International Journal of Computer Applications Technology and Research
Volume 5–Issue 10, 654-656, 2016, ISSN:-2319–8656

Pipelining Concept for Low Power DES Implementation

Ansiya Eshack
School of Technology & Applied Sciences, M. G. University Regional Centre, Edapally, Ernakulam, India

S. Krishnakumar
School of Technology & Applied Sciences, M. G. University Regional Centre, Edapally, Ernakulam, India

Abstract: An implementation of Data Encryption Standard (DES), one of the most widely accepted cryptographic standards, using the pipelining concept is described in the paper. Pipelining is an approach used to reduce power consumption in systems. The simulation of the design employs the Xilinx software and results show that the number of slices used in the device has reduced compared to previous available works. This reduction in the slices ensures lower use of power by the system. The throughput of the system has also increased due to pipelining.

Keywords: Cryptography, DES, Fiestal Structure, FPGA, Low-power, Pipelining, Verilog

1. INTRODUCTION

Cryptography is a tool used to maintain the confidentiality of information. It protects a piece of data, sent from one user to another, by converting it into a form which is not readable by an eavesdropper. Data Encryption Standard (DES) is a cryptographic standard symmetric algorithm which is based on the Feistel structure and was published as an official Federal Information Processing Standard (FIPS) for the United States in 1977 [1].

The DES Algorithm is the most commonly used block cipher, meaning it has a structure where the same blocks are repeated a number of times [2]. It works through a series of 16 rounds and each round consists of the same operations, ie, bit-shuffling, non-linear substitutions & exclusive-OR operations. This structure of DES Algorithm makes it suitable for pipelining and the 16 rounds of the algorithm are unrolled and pipelined using additional registers [3], [4].

Pipelining is one of the methods available to lower the power consumption of a system. It also increases the speed of the system by simultaneously processing two or more blocks of data, ie, two or more pairs of data are worked on at the same time. In a non-pipelined system, however, only one block of data is processed at a time. This greatly increases the time taken to produce the output.

In the DES system, though the inputs are applied every clock cycle, the outputs come only after each 16 clock cycles, ie, in the 17th cycle, 33rd cycle, 49th cycle and so on [2]. When pipelining is applied to this system and the inputs are fed continuously during each clock cycle, the outputs also come out continuously from the 17th clock cycle onwards. There is an initial delay of 16 clock cycles, but then the outputs are generated every cycle.

In this paper, a 16-stage pipelined DES structure in Electronic Code Book (ECB) mode is proposed. The ECB mode is used here as it supports pipelining [5]. The work is written in Verilog Hardware Description Language and implemented on Xilinx Spartan-3e FPGA. The proposed method uses lower number of FPGA slices as compared to earlier works. This reduction in the number of slices has led to reduced power consumption. As the outputs come every clock cycle, the throughput of the system also increases as compared to a non-pipelined system.

The rest of the paper is as follows: Section 2 discusses the basics of DES algorithm. Section 3 explains the pipelined DES algorithm. Summary of the implementation and the details of FPGA resources used are presented in Section 4. A performance comparison is done with similar works in this field and this is given in Section 5. Conclusion and future scope of the work is discussed in Section 6.

2. DES ALGORITHM

DES is also called Data Encryption Algorithm and is one of the basic cryptographic algorithms in use since the 1970s. The basic concept of DES algorithm is as shown in Figure 1. The input is a text message of 64-bits. This is converted to a cipher (encrypted) text of 64-bits with the help of a 56-bit key. The key is actually 64-bit to begin with; every eight bit of the key is discarded prior to encryption to make it 56-bits. The encrypted message is converted back (decrypted) to the original text using the same 56-bit key. The algorithm for DES Encryption and Decryption is given below.

Algorithm

1 An Initial Permutation (IP) is performed where all the 64-bits of the Input message or Cipher message are rearranged or permuted.

2 The permuted block is then divided into two halves, the left half consisting of 32 bits and the right half consisting of 32 bits.

3 Both the left half and the right half undergo 16 rounds of encryption with different sub-keys generated during each round. The order of sub-keys used during the rounds in DES Decryption will be the reverse order of those used in DES Encryption.

