Kurdistan Journal of Applied Research (KJAR)
Print-ISSN: 2411-7684 | Electronic-ISSN: 2411-7706

Website: Kjar.spu.edu.iq | Email: kjar@spu.edu.iq

Implementation of Simplified Data Encryption Standard on FPGA using VHDL

Salim Qadir Mohammed
Communication Department
Technical College of Engineering
Sulaimani Polytechnic University
Sulaimani, Iraq
Salim.muhammed@spu.edu.iq

Article Info

Volume 7 – Issue 1- June 2022

DOI: 10.24017/Science.2022.1.2

Article history:

Received:15/12/2021
Accepted:14/03/2022

Keywords:

Cryptography, DES, S-DES, FPGA, VHDL

ABSTRACT

Due to enormous development in communication devices, globally internet-connected networks are largely used in all human activities. The security of information has been becoming a major concern for all users and clients, who depend on the network system. Cryptography has played a significant role to combat these challenges and improve confidentiality, integrity, and authentication of data communication in the network. The Data Encryption Standard (DES) is one of the most familiar types of cryptography. The selection of hardware implementation tools is important for researchers and academics to design and build prototypes of their model proposals efficiently. Therefore, this paper, focuses on hardware implementation of a Simplified DES (S-DES) model with minimum FPGA resources in terms of Configurable Logic Blocks (CLB). This is to reduce the complexity and the number of logic elements used in the design, which in turn leads to reduction in the latency and improvement in throughput of the system. The S-DES program has been deployed using Quartus II software environment using VHDL language. S-DES consists of symmetric encryption, decryption, and key generation blocks. The S-DES has been successfully synthesized, compiled, and implemented on Altera - Cyclone IV- 4CX150 FPGA device. The implementation of the S-DES for two rounds required only 32 CLB which is supersedes the available implementation in the literature.

1. INTRODUCTION

The protection of confidential information is fundamental demand in recent applications like financial transactions, e-mail clients, online audio/video meetings, using authentication techniques. The authenticate network internet-connected needs some ways, to guarantee the authorized people have the right to get data transferred through an insecure channel. Therefore, to achieve this mission to provide a secure system; cryptography is required [1].

The definition of Cryptography is originated from Ancient Greek, which means “hidden”. It is

a science or knowledge that keep information system secure and safe by transform messages into a particular form to maintain authentication through transmitting an unsafe medium. The cryptographic is the procedure to change the plaintext into ciphertext and get it back into plaintext, by using some mechanisms, which are called the encryption and decryption methods. In contrast, cryptanalysis is the attempt to find vulnerabilities and weaknesses in a cryptographic algorithm. Many or most applications nowadays are used cryptography in the authentication process, for example, credit cards, wireless cell phones, online shopping, IP prepaid TV, etc. Cryptography is always needed in many control applications, such as car-control devices, smart elevators, electrical trains, etc.

The most famous type of cryptography is the Data Encryption Standard. The DES is an algorithm used to encrypt the data for keeping it immune from unauthorized persons. DES is firstly presented and developed by IBM in 1970, and officially authorized in the United State of America in,1977 by the National Institute of Standards and Technology (NIST). DES is widely employed for various applications, especially, confidential information is essential. Despite, it has been replaced by the Advanced Encryption Standard (AES), since 2001.But practically is still dependent on it for many hardware or program applications [2]. Cryptography can be implemented either by using a software program or hardware tools. FPGA is one of the efficient hardware tools, which can be programed by VHDL (Very High-Speed Integrated Circuits Hardware Description Language) language [3].

High security of data transferred can be achieved with using cryptography techniques, that required a complex design and high cost.DES can be implemented with software or hardware, like FPGA and ASIC.Hardware design has advantages of high throughput and faster than software model.DES is still used in some applications and the triple DES is widely use nowadays. Also, DES is seen as based for other symmetric algorithms and actually it is the best model to explain the symmetric encryption algorithm [4]. So, using DES provides a considerable level of security with simple design and cost effective.

In this paper, simplified data encryption system, which consists of encryption, decryption and keys generation are designed with codes, that written in VHDL language, then synthesized and converted into proprietary interconnect description by usually vender provided synthesis tools. The output file that contains the interconnect description is called a bit stream. The design is accomplished by downloading via JTAG (Joint Test Action Group) cable into FPGA device offered by Altera, Cyclone family "Cyclone IV 4CX150FPGA". The rest of this paper is organized as follows. Section 2 presents the literature review and highlights the objectives of the paper. Section 3, describes Field Programmable Gate Array (FPGA) idea and its utilization and its implementations. Section 4, explains the methodology basis of the proposed system in detail, and shows the system's architecture. Section 5 illustrates the experimental results. Section 6 presents the conclusion.

