International Journal of Computer and Communication Technology

Volume 6 | Issue 2 Article 9

April 2015

IMPLEMENTATION OF PIPELINED DES USING VERILOG

P. SANTHAMMA
Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala., psanthamma@gmail.com

B. RAGHAVAIAH
Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala., braghavaiah@gmail.com

N. SURESH BABU
Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala., nsureshbabu@gmail.com

Recommended Citation
SANTHAMMA, P.; RAGHAVAIAH, B.; and BABU, N. SURESH (2015) "IMPLEMENTATION OF PIPELINED DES USING VERILOG," *International Journal of Computer and Communication Technology*: Vol. 6 : Iss. 2 , Article 9.
DOI: 10.47893/IJCCT.2015.1286
Available at: https://www.interscience.in/ijcct/vol6/iss2/9

IMPLEMENTATION OF PIPELINED DES USING VERILOG

P.SANTHAMMA, B.RAGHAVAIAH & N.SURESH BABU

Department of Electronics and Communication Engineering,
Chirala Engineering College, Chirala.

Abstract: An implementation of the Data Encryption Standard (DES) algorithm is described. Protection of data during transmission or while in storage may be necessary to maintain the confidentiality and integrity of the information represented by the data. The algorithms uniquely define the mathematical steps required to transform data into a cryptographic cipher and also to transform the cipher back to the original form. The proposed pipeline method is for improving the speed of execution when compare with the non-pipeline method.

Keywords: *Data Encryption Standard, Field Programmable Gate Arrays, Finite State Machine.*

1. INTRODUCTION

The Data Encryption Standard (DES) specifies two FIPS approved cryptographic algorithms as required by FIPS 140-1. When used in conjunction with American National Standards Institute (ANSI) X9.52 standard, this publication provides a complete description of the mathematical algorithms for encrypting (enciphering) and decrypting (deciphering) binary coded information. Encrypting data converts it to an unintelligible form called cipher. Decrypting cipher converts the data back to its original form called plaintext. The algorithm described in this standard specifies both enciphering and deciphering operations which are based on a binary number called a key.

A DES key consists of 64 binary digits ("0"s or "1"s) of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits, which are not used by the algorithm, may be used for error detection. The 8 error detecting bits are set to make the parity of each 8-bit byte of the key odd, i.e., there is an odd number of "1"s in each 8-bit byte.

A TDEA key consists of three DES keys, which is also referred to as a key bundle. Authorized users of encrypted computer data must have the key that was used to encipher the data in order to decrypt it. The encryption algorithms specified in this standard are commonly known among those using the standard cryptographic.

Sometimes keys are generated in an encrypted form. A random 64-bit number is generated and defined to be the cipher formed by the encryption of a key using a key encrypting key. In this case the parity bits of the encrypted key cannot be set until after the key is decrypted. Security of the data depends on the security provided for the key used to encipher and decipher the data.

Data can be recovered from cipher only by using exactly the same key used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. However, it may be feasible to determine the key by a brute force “exhaustion attack.” Also, anyone who does have the key and the algorithm can easily decipher the cipher and obtain the original data.

A standard algorithm based on a secure key thus provides a basis for exchanging encrypted computer data by issuing the key used to encipher it to those authorized to have the data.

Cryptography is probably the most important aspect of communication security becoming increasingly important as a basic building block for computer security.

Cryptographic systems are characterized along three independent dimensions:

1. **The type of operations used for transforming plaintext to cipher text:** All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext is mapped into another element, and transposition, in which element in the plaintext are rearranged. The fundamental requirement is that no information be lost. Most systems referred to as *product systems*, involve multiple stages substitutions and transpositions.
2. **The number of keys used:** If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.
3. **The way in which the plaintext is processed:** A *block cipher* processes the input one block of elements at a time, producing an output block for each input block. A *stream cipher* processes the input elements continuously, producing output one element at a time, as it goes along.

Before beginning, we define some terms. A symmetric encryption scheme has five ingredients:

- **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption Algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a value

independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

- **Cipher text:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different cipher text. The cipher text is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm:** This is essential the encryption algorithm run in reverse. It takes the cipher text and the secret key and produces the original plaintext.

Cryptographic algorithms such as the Data Encryption Standard (DES) are frequently implemented in Field- Programmable Gate Arrays [4]. Orders of magnitude speedup over software implementations are due to the following attributes of DES:

- It is a pure datapath.
- The primitive operations are bit-level substitutions and permutations, which are inefficient in software. Fixed permutations are essentially free in hardware
- DES has 16 rounds, which can be unrolled and pipelined in hardware.

2. DES ALGORITHM

DES algorithm uses complicated logical functions such as various types of permutations, XOR and SHIFT functions. Since the key employed is transformed to mentioned function, by following the algorithm provided, the only way to decrypt the plaintext is to apply the same key in decryption algorithm as well. DES takes 64 bits plaintext and 56 bits key as input and generates 64 bits cipher data as output [2]. The block diagram of algorithm is shown in Fig. 1. Sometimes the key is considered as 64 bits where 8 bits is used for parity check. The DES structure is first described by Horst Feistel in 1973 [2], as shown in Fig. 2.

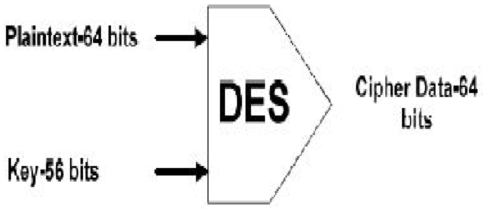


Fig.1. DES Block View

In this method, after initial permutation (IP) of the plaintext, it is divided into two halves L(0), R(0). The two halves pass through 16 rounds. Then after the final permutation (FP), the cipher data is produced. IP and FP work exactly in opposite ways to each other [2].

