OPTIMAL IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHMS ON FPGA

Gabriel V. IANA[1], Petre ANGHELESCU[1], Neacsa TRAIAN[1], Gheorghe SERBAN[1]
[1]University of Pitesti
gbiana2000@gmail.com, anghelescu.petre@upit.ro, serban@upit.ro

Keywords: DES, FPGA, hardware implementation, encryption

Abstract: *Encrypting data structure type FPGA or ASIC hardware is done at a much higher speed than the software implementations. TDES is one of the most well known algorithms with secret key used in data encryption. A variety of implementations in FPGA or ASIC hardware structures are currently made. This paper presents the implementation of this algorithm on a XILINX SPARTAN 3 reconfigurable structure. The implementation process is followed by the optimizations and a bigger speed data processing on the used surface of the circuit was achieved – such as a speed of 4.8 GB/s and 26% area structure. It describes how it works and finally presents the comparative results obtained from the implementations of digital module*.

1. INTRODUCTION

FPGAs (Field programmable gate array) structures are the most attractive to enable implementation of custom digital hardware modules in some specific encryption algorithms. In this paper, the implementation of a TDES (Triple Data Encryption Standard) algorithm is made. The FPGA structures enable implementation of the algorithm at higher frequencies than the software, because parallelization facilities the encryption blocks on more digital submodules. Moreover, it allows a reconfiguration and also a hardware upgrade to the digital modules and the adaptation of them fast enough to be implemented in ASIC structures (Application Specific Integrated Circuits), which will operate at higher speed [1, 2].

The need of communication and the rapid increase of the amount of the information transmitted using network communications have determined the development of new secure techniques to protect the information. On communication systems, security mechanisms are necessary for increasingly complex messaging safe. Messages can be transferred as information between companies, bank transfers, etc. To secure communication channels several algorithms are used, such as bio-inspired algorithms (using cellular automata structures), chaos based cryptography [3], hash functions, public key algorithm or symmetric encryption algorithm where a DES algorithm is employed. This algorithm is organized into several repetitive rounds that consist in some bitwise logical operations, movement permutations and substitutions [4, 5].

2. CRYPTOGRAPHIC TECHNIQUES

The recent developments in communications and computer technology have made the security and privacy of data a major problem and concern in diverse fields like internet banking, e-commerce, information hiding, business in general, defence, etc. This has led to development of various techniques and adoption of cryptography. As a cross-discipline of cryptology and image processing, image/video security has also attracted much attention recently. The following two topics are chiefly focused: digital watermarking of image and video, image video encryption, the former corresponds to information hiding in cryptology, and the latter is an application of pure cryptology to protect multimedia contents.

Telecommunications technology advances permanently. The cryptography is as old as the need of sending confidential messages for long distances and protected stored data. Now, in the time of computer global communication and mobile telephony, there is a necessity of creating new, but fast and secure algorithms to protect the information. The growth of electronic commerce and the emphasis of privacy have intensified the need to find a fast and secure cryptographic method.

The main objective of cryptography is to develop a cryptosystem, which converts an original intelligible message, referred as plaintext, into apparently random non-sense message, referred as ciphertext and it also recovers the message back in its original form [3].

3. RECONFIGURABLE STRUCTURE

In this paper there has been used the smallest reconfigurable hardware structure SPARTAN 3 [6]. The circuit is organized as a matrix that contains configurable logic blocks and an interconnection network. Interfacing with the outside of device is made through I/O configurable block. The CLB (Configurable Logic Block) enable the implementation of synchronous logic combinational. Each logical block contains four slices that are organized in two pairs of two, called the left and right side pair and has attached a switching matrix as is shown in fig. 1.

Each pair has one independent carry channel. Pair from the left can be configured as distributed RAM. A slice has two function generators, multiplexers, and storage elements, fast carry logic and arithmetic gates.

Spartan 3 is a simple structure derived from Virtex II. The circuit used to implement the TDES module described in this paper is XC3S400 and has the following characteristic: has a 400k system gate, 896 CLB held on 32 rows and 28 columns, 56kBits distributed RAM, 288kbits RAM blocks, and 16 multipliers.