2. LITERATURE REVIEW

Ke Wang [5] in 2009 presented a proposal model of DES that was implemented on Xilinx Virtex4 FPGA for two rounds. His design has successfully implemented the Data Encryption Standard. The proposed DES algorithm of [5] has been implemented on four different FPGA family boards, without given information of the number of logic blocks that required to build this model.

Prasetyo K.N. et al. [3] in 2014 proposed the Blowfish data encryption model used for the internet of things (IoT) and implemented it on Xilinx Virtex4 FPGA.The result showed a better outperform. The symmetric key size of the Blowfish algorithm was varied from 448 bits to 384 bits, 320 bits, 256 bits, 128 bits, and 64 bits. Due to limited FPGA resources, it is impractical to implement this algorithm with these length of keys on FPGA hardware.

Purvi Garg et al. [6] in 2015 presented Cryptanalysis of Simplified Data Encryption Standard based on Genetic Algorithm. Their outcomes showed, that the Genetic Algorithm has higher

performance than Brute Force to analyze S-DES, but without hardware implementation on FPGA device.

Soufiane, and Seddik [2] in 2015 presented High throughput FPGA Implementation of Data Encryption Standard using Time Variable Sub-Keys technique. The proposed design was implemented on Xilinx FPGA Board. DES model accomplished a data rate of 9453.47 Mbps using 2046 CLB slices. Their results exhibited that the designed execution was fast hardware implementations with considerable security. However, the design required a large number of 2046 of FPGA CLB slices, in addition to synchronization issue between sender and receiver.

Pandey J. G. et al. [1] in 2016 proposed DES design and implemented on Xilinx Virtex5 FPGA. DES model has been achieved efficiently and showed good performance with few numbers of CLB slices. S-boxes are the only nonlinear part in the DES algorithm, that provide confusion, which is essential to build a strong and secure encryption algorithm. Simple or analytical methods may have negative impact on security of encryption algorithm.

Chabukswar P. M. et al. [7] in 2017 proposed an enhanced model of DES by using four different methods for key generation. Multiplexers are used to execute S-Boxes in substitution of f-functions. The design has been implemented on Xilinx Virtex FPGA.Dynamic key generation consists from four types of keys (Direct, LFSR, Chaotic and 2S' complement keys) for DES encryption, which is used in short term applications. The design of [7] was complex and suffered from synchronization issues when the length of the original key kept 64 bits.

Oukili S. et al. [8] in 2017 presented high throughput FPGA implementation of AES using parallel processing on Xilinx Virtex6 FPGA. The AES model achieved a data rate of 79 Gbps using 4830 CLB slices. Their results showed fast hardware implementations with considerable security. The S-boxes are most crucial element of AES implementation. Using combinational logic gates to design S-boxes must be uncompromised with achieving high security of algorithm.

Kristianti V. E. et al. [9] in 2018 proposed DES design by using 8 round algorithms based on a system on chip (SoC). The proposed DES design has been implemented on XC3ES500E FPGA. DES implementation has shown that, only 9.7% of resources were required while it needed 21.2% resources in conventional method design with 16 rounds. In spite reduction of resources usage for 8 rounds, but the design still consumes large amount of 374 slices.

Zeebaree S. R. M. et al. [10] in 2019 proposed high throughput parallel /sequential Simplified Data Encryption Code Breaker when was implemented on Xilinx FPGA. They proposed two models one with a sequential single processor element and the second model with parallel 512 parallel processors elements. Their result showed that the second model is very fast toward breaking the code compared to the first one. The proposed algorithm has been implemented on FPGA hardware that contains 512 parallel processor elements and utilizes 3816 CLB slices.

Arnab Roy et al. [11] in 2020 proposed a triple data encryption standard (TDES) algorithm combined with linear feedback shift register (LFSR) and the design is implemented on a pipelined FPGA board device. LFSR is used to increase security of the system by randomized the symmetric keys for both encryption and decryption parts. A Parallel pipelined improves throughput of the proposed approach. The proposed model is written by Verilog HDL code and the program is implemented on FPGA board from Vertex-7 device provided by Xilinx. The authors have showed, that the parallel pipelined implementation of pseudo LFSR with TDES has a better performance than non-pipelined model in terms of throughput, but required more resources. The pseudo-randomized pipelined TDES required 6220 LUTs slices.

Sara M.H. et al. [12] in 2021 proposed an approach to concatenate the international data encryption algorithm (IDEA) with AES to improve security of AES.The proposed model is implemented in real -time FPGA device offered by Xilinx. The design model consists two types of encryptions, IDEA and AES together, but increases the complexity of the model which requires many CLBs.