4 The two halves are joined together and a Final Permutation (FP) is performed on this 64-bit text.

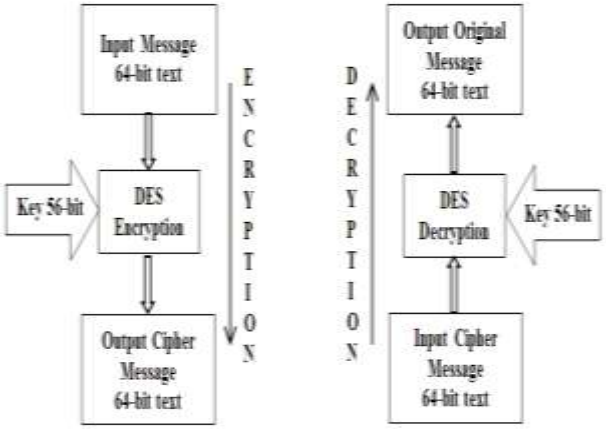

Figure 1. Basic concept of Encryption and Decryption using DES algorithm

In DES Encryption, the Input message is encrypted to produce an Output Cipher message. In DES Decryption, the Cipher message is decrypted to produce the Original message. The Figures 2 and 3 sketches the different steps involved in the DES Encryption and Decryption.

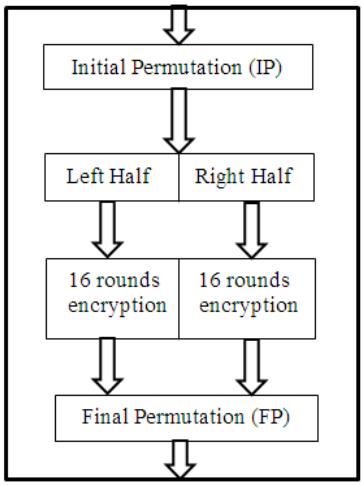

Figure 2. DES Encryption

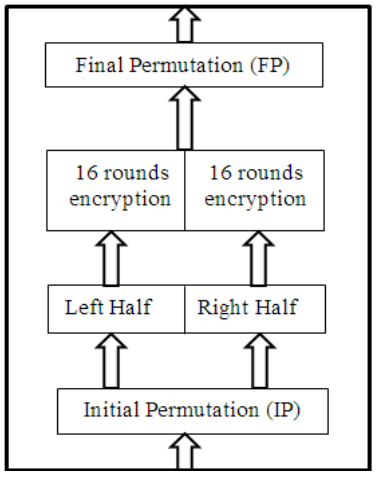

Figure 3. DES Decryption

3. PIPELINED DES ALGORITHM AND IMPLEMENTATION

DES Algorithm follows an iterative Fiestal structure developed by Horst Fiestal. This structure allows for efficient pipelining in 4, 6, 8 or 16 stages [5]. The proposed design performs 16 stage pipelining in the DES algorithm. This is done using additional registers in each of the 16 rounds of the algorithm.

The device XC3S500E from the Spartan-E family [6] is used for the implementation of both non-pipelined and 16-stage pipelined DES system. The implementation results are explained in this section.

As discussed above, in a non-pipelined DES system for an input given in a clock cycle, the output occurs only after a delay of 16 clock cycles. Even though inputs are given every clock cycle, there will be a delay of 16 clock cycles between the respective inputs and outputs. The simulation result of such a system is shown in Figure 4.

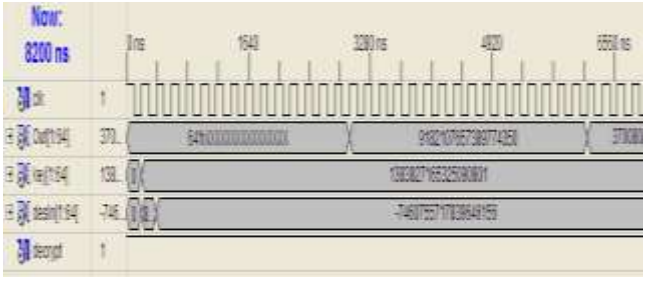

Figure 4. Non-pipelined DES System

In the 16-stage pipelined DES system, when the input data is given every clock cycle, it can be seen that though the first output comes only after 16 clock cycles, ie, in the 17th clock cycle, the subsequent outputs come from the 18th clock cycle onwards. Thus by pipelining, the delay of 16 clock cycles in between the respective inputs and outputs is avoided. This greatly increases the throughput of the system. The number of registers and other logic components used also doesn't increase substantially with pipelining. Fig. 5 shows the output of a 16-stage pipelined DES system.

