# Application and Implementation of DES Algorithm Based on FPGA

Liuwei Zhao[1, a*] and Yanbin Zhang[1, b]

[1]School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China

[a]913408244@qq.com, [b]z_liuwei@163.com

**Keywords:** Pipeline architecture; FPGA; DES; IP Core

**Abstract.** In order to meet the data transmission of security and real time requirement in trading system, in this paper, a design scheme of 16 level pipeline encryption system based on FPGA is proposed. Using the 7K325T model of the Xilinx company`s FPGA chip to achieve circuit design. The results show that the circuit can work stably, the maximum frequency of the encryption module is up to 166MHz, and the encryption speed is 10.62Gb/s, which can ensure the security and efficiency of the data.

## Introduction

In the options, futures and other derivatives transactions, the transaction of real-time data processing speed has more and more high requirements. Normally, in order to improve the security of data transactions, transaction data are required to do encryption, and this step to deal with the time occupied by the larger proportion of the entire system. In addition, customers have more and more high requirements for the trading system delay and throughput. When dealing with massive amounts of data, the TCP/IP protocol will occupy a large amount of CPU resources. Therefore, with the emergence of Gigabit Ethernet, network link speed is higher than the processing speed of TCP/IP protocol, the real-time data processing can`t be guaranteed [1].

At present, the mainstream of data processing is realized by hardware acceleration. While using GPU and multi-core processor to achieve the acceleration scheme in dealing with transaction data is not only delay high but poor stability, has can`t meet the needs of transaction. According to the structure of FPGA and the characteristics of parallel processing data, we can know that using FPGA can effectively reduce the delay and power consumption [2]. Foreign countries have adopted the FPGA platform to achieve the acceleration program, the minimum delay can be achieved within 1us,

With a good application value.

DES algorithm is a packet with 16 iterations of symmetric ciphers, not only has the same structure with the encrypt and decrypt, but also involves only logical and LUT operation [3]. Therefore, Using FPGA to achieve the encryption algorithm has become relatively simple. In this paper, we put forward a kind of encryption scheme based on FPGA 16 level assembly line, and use the hardware boards to be verified.

## Encrypted Principle

The 64-bit of plaintext are divided into 32-bit left and 32-bit right by the initial permutation, the transformation process is shown in Fig. 1. In the iterative process of I cycles, on the right of the $R_i$ under the action of the 48-bit sub key to do F transform, and the data obtained do XOR operation with $L_i$, $R_i$ is directly used as the next round of the $L_{i+1}$[4]. Mathematical description such as the Eq.1.

$$\begin{cases} L_i = R_{i-1} \qquad 1 \leqq i \leqq 16 \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \ 1 \leqq \mathrm{i} \leqq 16 \end{cases} \tag{1}$$

In the iteration process, F function is the part of iterative operation [5]. Its process is the first to extend the 32-bit $R_{i-1}$ to 48-bit data, then do XOR operation with sub key $k_i$, and the 48-bit data are replaced by 32-bit by S-box. Finally, the final result can be obtained by the replacement of P. Form of function as shown in the Eq.2.

$$f(R,K) = P\big(S(E(R) \oplus K)\big) \tag{2}$$

The extended E operation is to extend the 32-bit data to 48 bits.

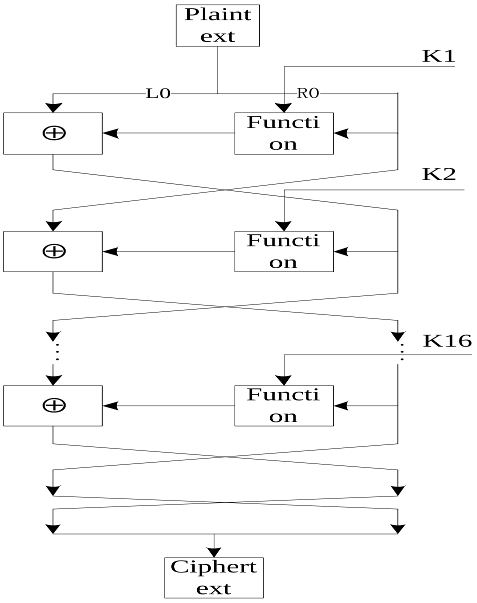


Figure 1. DES algorithm encryption process

S-box as the only non-linear transformation in the DES algorithm, which is good or bed will affect the performance of DES algorithm. The operation process is related to eight S-boxes, and the function of the S-box is to convert 6-bit of input data to 4-bit of output data. Each S-box is a fixed 4 rows and 16 columns table which contain 64 numbers. The operation process as shown in the Eq.3.

$$s_i(x_1 \dots x_6) = s_i((x_1x_6)_2, (x_2 \dots x_5)_2) \tag{3}$$

## The Design of the Encryption System

Combined with the actual situation and the design requirements, we propose a modularity of encryption scheme. The circuit is divided into PCIe [6, 7] module, TOE module, Encryption module, Ethernet MAC. TOE module, PCIe module and Ethernet MAC are implemented by IP core on the FPGA. The overall design process is shown in Fig. 2.