AL-Salam H.I.M.et al.[13]in 2021 presented different models of cyber security systems like single, dual, and quad types. Models are mainly consisting three major parts, UART, FIFO and DES algorithm. The FPGA implementation of models have

showed a better performance of quad model among others. The presented model consumes large amount of FPGA resources, it required 27962 logic elements.
As mentioned previously in literature review, the FPGA resources can be considered as one of the most crucial factors to obtain low cost, high performance, reconfigurable and robust algorithms. Therefore, this paper has focused on optimum usage of FPGA resources, through providing efficient VHDL codes for the S-DES algorithm. Achieving necessary properties, that are required to build a strong ciphers algorithm, which includes confusion and diffusion by using permutation operations for many blocks. At the same time keeping the nonlinearity of S-boxes to improve security of the algorithm.

3. FIELD PROGRAMMABLE GATE ARRAY (FPGA)

FPGAs' core is usually consisting of an array of programmable logic cells and programmable interconnect matrix. An FPGA array is surrounded with programmable I/O (input /output) cells. FPGA is capable of reprogramming after manufacturing and installing in the fields [14]. Figure 1 gives basic architecture of FPGA.

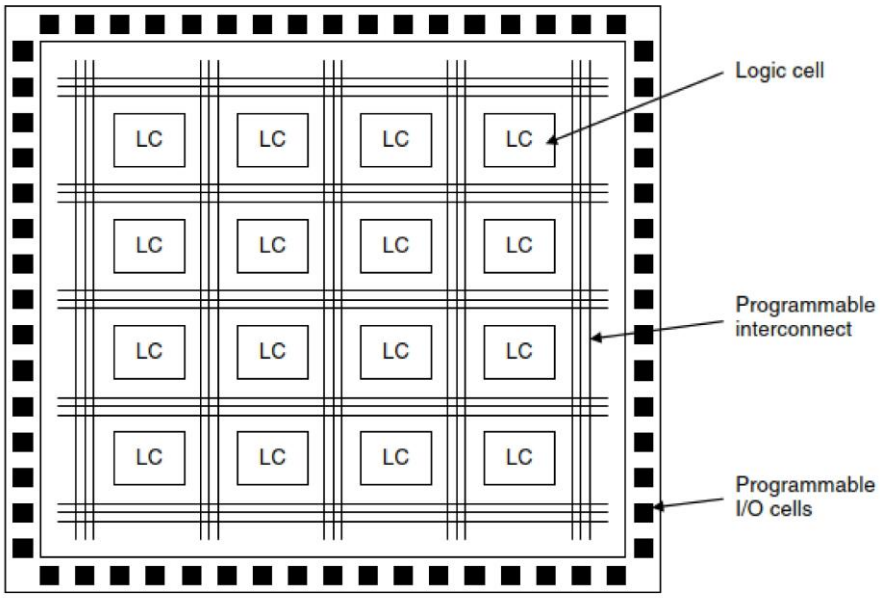

Figure 1: Generic FPGA architecture [14]

Logic block or function block is actually comprising a look-up table (LUT) and a D flip-flop to provides construction logic gate circuits as shown in Figure 2. CLB is based on one or more LUT to store a binary logic (0 or 1) in SRAM (static RAM) [15]. The program codes have been written in VHDL or Verilog languages are converted through software provided by different vendors into configuration output file called a bit stream. This bit stream can be downloaded into FPGA device by JTAG cable to store in SRAM memory or flash memory.

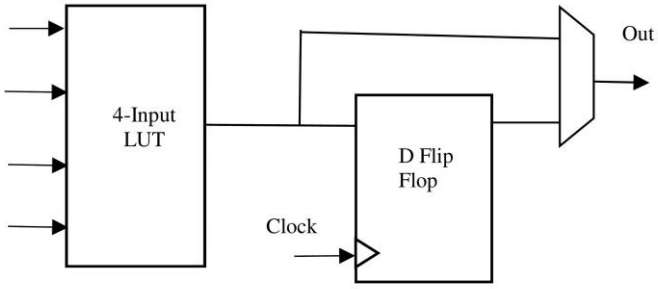

Figure 2: Look-up table (LUT) and a D flip-flop

For many of latest FPGA devices, both bit-serial and bit-parallel configuration modes are supported. The leading FPGA manufacturers include Xilinx, Altera, Actel and Cypress Semiconductor. FPGA is enabling to implement digital signal processing applications with optimum facilities that is required a high throughput compared to other digital signal processing processors. FPGAs ability of redesigning in hardware, provides a unique specialty for hardware implementation in many DSP (digital signal processing) applications [16]. Conventionally, technicians who want to design DSP using FPGA hardware implementation by HDL language such as Verilog HDL and VHDL, which can be achieved without writing any VHDL codes through using DSP Builder. This tool enables designers to work in a MATLAB environment, then program the FPGA boards by codes that have been generated by the DSP Builders.

Altera's DSP Builder software offers interaction between Simulink and FPGA hardware as shown in Figure 3. It facilitates hardware implementation of DSP purposes, runs a system-level verification software to the designer who is not familiar with VHDL design codes, and lets the designer to execute DSP applications in FPGAs without HDL.

