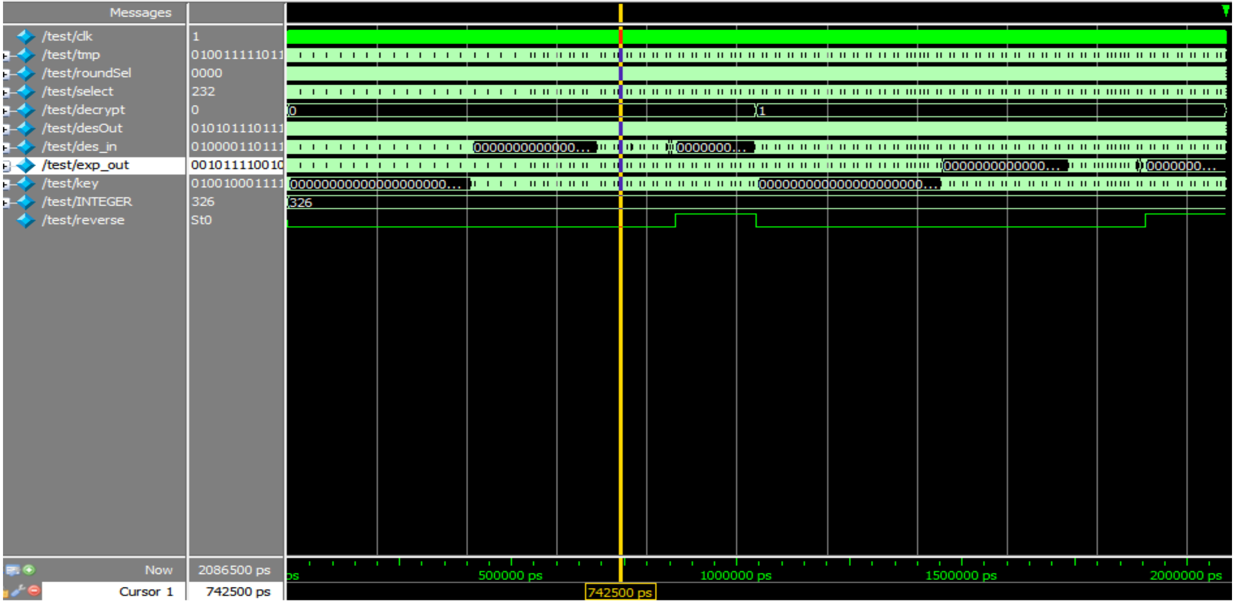


Fig. 16 Technology Schematic of TDES

Table 1 shows synthesis results of DES and Triple DES algorithm for cyclone I device.