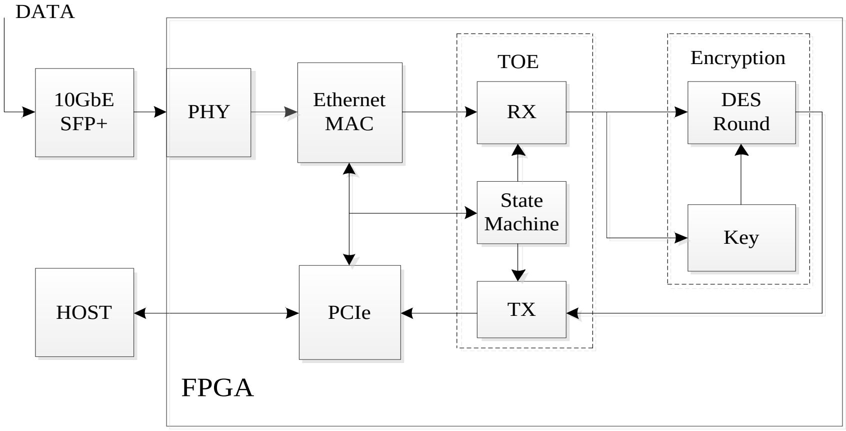


Figure 2. General block diagram of system design

The main process is that the data on the network is transmitted to the FPGA through the optical module, and send to encryption module after TCP/IP protocol processing. When the encryption module encrypts data, sub-module generates the sub-keys with the pipeline to the DES Round module. The encrypted data is processed by the TCP/IP protocol and sent to the PCIe module, then transmit to host for read and storage.

**DES Round Module.** In the 16 rounds of iterative design, I mainly consider three implementations. The first scheme is time division multiplexing access, we can put a loop structure or a plurality of rounds as a multiplex unit, and call the reuse unit to achieve the DES algorithm under the clock control. The second is the use of pipeline structure. In other words, we can open all of the 16 lines of DES algorithm, and make each round as a pipeline. The third is to use a combination of circuit design, which the 16 times of iterative process to achieve by combinational logic circuits.

Time division multiplexing access scheme using the speed in exchange for the area, which has less resource occupancy rate compared with the pipeline architecture, so it does not meet the design requirements. Combinational circuit has the longest critical path and the maximum delay compared with others. Therefore, in order to enhance the rate of encryption, we adopt the pipeline structure design circuit.

In the iterative process of 16 rounds, each round of iteration process exactly the same, in this paper, only the first and the second round of iteration are given, as shown in Fig. 3.Encryption and decryption process is similar, but the use of the sub key are the opposite. Each iteration consists of XOR, Function module and register.

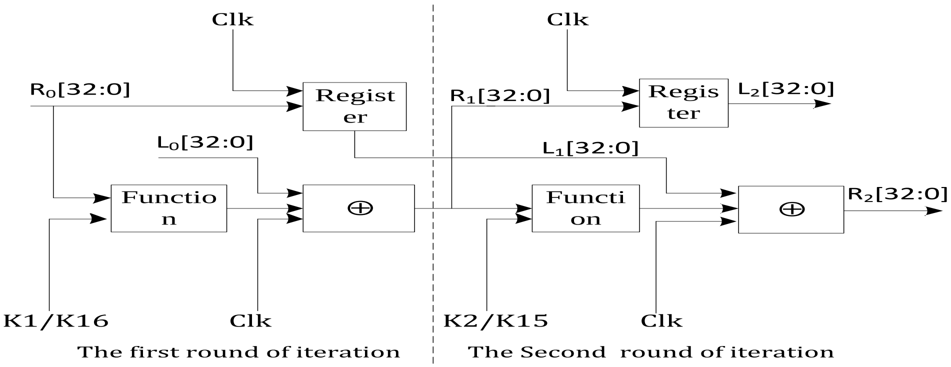


Figure 3. Iterative process

Substitution-box is the core of Function module; the design quality will affect the speed of operation. Usually, the design method of the S-box has the look-up table, ROM storage, the double case method and the expression method. According to the literature [8], the expression has the lowest delay and the fastest speed after synthesize. The look-up table has the lowest resource usage. Therefore, in order to improve the speed of operation, I select the expression method to design S-box.