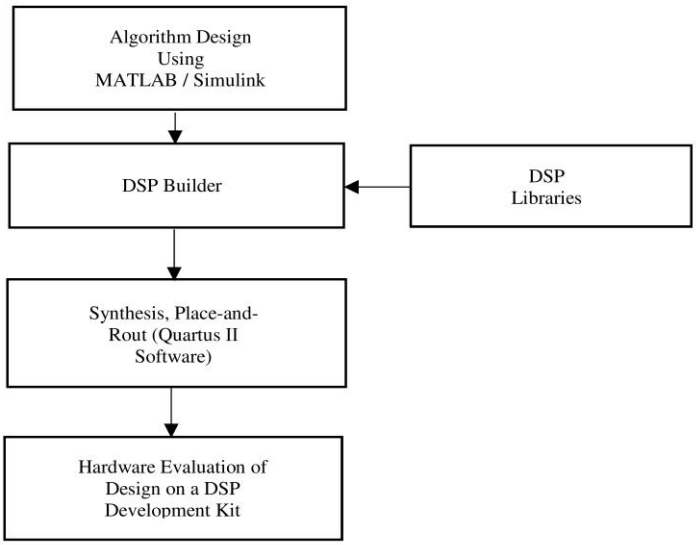

Figure 3:DSP Builder Altera FPGA [16]

4. METHODOLOGY

Edward Schaefer had introduced the simplified data encryption standard, for the first time at Santa Clara University. The S-DES is a symmetrical encryption class derived from the Data Encryption Standard technique which, is generally based on Feistel's approach proposal [13]. The S-DES was for simplification of the DES principles and educational activities. So, it is not quite secure for application purposes. In the Simplified Data Encryption Standard method, a block of 8-bit of plaintext is has taken with a 10-bit of the key as an input, that would be processed into digital binary. The plaintext and the private key, which has been converted and, then portioned into data block form with 8-bit of ciphertext length as output. This conversion process is called encryption. To obtain the transmitted plaintext, the decryption process has been done. The decryption algorithm is achieved, when the ciphertext is received with the same private key, that must send to both sender and receiver simultaneously in a secure channel by repeating all steps of the encryption algorithm with reversing the order of subkeys used only. In the receiver the decryption has been done, when 8-bit block ciphertext is provided with the 10-bit key, the result is the transmitted plaintext with a length of 8-bit as output [4]. If the plaintext is in different forms like alphabet or symbols, it must be changed at first into a decimal format or hexadecimal format by using ASCII Code table, then after that, it has been transformed to the binary format [10]. The complete process of encryption and decryption algorithms can be shown in Figure 4[4].

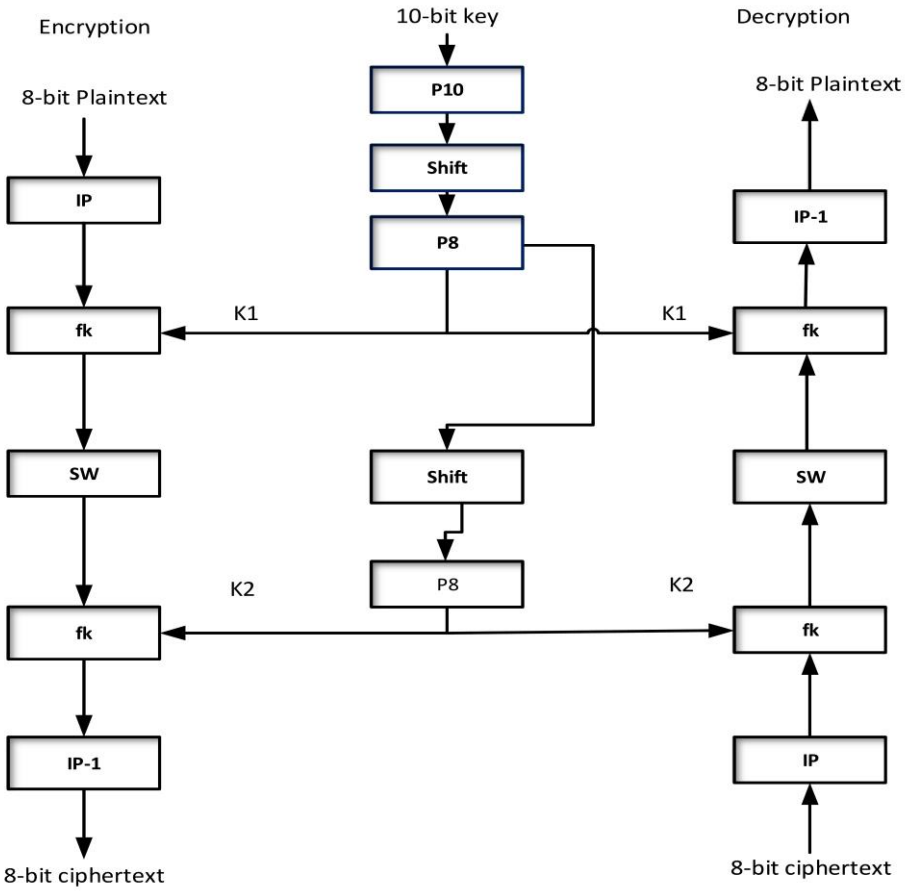

Figure 4: S-DES [4]