4. TDES ALGORITHMS

In cryptography TDES encryption block consists in DES (Data Encryption Standard) blocks used for three times. When it was concluded that a 56-bit key DES encoder is not sufficient to protect against attacks, TDES was chosen as the easiest way to increase the number of bits of the key without going to a new algorithm.

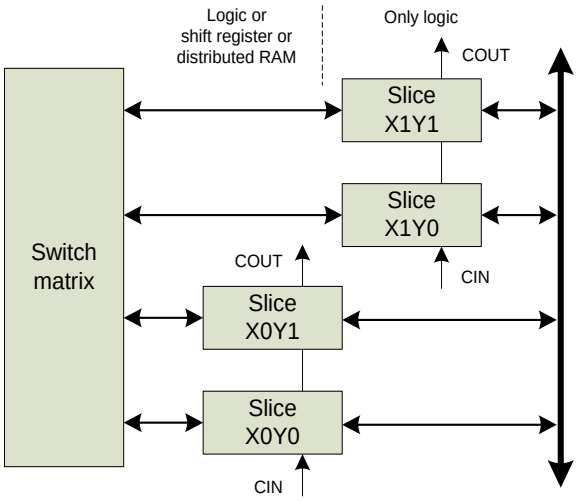

Fig. 1. Configuration of the slice into a CLB

The simplest variant of TDES operates as follows:

$$DES(k_3;DES(k_2;DES(k_1;M))) \quad (1)$$

where M is the message which will be encrypted and K1, K2, K3 is the DES key. Another variant of TDES used and in this paper was chosen because we needed to simplify the interoperability between DES and TDES. In this variant the middle step is usually replaced by a decryption (EDE mode), resulting the following relationship after the algorithm operates.

$$DES(k_3;DES^{-1}(k_2;DES(k_1;M))) \quad (2)$$

DES is a block cipher archetype - as an algorithm that takes a fixed length string of bits of normal text and converts it through a series of complex operations in an encrypted bit string the same length. In this DES algorithm the block size is 64 bits. DES also uses a key to customize the transformation, so that only those who know the decryption key used to carry out. The key consists of 64 bits; however, only 56 of them are used by the algorithm itself. Eight bits are used as parity bits and are not required after this test. So the key is effective only on 56 bits, and so is usually cited.

The general structure of the algorithm appears in fig. 2 and is 16 identical processing steps, appointed rounds. There are one initial and one final permutation, called IP and FP, which are inverse functions (IP "undo" the action of PF and vice versa). IP and FP have almost no cryptographic significance, but were included for facilitating loading and unloading blocks using hardware since 1970.

Before the main rounds, the block is divided into two halves, each of 32 bits, and alternatively processed. This alternation is known as the Feistel scheme. Feistel structure ensures that the encryption and decryption processes are very similar – the only difference is the order of application subkeys - opposite to decryption. The rest of the algorithm is identical. This simplifies implementation, particularly the hardware, because you do not need separate algorithms.

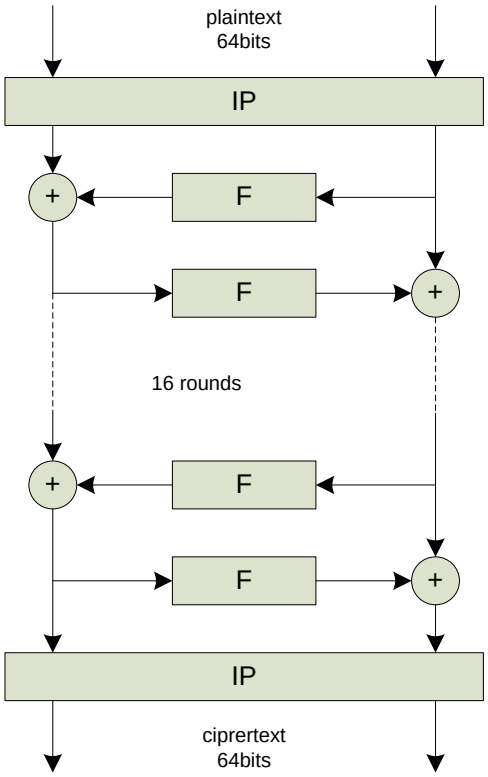

Fig.2. Schematic bloc of DES algorithm

The plus symbol denotes the exclusive OR operation (XOR). Function Block F mixes half of a subkey. Outcome function F is combined with other half of the block and the halves are swapped before the next round.