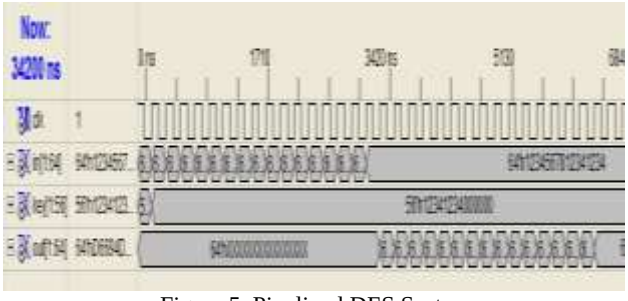

Figure 5. Pipelined DES System

The device utilization summary for the device XC3S500E is shown in the Table 1. The design uses only 1904 (20%) slice flip flops, 3344 (35%) look-up tables and 186 (80%) input-output blocks. The number of slices used is 2167 which is only 46% of the total available slices in the device.

Table 1. Device Utilization Summary

Utilization Summary for the Device XC3S500E – 5FG320			
Logic Utilization	**Used**	**Available**	**Utilization**
No. of Slices	2167	4656	46%
No. of Slice Flip Flops	1904	9312	20%
No. of 4 input LUTs	3344	9312	35%
No. of bonded IOBs	186	232	80%
No. of GCLKs	1	24	4%

4. PERFORMANCE COMPARISON

Table 2 shows a performance comparison of different pipelined DES implementations.

The system designed by M. McLoone and J. McCanny [4] used 6446 slices for the operation. Further, the works designed by Praveen K., et.al. [6] used 2881 CLB slices, while that of V. Patel [5] used 2814 slices. Two other similar works include that of Raed Bani-Hani [7] which uses 2584 slices and that of Abd El-Latif [8] that used 2566 slices respectively. However it can be seen that the proposed implementation uses only 2167 CLB slices (46%) of the entire slices available in the device. Thus there is a reduction in the number of slices used compared to the earlier works. This reduction also leads to decrease in the power utilization of the device.

Table 2. Performance Comparison

Sl. No	Author	Device Used	CLB Slices	Max. Freq. (MHz)
1	M. McLoone and J. McCanny [4]	XCV1000	6446	59.5
2	Praveen K., Prabaharan Poornachandran, et. al.[6]	XC5VLX110T	2881	--
3	V. Patel, R. C. Joshi, A. K. Saxena [5]	XC3S500E	2814	111.882
4	Raed Bani-Hani, Salah Harb, et. al.[7]	XCV300	2584	102
5	K. M. A. Abd El-Latif, H. F. A. Hamed, et. al. [8]	XC3S500E	2566	113.75
6	Proposed Method	XC3S500E	2167	175.77

5. CONCLUSIONS

Here a hardware design for DES algorithm is proposed and implemented on the device XC3S500E (Spartan-3E) from the Xilinx FPGA family. The design has implemented the pipelining concept into the DES encryption and decryption.

The result of the hardware usage of the work is compared with that of previous works. It is seen that there is a reduction in the number of CLB Slices used by the proposed design. This reduction in the number of slices also leads to reduced power consumption by the system. Further, use of pipelining increases the performance of the system. The maximum allowed frequency of the proposed method is 175.77MHz.

6. 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] Cryptographic Algorithms on Reconfigurable Hardware, Chapter 8, Springer Science and Business Media, 2006.

[3] S. Trimberger, R. Pang, and A. Singh, A 12 Gbps DES Encryptor/Decryptor Core in an FPGA, Proc. Cryptographic Hardware and Embedded Systems (CHES '00), pp. 156-163, 2000.

[4] McLoone, McCanny, High-performance FPGA implementation of DES using a novel method for implementing the key schedule, IEE proceedings on Circuits, Devices and Systems, Vol 150, pp. 373-378, 2003.

[5] V. Patel, R. C. Joshi and A. K. Saxena, FPGA implementation of DES using pipelining concept with skew core key-scheduling, Journal of Theoretical and Applied Information Technology, vol. 5, no. 3, pp. 295-300, March 2009.

[6] Praveen K, Prabaharan Poornachandran, et. al., Implementation of DES using Pipelining concept with Skew Core Key Scheduling in Secure Transmission of Images, ACM Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, 2012

[7] Bani-Hani, R., et al., High-Throughput and Area-Efficient FPGA Implementations of Data Encryption Standard (DES). Circuits and Systems, 5, 45-56, 2014.

[8] Hardware Implementation of DES Using Pipelining Concept with Time-Variable Key, Karim Moussa Ali Abd El-Latif, IEEE Intl. Conf. on Microelectronics (ICM 2010)

[9] Spartan-3E FPGA Family Data Sheet, DS312 July 19, 2013