4.1 The encryption of (S- DES) Algorithm

The encryption procedure of the S-DES algorithm, basically can be divided, into five steps, which can be seen in figure 4 as the followings:

1- An initial permutation (IP), it is nothing more than just change the positions of input bits in order to randomize the plaintext data bits to increase the security of the message.
2- A complex function block, which is labeled fk, has been included some operations, like XORed, permutation, and substitution operations and depends on sub keys that are provided by the key generation block.
3- The Swap stage, that switches (SW) the two halves of the data, L1 (the left most significate bits) and R1 (the right least significant bits).
4- The complex function block for the second round.
5- The inverse of the initial permutation (IP^{-1}) stage.

4.2 The architecture of f-Function (fk)

The function block fk is a complex configuration of simplified data encryption standard algorithm, consisting of the following components, expansion- permutation, XORed function, substitution functions-choice (S-box), permutation as can be seen in figure 5.

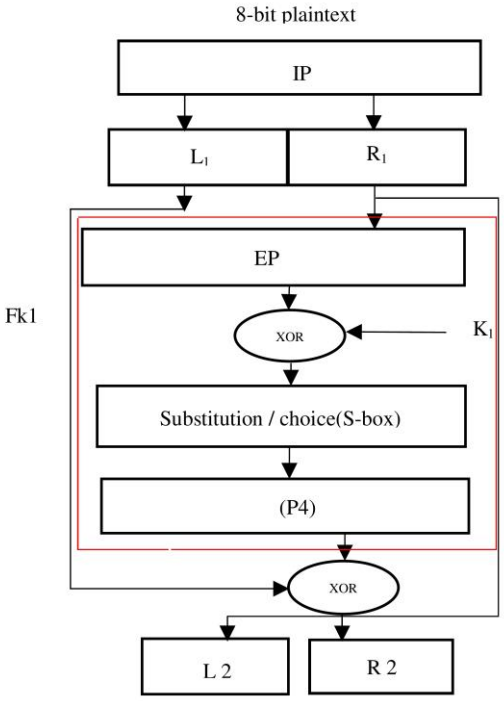

Figure 5: S-DES Encryption fk details

The operations are done in the function fk would be as followings:
Let L0 and R0 are the input plaintext after separated it into 4-bit left and 4-bit right for each one. The R0 block esteem would be entered to the expansion-permutation block to apply expansion from 4-bit into 8-bit and permutation process due to this sequence.

1. Apply expansion/permutation E/P input = [1 2 3 4]
E/P output = [4 1 2 3 2 3 4 1]

2. The E/P output is XORed with sub key1 that is received from the key generated process, the result will be passed into the substitution-choice S-Box block
3. The S-DES perform substitutions using S-Boxes, where it considered as a square matrix with 4-element by 4-element dimensions. The input is used to select row and column of the selected element outputs from S_0 and S1 according to the S-Boxes tables below, where the input Bit1bit4 specifies row (0,1,2,3 in decimal) and Bit2bit4 specifies the column (0,1,2,3 in decimal). The two-bit output of each S-boxes (S_0 and S1) are concatenated to be the input the next permutation with length of four bits.

$$S_0 = \begin{vmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{vmatrix} \qquad S_1 = \begin{vmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{vmatrix}$$

4. The last block will do permutation for input = [1 2 3 4]
   Apply permutation P4 output = [2 4 3 1].
5. The output of permutation P4 is XORed with the L_0 and the result is the left half L_1 with 4-bit length (MSB) of the f-function fk1 of the first stage and second half would be the same right part of the data output from (IP) initial permutation R_0 with 4-bit length (LSB), it will be denoted by R_1.

4.3 Key Generation Block

The key generation process is the main crucial part in the encryption algorithm procedure as whole. In general, the main key can be provided by many several ways for example, designed by using linear feedback shift register (LFSR) and send this private key to both sender and receiver through a secure channel. This main key, which has length of 10-bit is used to generates subkeys as follows:

The 10-bit private key is at first permutated with P10 permutation operation the result would be P10 output= [3 5 2 7 4 10 1 9 8 6]. The output of P10 permutation has been separated into L, where the most significant bit (MSB) and R, where the least significant bit (LSB) and each has five-bit length. The L and R block are rotated shift left by one bit, then concatenated together to passed into P8 permutation block. The P8 permutation output would be P8 output = [6 374 8 5 10 9]. The output of P8 permutation has been considered as subkey1(k1) generation, which is used in first round in fk function. The L block and R block after rotated shift left by one bit passed into another rotated shift left, but with two bits. The output of both L and R blocks after shifting by two bits concatenated together and subjected as input to the P8 permutation block. The output of P8 permutation block P8 output = [6 374 8 5 10 9] is the second subkey2 generated (k2), where used in the second round of fk function. The block diagram of key generation circuit can be seen in the Figure 6.