Table 1. Synthesis results for Cyclone II

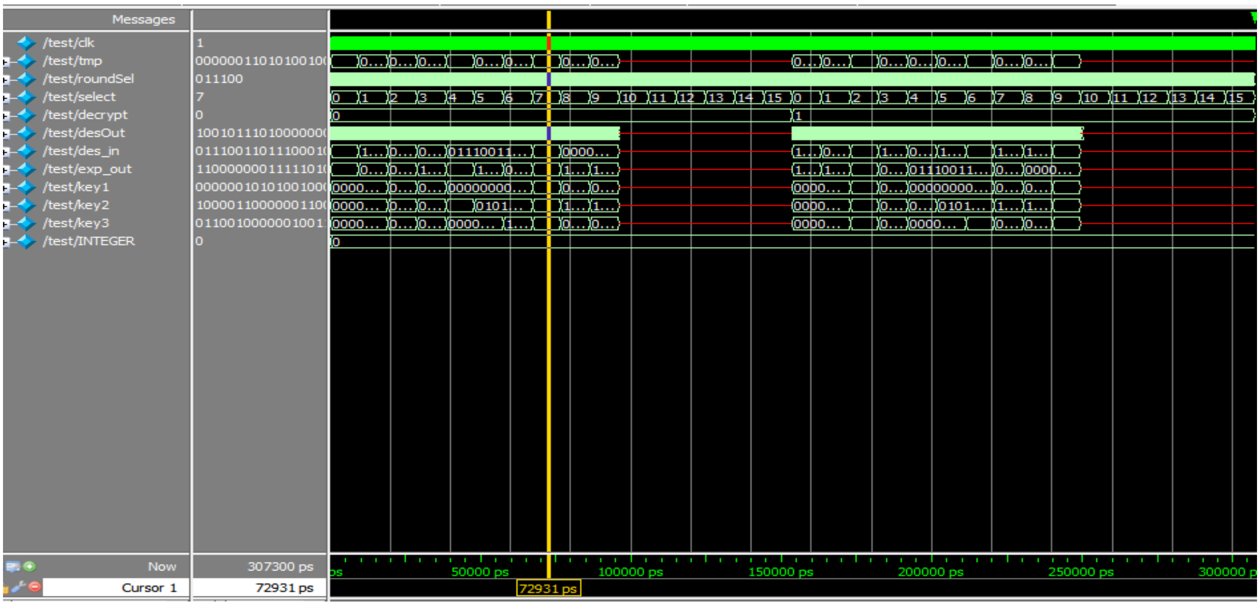
	DES implementation	**TDES implementation**
	Cyclone II (EP2C20F484C7, package FBGA, speed grade-7)	Cyclone II (EP2C20F484C7, package FBGA, speed grade-7)
Total logic elements	805 / 18,752 (4 %)	1,088 / 18,752 (6 %)
Total combinational functions	805 / 18,752 (4 %)	1,088 / 18,752 (6 %)
Dedicated logic registers	64 / 18,752 (< 1 %)	64 / 18,752 (< 1 %)
Total registers	64	64
Total pins	190 / 315 (60 %)	304 / 315 (97 %)
Total virtual pins	0	0
Total memory bits	0 / 239,616 (0 %)	0 / 239,616 (0 %)
Embedded Multiplier	0 / 52 (0 %)	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)	0 / 4 (0 %)



```
Messages
/test/clk          1
/test/tmp          010011111011
/test/roundSel     0000
/test/select       232
/test/decrypt      0
/test/desOut       010101110111
/test/des_in       010000110111
/test/exp_out      001011110010
/test/key          010010001111
/test/INTEGER      326
/test/reverse      St0

Now                2086500 ps
Cursor 1           742500 ps
```

Fig. 17 Output of DES



```
Messages
/test/clk        1
/test/tmp        000000110101001OO
/test/roundSel   011100
/test/select     7
/test/decrypt    0
/test/desOut     10010111010000000
/test/des_in     0111001101110001O
/test/exp_out    1100000001111101O
/test/key1       000000101010010OO
/test/key2       1000011000000110O
/test/key3       0110010000001001
/test/INTEGER    0
Now       307300 ps
Cursor 1  72931 ps
```

Fig. 18 Output of TDES

Fig. 17 and Fig. 18 show the simulation waveform of DES and TDES algorithm. These are the sequential implementations and need 16 and 48 cycles respectively to complete a full encryption/decryption cycle.

Table 2. Synthesis results for DES in Spartan 3E

	DES implementation	
	Spartan 3E (XC3S1600E, package fg484,speed grade-5)	
Logic Utilization	Used	Utilization
Number of Slices	448 out of 14,752	3%
Number of Slice Flip Flops	64 out of 29,504	1%
Number of 4 input LUTs	852 out of 29,504	2%
Number of bonded IOBs	190 out of 376	50%
Number of GCLKs	1 out of 24	4%

Table 2 and Table 3 show synthesis results of DES algorithm for Spartan 3E and Vertex5 device respectively.

Table 3. Synthesis results for DES in Vertex 5

	DES implementation	
	Vertex 5 (XC5VLX50, package ff1153,speed grade -1)	
Logic Utilization	Used	Utilization
Number of Slices	114 out of 7,200	1%
Number of Slice Flip Flops	64 out of 28,800	1%
Number of fully used LUT-FF pairs	64 out of 389	2%
Number of bonded IOBs	190 out of 560	33%
Number of BUFG/BUFGCTRLs	1 out of 32	3%

In Table 4 and Table 5 the comparison between the two existing implementations and the proposed architecture implementation for the Vertex 5 [6] series and Vertex E series [7] respectively is shown.

Table 4. Synthesis results for TDES in Vertex 5

	TDES Existing implementation result [6]	**TDES proposed implementation result**
	Vertex 5 (XC5VLX50,package ff1676,speed -1)	Vertex 5 (XC5VLX50,package ff1676,speed-1)
Number of Slices	1206 / 28,800 (4%)	64/ 28,800 (1 %)
Number of Slice Flip Flops	1690 / 28,800 (5%)	887/ 28,800 (3%)
Number of fully used LUT-FF pairs	447 / 2449 (18%)	25 / 926 (2%)
Number of bonded IOBs	302 / 440 (68 %)	304 / 440 (69 %)
Number of BUFG/BUFGCTRLs	1 / 32 (3%)	1 / 32 (3%)