**Sub-key Module.** The plaintext of DES algorithm is the 64 bit, we can get the 56 bits of valid data while get rid of the parity bit, the principle is shown in Fig. 4. The key can generate 16 sub keys through the PC-1, rotate left 16 times, PC-2. If we use the general way that we describe the whole process, which is not only a high occupancy rate of resources, but also relatively complicated to implement on FPGA. According to principle of the generation of sub key, the whole operation process is only related to the fixed displacement and rotate left, the relationship between each sub key and original key is fixed. Therefore, we can deduce the fixed relationship between sub key and original key [9]. Then the direct assignment method can be used to obtain all the sub keys once and for all. We can know that the circuit is mainly realize by the latch, which does not only avoid a lot of logic operations, but also improve the efficiency of the operation through the simulation environment.

In addition, in order to make the pipeline work stable, sub-key need to transported into function module under the control of the clock, so we inserted some registers into the circuit to store the generated keys.

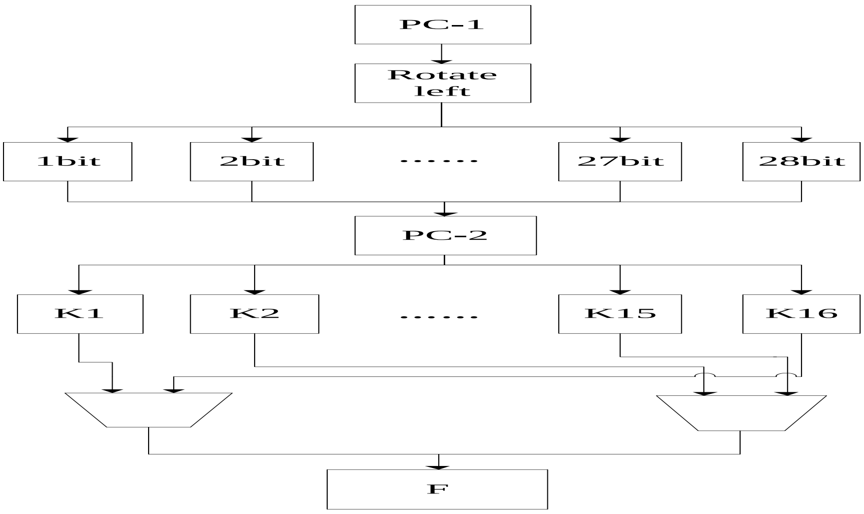
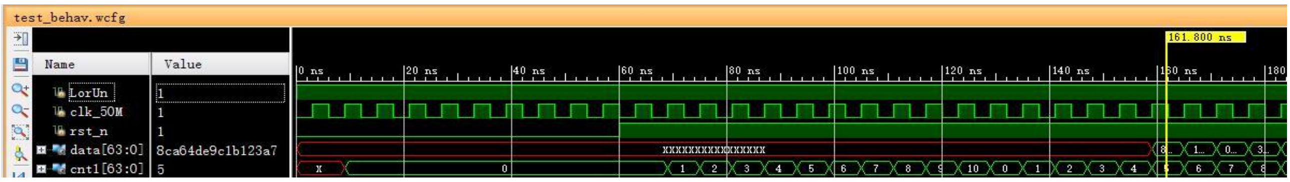


Figure 4. Sub key generation process

## The Implementation of Encryption System

At the completion of the preliminary circuit design, we need to synthesize and implement circuit, and download the bit stream file which the simulation software to generate to FPGA for validation and debugging [10]. We select Vivado 2015.2 as the development environment, and select DNPCIe_10G_K7_LL_QSFP as the electrocircuit detection platform. It has a four channels of PCIe, a 40Gbe QSFP+ and the 7K325 FPGA, which can meet the hardware requirements of the encryption system.