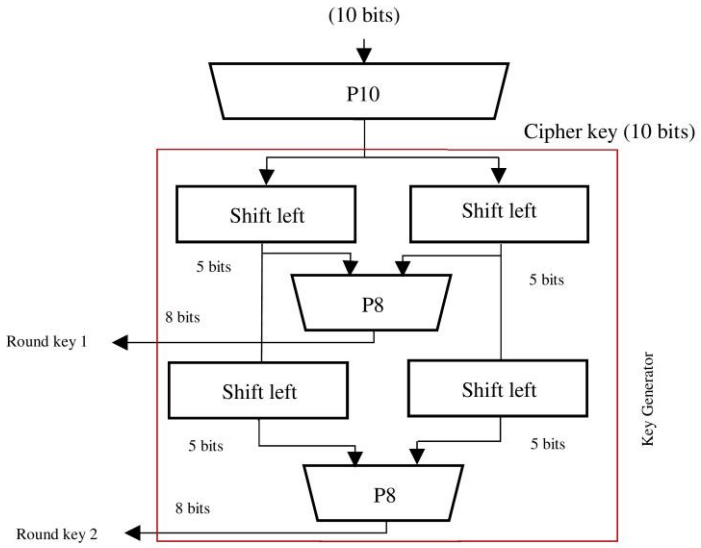

Figure 6: Key generation circuit

4.4 The Encryption / Decryption Operation

The input plaintext with length of 8-bit is encrypted or decrypted through procedure consisted from five stage has been mentioned previously, which are initial permutation (IP), the f-function fk1, swap operation, the second f-function fk2 and, finally inverse permutation (IP-1). The operation of f- function fk is has been illustrated in 4.2 section, therefor the process of the encryption is as the following:

The 8-bit plaintext message is basically, permutated with initial permutation (IP) and the output of this stage would be IP output IP= [2 6 3 1 4 85 7]. The output of (IP) initial permutation is passed into f-function one fk1 the outputs of this function are L1 and R1 halves blocks with 4-bit each, then entered the swap block to switching it from left to right and vice versa. The same operation is done in the second f-function fk2 and the output of this stage is concatenated then, passed into the inverse permutation (IP-1). The output of inverse permutation (IP-1) is

(IP-1) output = [4 1 3 5 7 2 8 6], where the output of this block is represents the ciphertext message with 8-bit length need to send it with the private key to the receiver.

In general, for each round the process would be as the followings:

$Li = Ri{-}1$ (1)

$Ri = Li{-}1 \oplus F(Ri{-}1, Ki)$ (2)

This mean, the left half of next round is the right half of previous round and the right half of the next stage is the result of left previous round XORed with f-function of right half of previous stage with subkey of this stage [5].In the receiver in order to decrypt the received message the same procedure, which has been done in the encryption algorithm need to repeated ,but with one exceptional thing. The decryption algorithm is identical to the encryption algorithm, but with the reverse order of subkeys, which mean in the first f-function the second subkey2 (k2) is used at first, then in the second f-function the first subkey1 (k1) is used as clearly seen in Figure 4.

5. RESULTS AND DISCUSSION

The proposed design architecture of simplified data encryption standard as shown in Figure 4, is implemented in VHDL design language, and synthesized using Altera DE2i-150

Development and Education Board, Altera Cyclone IV 4CX150FPGA device. As seen in Figure 7.