After the last round, the halves are not exchanged; it is a feature of the Feistel structure which makes encryption and decryption processes similar.

5. HARDWARE IMPLEMENTATION

The TDES algorithm was designed to encrypt blocks of 64 bits using two keys with a same length of 64 bits [6]. At the TDES circuit, actually stays *TOP_DES* module that is applied for 3 times. At encryption, first *TOP_DES* block encrypt the data using the *key1* input key, the second block *TOP_DES* decrypt the encrypted string from first block using *key2* input key, and lead to scrambling encrypted block. The third block *TOP_DES* encrypts the decrypted message before using the first key *key1*. At the decryption process is reversed in order, the first block *TOP_DES* decrypt by using the first key *key1*, the second block encrypt by using the second key *key2*, and the third block decrypt by using the first key *key1*.

In the fig. 3 is presented the structural implementation in FPGA of TDES algorithm.

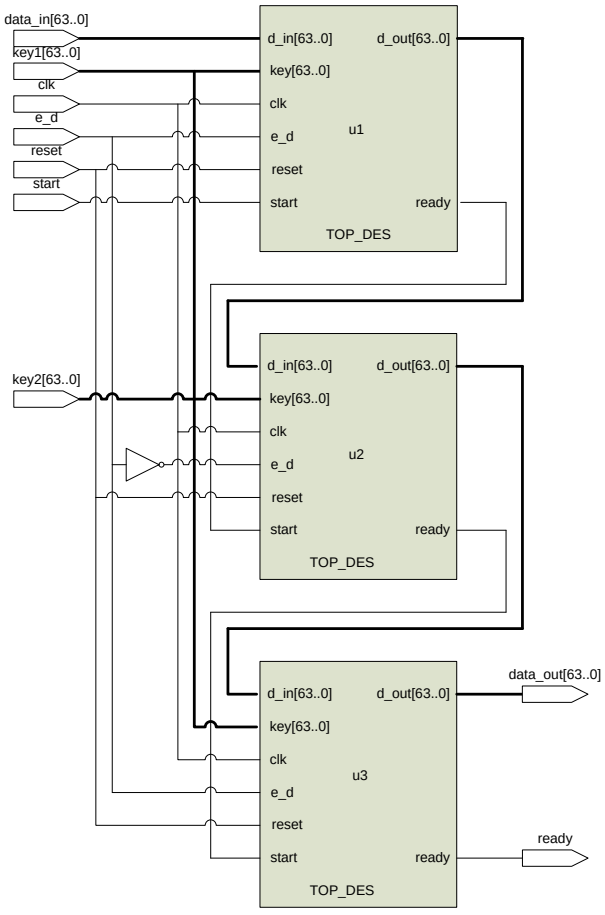

Fig. 3. TDES schematic block

The Top Level DES blocks scheme (*TOP_DES*) presented until now don't participate in the actual encryption/decryption process of data. They are only control functions.

Basically, encryption/decryption is done in the next block that we will name *des*. The structure of *TOP_DES* module is shown in fig. 4.

Through *counter_4* block the encryption/decryption steps are counted. The algorithm runs in 16 rounds.

The *mux_selector* clock command internal selections of *des* module and the *ready* module let us know when encryption or decryption process ended.

The description of des block shown in fig. 5 is as follows:

- *subkey* block generates the subkeys used in all 16 rounds for encryption/decryption process. In each round is using one subkey;
- the initial data block IP is divided into two halves, which pass through *mux21* blocks and stored in registers of the two *reg32* blocks;
- first half (*reg32* block stored in the right) is passing into the *e* block, is processed, and after the data arrive in the block *XOR48*, where XOR is made with subkey from the current round;
- it continues through *sbox* block then pas by *P* block where the permutations are realized;

