An FPGA Design of DES Algorithm

A. Kongmunvattana and P. Chongstitvatana

The department of computer engineering
Chulalongkorn University

ABSTRACT

This article presents a design of the Data Encryption Standard (DES) chip. The DES is discussed concisely. The result of the previous works are discussed, the encryption speed. Then the description of an implementation based on FPGA devices is presented including overall system structure, a review of design considerations, an estimation of gate array used in the design and performance evaluation.

1. INTRODUCTION

The Data Encryption Standard (DES) algorithm [7], which a 64-bit block of plaintext is enciphered into a 64-bit ciphertext by means of a 56-bit of cryptographic key, consists of 16 rounds of encipherment. The deciphering process is identical to encryption except the key is used in reverse order.

The central technique which governing DES is the design of DES's permutation (refer as P-Box), substitution (S-Box) and key schedules. The permutation schedule must ensure that each bit of ciphertext be a function of all plaintext and key bits after a minimum number of rounds. A total of 2^{64}! different transformations from plaintext to ciphertext is obtained from S-Box design. The S-Box function must be designed in such a way that cryptographic strength is enhanced and an easy implementation onto a single chip can be achieved.

2. MOTIVATION

There exists commercially available chips such as Intel 8294A [3] but there is a restriction on exporting crytography technology to any country outside USA. Therefore an implementation for local use is necessary. The applications can be in the transmission of important data through a public network. An equipment can be designed to be connected to a modem and encrypt/decrypt all the transmitted data.

3. PREVIOUS WORK

Broscius and Smith [2] took the advantage of parallelism by decomposing the DES algorthm into several parallel units to minimize delay. Its prototype has the encryption rate of 93 Mbps. Mansor and others [4] reported an implementation of DES in hardware but only 8 bits data and key. Muller-Schloer [6] also reported a hybrid system using both RSA and DES. An interesting aspect of the design is the DES part, it used the Western Digital WD2001 DES chip which has the encryption rate of 1.3 Mbps. MIMOS [5] advertised a chip for DES which has a data rate of 5 Mbps. There are a few work in Thailand in the area of DES. Attachoo and Chaisingthong [1] described the use of DES for millitary purpose. Chaisingthong has a DES design which has been verified by a simulation. He aimed to produce a system which has a large number of bit for data block and key. This work aims to study a feasible (e.g. small enough to be put in an fpga chip) implementation.

4. THE DESIGN

The logical design of the DES chip can be optimize to reduce the number of gates. One way is to partition (slice) the data path along vertical axis (bit slicing) and executes the algorithm bit serially. Another way is to partition the data path horizontally by reducing the stages into a loop. This requires a loop control and changing of parameters between different iterations (Figure 1). After analyzing both design alternatives the latter was chosen as it used fewer number of gates than the former. The function of this DES chip can be separated into 3 stages: key schedule, encipherment and decipherment.

4.1 *Key Schedule*

The key schedule begins with an initial permutation which selects 56-bit out of 64-bit external key $(PK_1(Y))$, by stripping off the 8 parity bits (One bit in each 8-bit byte of key may be used for error detection in key generation, distribution and storage). These 56-bit key is loaded into two 28-bit shift registers (C_i, D_i) and then is shifted one or two positions to the left or right according to the key schedule for encipherment or decipherment $(PK_2(Y'))$. Thus, 48 of the 56 key bits will be selected, tranposed and used at each round of encipherment, refer as K_i $(i=1..16)$.

4.2 *Encipherment*

The 64-bit block of plaintext is subjected to an initial permutation $(IP(X))$ which is used to construct two sets of 32-bit vectors (L_1, R_1). Where R_1 is a right half of plaintext at round 1. After that, R_i $(i = 1..16)$ will be passed through a function which produces one of 48-bit block as an output $(E(R_i))$. Once R_i' is generated, it is added bit-by-bit (exclusive or) to K_i.

This 48-bit output is used as an input of S-Box. It is passed through a nonlinear transformation, to form a 32-bit output. The transposed 32-bit output is added bit-by-bit to L_i and the result represents the righ half of the next round R_{i+1} while the left half of the next round L_{i+1} duplicate from the right half of the previous round, R_i. After 16 rounds of interation, R_{16} and L_{16} are obtained. Both the left and right side are combined and transposed, inverse the initial permutation. The final output is a 64-bit block of ciphertext.

4.3 *Decipherment*

For the implementation of decipherment, only the key schedule has to be rearranged . K_i in the decipherment is in the reverse order of K_i in the encipherment. All other part of the design remain the same.

5. EXPERIMENT

Due to the constraint of the technology in which this experiment was carried out. The chip has to be operate in 16-bit block (instead of original 64-bit block) to enable it to fit in one chip of Xilinx 4003PC84-5. The design is general and can be implemented to operate in 64-bit block manner. The reduction in width of block makes it not very secure compare with the original encryption system but should be suffice to verify the design. Even so, the attack on the 16-bit system is not easy.

The chip uses 1946 logic gates, 172 D-flipflops and it has a minimun clock period of 33.5 nanosecond. The pinout out the chip is shown in Figure 2. It uses a common clock for both the encryption and the decryption units which allows them to be operated synchronously.

The testing is done by streaming data from a computer into encryption unit and connects the output from the encryption unit to the input of the decryption unit. The stream output of the decryption unit is then saved. It is then compared with the original input data. The clock rate used in the experiment is 8 MHz. The chip works correctly.