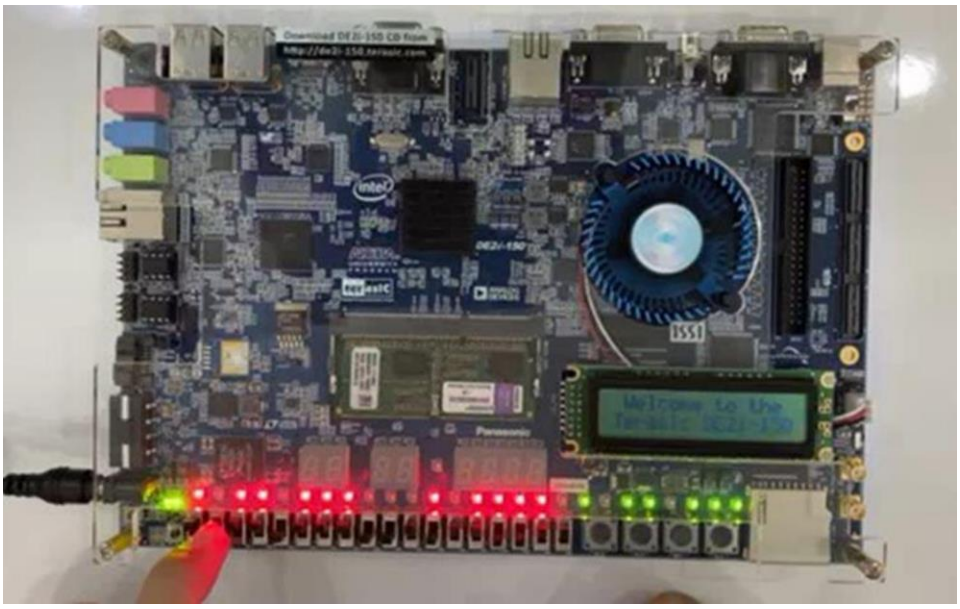

Figure 7: Implementation of S-DES on FPGA Board

5.1 FPGA Board Results

After the FPGA device has been programmed and configured to implement the designed S-DES. The required configuration file is generated by Quartus II Complier. Altera's DE2 board allows a configuration to be done in two different ways, known as JTAG and AS modes. The programming has been done by using JTAG USB cable. The eight-red led (8 RED LED) in left side has been used as sender plaintext, while in the middle is ten red led has represent the private key and the 8-green led (8 GREEN LED) in right side it is the received plaintext after the decryption algorithm is used. It is obvious the same message has been received even when, the key was changed. The FPGA device summary report for number of logic elements and other components are used in the proposed S-DES architecture is given in Figure 8.

Table of Contents

Compilation Report - s_des_enc_dec

Flow Summary	
Flow Status	Successful - Thu Dec 31 22:32:39 2020
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	s_des_enc_dec
Top-level Entity Name	s_des_enc_dec
Family	Cyclone IV GX
Device	EP4CGX150DF31C7
Timing Models	Final
Total logic elements	32 / 149,760 (< 1 %)
Total combinational functions	32 / 149,760 (< 1 %)
Dedicated logic registers	0 / 149,760 (0 %)
Total registers	0
Total pins	42 / 508 (8 %)
Total virtual pins	0
Total memory bits	0 / 6,635,520 (0 %)
Embedded Multiplier 9-bit elements	0 / 720 (0 %)
Total GXB Receiver Channel PCS	0 / 8 (0 %)
Total GXB Receiver Channel PMA	0 / 8 (0 %)
Total GXB Transmitter Channel PCS	0 / 8 (0 %)
Total GXB Transmitter Channel PMA	0 / 8 (0 %)
Total PLLs	0 / 8 (0 %)

Figure 8: FPGA device summary report

5.2 The Simulation Result

Simplified Data Encryption Standard has been implemented with VHDL and simulated in RTL level, which shown in Figure 9, with Altera Cyclone IV 4CX150FPGA device on Quartus II Simulator.

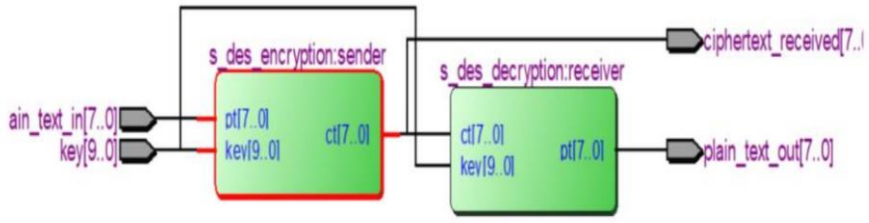

Figure 9: RTL level of S-DES algorithm

Simulation result for encryption and decryption is shown in Figure 10. The simulation has been achieved on Quartus II Simulator and running the simulation on Run Functional Simulation Mode. As shown in Figure 10, the received plaintext is arranged to be under the sender plaintext in the simulation waveform editor, in order to be easier for observation, while the received ciphertext is putted under the received plaintext and the private key at the bottom. Simulation has achieved for several sending plaintexts with different keys and the received ciphertexts have been successfully decrypted to get back the plaintexts.