Table 5. Synthesis results for TDES in Vertex E

	TDES Existing implementation result [7]	**TDES proposed implementation result**
	Vertex E (XCV1600E,package bg560,speed -8)	Vertex E (XCV1600E, package bg560,speed -8)
Number of Slices	1481 / 15552 (9%)	645/ 6912 (9 %)
Number of Slice Flip Flops	1256 /31104 (4%)	64/ 13,824 (1%)

Number of 4 input LUTs	2396 /31104 (7%)	767 / 13,824 (5%)
Number of bonded IOBs	302 / 404 (74 %)	303 / 404 (75 %)
Number of BUFG/BUFGCTRLs	1 / 4 (25%)	1 / 4(25%)

It can be inferred from the above two comparisons that the proposed implementation is very much compact and efficient in all
respects than others.

6. CONCLUSIONS

In this work a compact hardware implementation of DES and Triple DES was presented. The design was implemented in real hardware with Cyclone II FPGA. The proposed architecture was also implemented with Spartan 3E, Vertex 5 and Vertex E FPGA devices and compared with the existing results. Here Cipher Block Chaining modes have been used by combining the previous cipher text block with the current message block before encrypting. DES and Triple DES algorithm are used significantly in satellite communications and electronic financial transactions, cryptographic key encryption for automated key management applications, file encryption, mail encryption, and other applications. In fact, it is extremely difficult, if not impossible, to find a cryptographic application where the DES cannot be applied. Technologies are becoming smarter and compact day by day, so we hope this work will add new dimension in that trend. This design will play a remarkable role with its significant speed and efficiency.

7. ACKNOWLEDGMENT

First of all, author would like to thank Almighty Allah for making it possible to complete this work. The author is also grateful to many people for helping to perform this work. He would like to convey his cordial gratitude to Dr. M Abdul Awal, Chairman of EECS Department, North South University for providing the facilities for the completion of this work. Lastly, he would like to thank his parents for their help and support during this period.

8. REFERENCES

[1] "Data encryption standard (DES) ", National Bureau of Standards (U.S.), Federal Information Processing Standards Publication 46, National Technical Information Service,Springfield, VA,Apr. 1977.

[2] W. Stallings, *Cryptography and Network Security*: *Principles and Practice,* 4th ed , Prentice-Hall, 2006.

[3] K. Wong, "A single-chip FPGA implementation of the data encryption standard (des) algorithm" IEEE 1998 pp 827-832.

[4] F. Hoornaert, J. Goubert, and Y. Desmedt, "Efficient hardware implementation of the DES," in Proc. Adv. Cryptol. (CRYPTO'84), 1984, pp. 147–173.

[5] A. Dhir "Data Encryption using DES/Triple-DES Functionality in Spartan-II FPGAs", White Paper: Spartan-II FPGAs, WP115 (v1.0) March 9, 2000.

[6] P. Ghosal and M. Biswas, "*A Compact FPGA Implementation of Triple-DES Encryption System with IP Core Generation and On-Chip Verification*", International Conference on Industrial Engineering and Operations Management, 2010.

[7] F. Antonios, P. Nikolaos, M. Panagiotis, and A. Emmanouel, "Hardware Implementation of Triple-DES Encryption/ Decryption Algorithm", International Conference on Telecommunications and Multimedia, 2006.

[8] T. Schaffer, *A.* Glaser, and P. D. Franzon, "*Chip-package co-implementation of a triple DES processor*," IEEE Transactions on Advanced Packaging, pp. 194-202, *Feb. 2004.*

[9] Freesoft. (2011). Block Ciphers. Retrieved November 20, 2011, from http://www.freesoft.org/CIE/Topics/143.htm.

[10] Wikipedia. (2011). Data Encryption Standard. Retrieved November 20, 2011, from http://en.wikipedia.org/wiki/Data_Encryption_Standard.

[11] Wikipedia. (2011). Triple DES. Retrieved November 20, 2011, from http://en.wikipedia.org/wiki/Triple_DES.

[12] C. Boyd. "Modern Data Encryption," Electronics & Communication Engineering Journal, October 1993, pp 271-278.

[13] W. Diffie, "Cryptographic Technology: Fifteen Year Forecast" Reprinted by permission AAAS, 1982 from Secure Communications and Asymmetric Crypto Systems. AAAS Selecte8 Symposia. Editor: C.J. Simmons. Vol. 69, Westview Press, Boulder, Colorado, pp 38-57.