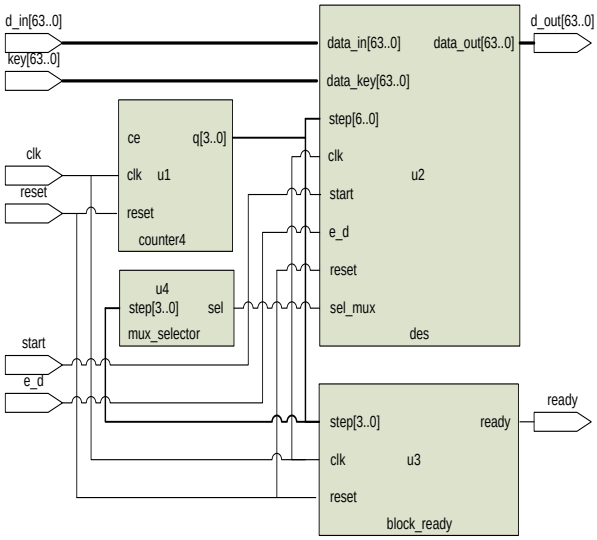

Fig. 4. Schematic block of TOP_DES module

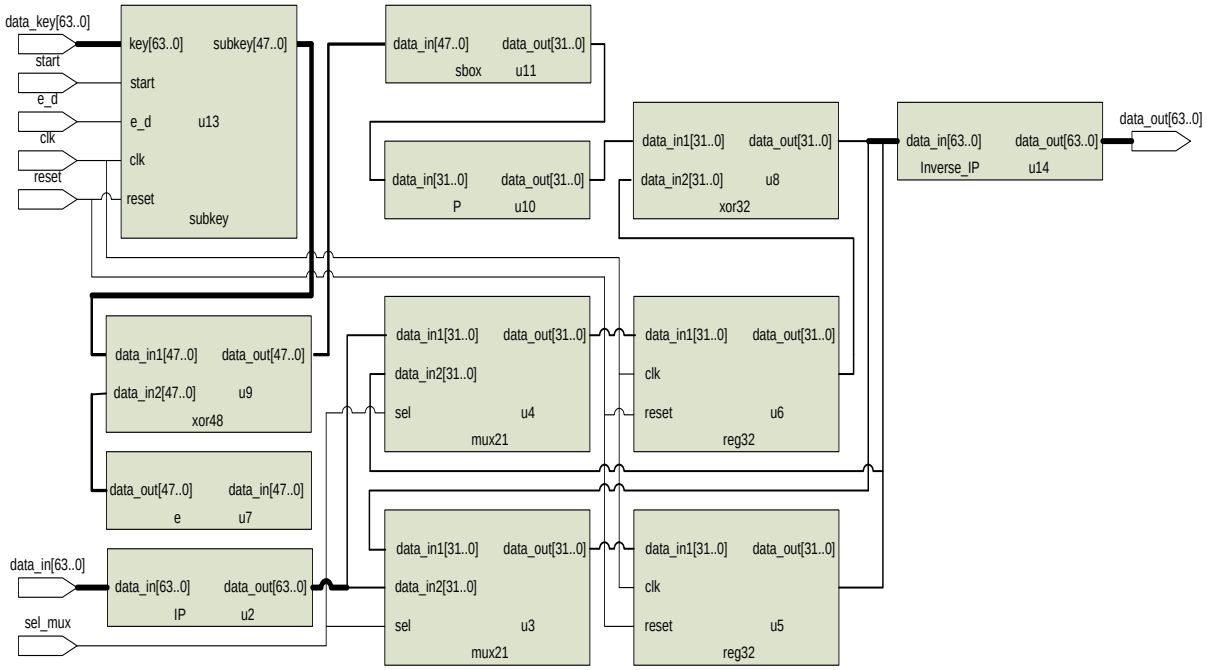

Fig. 5. Schematic block of des module

- the other half (*reg32* block stored in the left side) is passing to *XOR32* block where XOR operation is made with *P* block output data. The result data input in *reg32* block and after result of the operation reaches the bottom block of the establishment, then after the process resumes;
- in the same time, the data is reaching by entry of *e* and the entrance of *mux32* block from the upper scheme;
- the output data *XOR32* block arrive in the *inverse_IP* block that makes the reverse operations of *IP* block;
- after the 16th round of the output data *inverse_IP* block is encrypted or decrypted data properly chosen operation (encryption or decryption).