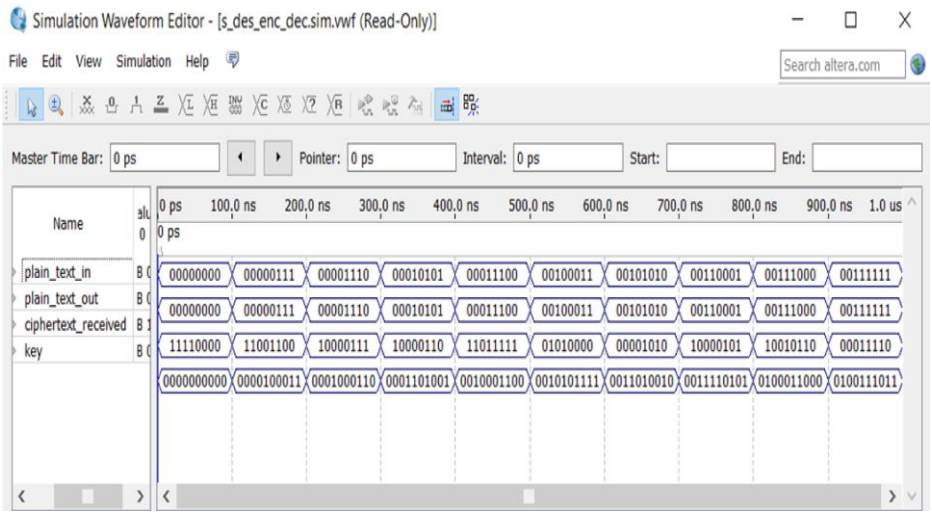

Figure 10: Simulation Result for S-DES algorithm

6. CONCLUSION

In this paper, S-DES algorithm is implemented for encryption plaintext in transmitter and decryption of ciphertext in receiver side. Keys generator are designed separately, it was used in both encryption and decryption operations. f-function is used as a component in both encryption and decryption designs. The substitution choice block in f-function is required in the simplified S-DES algorithm, which is designed by using multiplexers based on look-up tables in Altera FPGA Quartus II software environment. The S-DES is worked successfully when the design executed practically on FPGA board. Due to an efficient VHDL programing, only 32 logic elements were required to build this model, which means using less hardware resources.

REFERENCE

[1] J. G. Pandey, A. Gurawa, H. Nehra, and A. Karmakar, "An efficient VLSI architecture for data encryption standard and its FPGA implementation," 2016, pp. 1-5: IEEE.

[2] S. Oukili and S. Bri, "FPGA implementation of Data Encryption Standard using time variable permutations," 2015, pp. 126-129: IEEE.

[3] K. N. Prasetyo, Y. Purwanto, and D. Darlis, "An implementation of data encryption for Internet of Things using blowfish algorithm on FPGA," in ICoICT, 2014, pp. 75-79: IEEE.

[4] C. Paar and J. Pelzl, Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media, 2009.

[5] K. Wang, "An encrypt and decrypt algorithm implementation on FPGAs," in Fifth International Conference on Semantics, Knowledge and Grid, Zhuhai, China, 2009, pp. 298-301: IEEE.

[6] P. Garg, S. Varshney, and M. Bhardwaj, "Cryptanalysis of simplified data encryption standard using genetic algorithm," American Journal of Networks and Communications, vol. 4, no. 3, pp. 32-36, 2015.

[7] P. M. Chabukswar, M. Kumar, and P. Balaramudu, "An efficient implementation of enhanced key generation technique in data encryption standard (DES) algorithm using VHDL," in ICCMC, Erode, India, 2017, pp. 917-921: IEEE.

[8] S. Oukili and S. Bri, "High speed efficient advanced encryption standard implementation," in ISNCC, Marrakech, Morocco, 2017, pp. 1-4: IEEE.

[9] V. E. Kristianti, E. P. Wibowo, A. Pertiwi, H. Afandi, and B. Soerowirdjo, "Finding an Efficient FPGA Implementation of the DES Algorithm to Support the Processor Chip on Smartcard," in EIConCIT, Makassar, Indonesia, Indonesia, 2018, pp. 208-211: IEEE.

[10] S. R. M. Zeebaree, A. B. Sallow, B. K. Hussan, and S. M. Ali, "Design and Simulation of High-Speed Parallel/Sequential Simplified DES Code Breaking Based on FPGA," in ICOASE, Kurdistan Region,Iraq, 2019, pp. 76-81: IEEE.

[11] A. G. Arnab Roy, Deva Nand, "FPGA Implementation of a pipelined and pseudo-randomized TDES algorithm," IEEE vol. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 171-176, 2020.

[12] G. G. H. Sara M. Hassan, "Real-time FPGA implementation of concatenated AES and IDEA cryptography system," Indonesian Journal of Electrical Engineering and Computer Science (ijeecs), vol. 22, no. 1, pp. 71-82, April 2021.

[13] H. I. M. Al-Salman, Ehkan, P. and Al-Doori, M.H., "FPGA-based Design of Multiple Models for Industry 4.0 Cyber Security," Journal of Physics: Conference Series vol. 1997, no. 1, p. 012031, 2021. IOP Publishing

[14] I. Grout, Digital systems design with FPGAs and CPLDs. Elsevier, 2011.

[15] Y. Zhou, "Novel very fast FFT processors: on DSP algorithm design and FPGA-based implementation," 2006.

[16] F. R.-T. Reconfiguration, "White Paper FPGA Run-Time Reconfiguration: Two Approaches," 2008.