6. CONCLUSION

The realization of a DES chip in this work is shown to be practical. The chip can be implemented by a standard FPGA device. The experiment showed the tradeoff that is made in order to test the design. Although the experimental chip has only 16 bits data width, the design is implementable for 64 bits data. The chip has been tested to work correctly at the data rate of 8 Mbps. It has a maximum data rate of 29.9 Mbps.(estimate from the design verification tool). As a prototype, the chip can be used as a data encryption/decryption device connected to a serial port of a computer.

7. REFERENCES

[1] B. Attachoo and C. Chaisingthong,. “Data encryption algorithm for military purpose,” in *Journal of The Computer Association of Thailand*, vol. 20 no.106, pp. 39-50, 1994. (in Thai)

[2] A. G. Broscius and J. M. Smith, “Exploiting Parallelism in Hardware Implementation of DES,” *Distributed System Lab*, CIS dept., Univ. of Penn., 1992.

[3] Intel Corp., “8294A Data encyption / decryption unit data sheet,” in *Microcommunication handbook*, pp. 10/45 - 10/56., 1986.

[4] I. Mansor et al., “IC Design of speech scrambling system using OrCAD,” in *Proc. of the 2nd Regional Seminar ,* Singapore*,* vol. 7, pp. M38-M41, 1993.

[5] MIMOS, “MIMOS Secret chip,” *Malaysian Institute of Microelectronic Systems*, Kuala Lumpur, Malaysia, 1994.

[6] C. Muller-Schloer, “A Microprocessor-based crytptoprocessor,” in *IEEE Micro*, vol. 3 no. 5, pp. 5-15, October 1983.

[7] M. Y. Rhee, “Data Encryption Standard as block cipher algorithm,” in *Cryptography and Secure Communications*, McGraw-Hill, pp.47-102, 1994.

8. ACKNOWLEDGEMENT

NECTEC has generously funded the FPGA design tools and the Xilinx development system without which this work will not be possible. We would like to thank S. Prasitjutrakul for his help in setting up the design system and for his expert advice about FPGA.

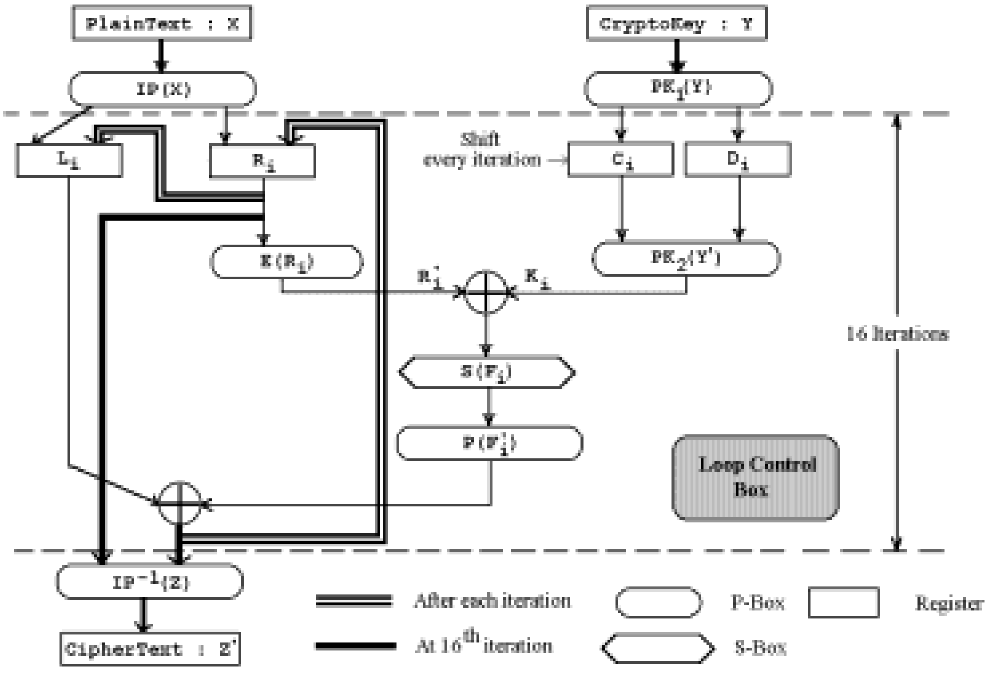

Figure 1. The design of the DES chip.

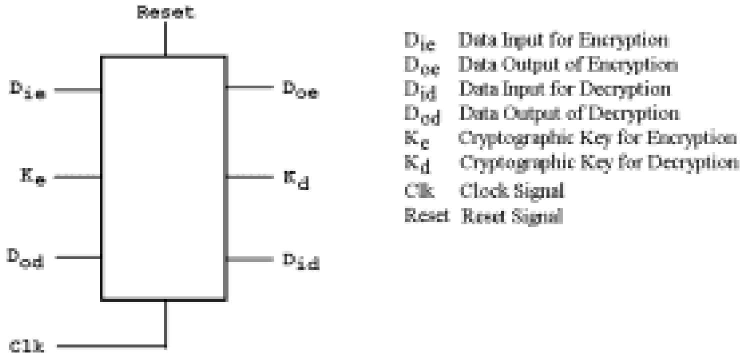
Reset
D_{ie}
K_e
D_{od}
Clk
D_{oe}
K_d
D_{id}
D_{ie} Data Input for Encryption
D_{oe} Data Output of Encryption
D_{id} Data Input for Decryption
D_{od} Data Output of Decryption
K_e Cryptographic Key for Encryption
K_d Cryptographic Key for Decryption
Clk Clock Signal
Reset Reset Signal

Figure 2. The pinout of the DES chip