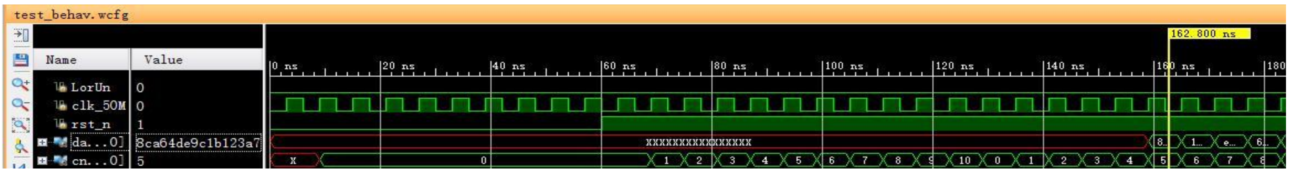
**Functional Verification.** When the data is encrypted, the data received by FPGA is processed in the memory, the data is transmitted to the DES module and begin to encrypt when the encrypted signal is set high. When LorUn=1, the circuit is in a state of encryption, the circuit is reset and begin to encrypt while rst_n=1. Data from zero to the circuit, and there is a plaintext read into the encryption module for computing after each clock. In order to simplify the verification circuit, the initial key and the plaintext are used in the same data resource. As shown in Fig. 5, circuit input plaintext cnt1=64`h0000000000000000, and output ciphertext data=64`h8ca64de9c1b123a7 after 16 cycles of operation, there is a ciphertext output after each cycle of clock, and simulation results are accurate.


test_behav.wcfg

| Name | Value |
|---|---|
| LorUn | 1 |
| clk_50M | 1 |
| rst_n | 1 |
| data[63:0] | 8ca64de9c1b123a7 |
| cnt1[63:0] | 5 |

Figure 5. Encrypted waveform

When LorUn=0, the circuit is in a state of decryption, and when rst_n=1, the circuit is reset and begins to decrypt. As shown in Fig. 6, circuit input ciphertext cnt1=64`h0000000000000000, and output plaintext data=64`h8ca64de9c1b123a7 after 16 cycles of operation, there is a plaintext output after each cycle of clock. The checking results is established to verify the mentioned method.


test_behav.wcfg

| Name | Value |
|---|---|
| LorUn | 0 |
| clk_50M | 0 |
| rst_n | 1 |
| da...0] | 8ca64de9c1b123a7 |
| cn...0] | 5 |

Figure 6. Decryption waveform

**Result Comparison.** According to the information proved on the display, software methods to process 128KB data, the time consumed by encryption and decryption is 0.015s and 0.0148s. But the design of the circuit in the process of the same data, the time of encryption and decryption are 0.00013s. In other words, the speed of encryption and decryption can reach 10.62Gb/s. Test results are shown in the table 1. We know that the pipeline architecture can effectively improve the speed of encryption and decryption.

Table 1 Comparison of test results

| Module<br>Method | Encryption | Decryption |
|---|---|---|
| **Software** | **66.67Mb/s** | **67.56Mb/s** |
| **Hardware** | **10.62GB/S** | **10.62Gb/s** |

## Conclusion

In this paper, a design method on 16 level pipelined encryption system is proposed for encrypted link. We can get the high speed encryption effect through the actual test. The encryption scheme of hardware can enhance the encryption speed of more than 100 times than software solutions, and release the CPU resource. Therefore, this scheme effectively solves the problem that encrypted procession take up much time.

## References

[1] Zhang Zhihong, Wu Qingbo, Shao Lisong, et al. Design and Implementation of TCP /IP Offload Engine Protocol Stack Based on FT Platform. Computer Technology and Development, 2014; 24(7): 1-4

[2] Wang Hepeng, Wang zhihong, Li Jianing, et al. New Processor for Data-Intensive Computing. Journal of Software, 2016:1-19

[3] Li Lang, Zou Yi, Guo Ying. Cryptography Engineering. BeiJing: Tsinghua University Press, 2014

[4] Zhang Handong, Zhu Minghui. The Implementation of Improved DES Algorithm Based on FPGA. Application of Electronic Technique, 2011; 37(4): 138-141

[5] Ge Yong, Li Hua, Ning Yongcheng. The Implementation of DES Algorithm Based on FPGA. Electronic Science and Technology, 2013; 26(7): 172-176

[6] Li Muguo, Huang Ying, Liu Yuzhi. Design of DMA Transmission with PCIe Bus Interface Based on FPGA. Computer Measurement & Control, 2013; 21(1):233-235

[7] H Kavianipour，S Muschter，C Bohm. High Performance FPGA-Based DMA Interface for PCIe. IEEE Transactions on Nuclear Science,2014， 61(2):745-749.

[8] Zhu Xinxin, Li Shuguo. High-Performance 3DES Algorithm Implement Based on FPGA. Microelectronics & Computer, 2015; 32(9): 54-59

[9] IOAN, A. D. New Techniques for Implementation of Hardware Algorithms inside FPGA Circuits. Advances in Electrical and Computer Engineering, 2010, Vol.10 (2), pp.16

[10] He Bing. FPGA Xilinx Authoritative Design Guide: Vivado 2014 Integrated Development Environment. Beijing: Publishing House of Electornics Industry, 2015.2