6. RESULTS

After the implementation of TDES algorithm on XC3S400 FPGA the following performances of the TDES circuit were obtained:

Minimum period: 14.022ns (Maximum Frequency: 71.318MHz; minimum input arrival time before clock: 15.030ns; maximum output required time after clock: 18.844ns; maximum combinational path delay: 19.852ns

In the Table I the aria and logic utilization, after the implementations of TDES algorithm, are shown.

Table I. Structure utilization for TDES module

Logic used	**used**	**avalaible**	**Area (%)**
Number of Slices	958	3584	26
Number of Slice Flip Flops	237	7168	3
Number of 4 input LUTs	1828	7168	25
Number of IOs	261		
Number of bonded IOBs	245	264	92
Number of GCLKs	1	8	12

This module was implemented and optimized for used area and not for the speed processing. However, the circuits can encrypt/decrypt with 4.8Gb/s at a 958 number of slices used.

In Table II other implementations of TDES module on reconfigurable structures are presented [7][8].

Table II. Structure utilization for TDES module for other hardware implementations

Algorithm	Author	Device used	Nr. used slice	Max. frequency
TDES	Free-DES	XCV400	5263	47.7
	McLoony, McCanny	XCV1000	6446	59.5
	Patterson (Jbits)	XCV150	1584	168
	Prasun Ghozal	XC3S1000	1585	-

7. CONCLUSIONS

In this paper was proposed to achieve a optimized implementation on low-cost FPGA (XC3S400) The algorithm implementations take a reduced area, practically a number of 958 slices from reprogrammable structure. Size can be reduced further by using DES block once.

Represented solution for generating subkeys block was to redesign thus subkeys are not calculated in each round; in this case everything was generated before the beginning of the algorithm. TDES has also attached an input port that select the order that will apply to subkeys (default pin select encryption or decryption).

In the immediate future, the presented encryption algorithm will be compared with other algorithms, including bio-inspired ones based on cellular automata (CA) theory.

8.ACKNOWLEDGMENT

This work was supported by CNCSIS UEFISCSU, project number PN II-RU PD 369/2010, Contract No. 10/02.08.2010.

9. REFERENCES

[1]. R. Chaves, B. Donchev, G. Kuzmanov, L. Sousa, S. Vassiliadis, “BRAM-LUT Tradeoff on a Polymorfic DES Design”, Springer-

Verlag Berlin Heidelberg, HiPEAC, pp. 55-65, 2008

[2]. P. Kitsos, N. Sklavos, M.D. Galanis, O. Koufopavlou, "64-bit Block ciphers: hardware implementation and comparison analysis", Computer and Electrical engineering 30, pp. 593-604, Elsevier, 2004

[3]. Anghelescu, P., Ionita, S., Sofron, E., "Encryption Technique with Programmable Cellular Automata (ETPCA)", Journal of Cellular Automata, ISSN 1557- 5969, Volume 5, Issue 1-2, 2010.

[4]. N. A. Saquib, F. R. Henriquez, A. D. Perez, "A Compact and Efficient FPGA Implementation of the DES Algoritm", ReConFig'04, Colima, Mexico, September 20-21, 2004

[5]. McLoone, M., McCanny, "High-performance FPGA implementation of DES using a novel method for implementing the key schedule". IEE Proc.: Circuits, Devices & Systems 150, pp. 373–378, 2003

[6]. Xilinx. Sparan-3 programmable gate arrays data sheet, from http://www.xilinx.com.

[7]. G. Rouvroy, F. X. Standaert, J. D. Legat – "Design strategies and modified descriptions to optimize cipher FPGA implementations: fast and compact results for DES and Triple-DES", Lecture Notes in Computer science, Springer, ISBN: 978-3-540-40822-2, pp. 181-193, 2003.

[8]. Prasun G., Malabika B., Manish Biswas., "A Compact FPGA Implementation of Triple-DES Encryption System with IP Core Generation on On-Chip Verificatioin", Proceedings of the 2010 international Conference on Industrial engineering and Operations Management, Dhaka, 2010