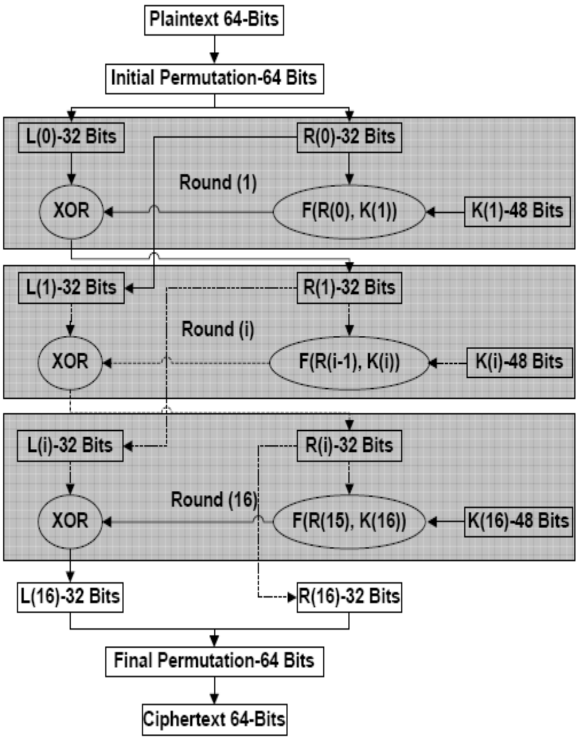


Fig.2. DES Algorithm

Each Round i has three inputs: L (i-1), R (i-1) and K (i) where K (i) is generated from Key Scheduler program which is described in the following section. All rounds have the same structure. In Round i, L (i) is equal to R (i-1). But R (i) is derived from [L (i-1) XOR F(R (i-1), K (i))a]. Where XOR is an exclusive-OR operator and F is a round function. The function diagram of F is shown in Fig. 3.

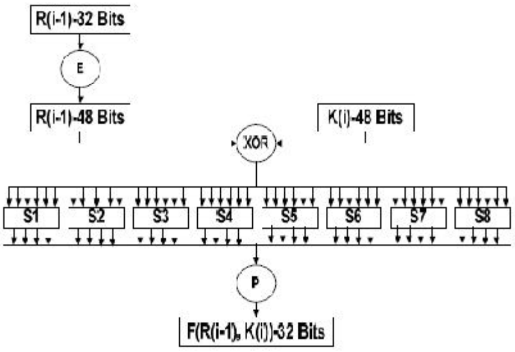


Fig.3. F(R, K)

The F(R, K) or round function is the core of the DES. It substitutes the right halve of data and generate 48 bits of data. After taking XOR with corresponding key K (i), it passes the data through S-box functions where each S-box function has its own look up table. After that a permutation of output of Sbox function is provided and, the result of F(R, K) is generated.

The key scheduler program generates a sub key K (i) and form the 56 bits key by taking different permutation and rotate the function to left (in encryption mode) or right (in decryption mode) on the original key. That means, for each round i, a different sub key, K (i), is generated. The key scheduler diagram is shown in Fig. 4.

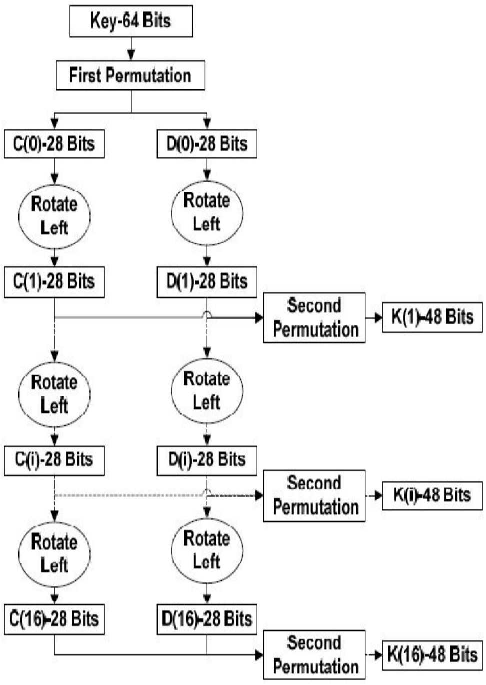


Fig.4. Key Scheduler

The decryption algorithm is similar to encryption algorithm. The only difference is the sequence of generated sub key. It assigns the K (6) to round 1 and K (5) to round 2 and so on [2]. An alternative to that would be to use rotate right instead of rotate left in key scheduling part.

3. NON-PIPELINED IMPLEMENTATION

The non-pipelined mode design is divided into three main function blocks. The block diagram of the design is shown in Fig. 5.

In traditional implementation, counters are used in order to control the round sequence. On the other hand, in prevalent designs, Finite State Machines (FSM) [4], [8] are used to control the round sequence.

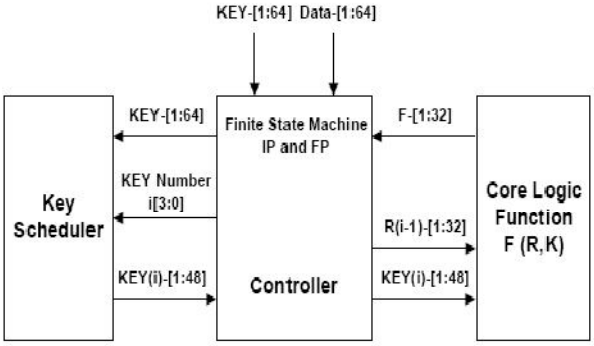


Fig.5. Block Diagram of Design

In non-pipelined mode implementation, FSM is used to control the round sequence and handshaking between different function blocks. It generates the appropriate data for key scheduler and F(R, K) function blocks, and it receives the produced data from them.

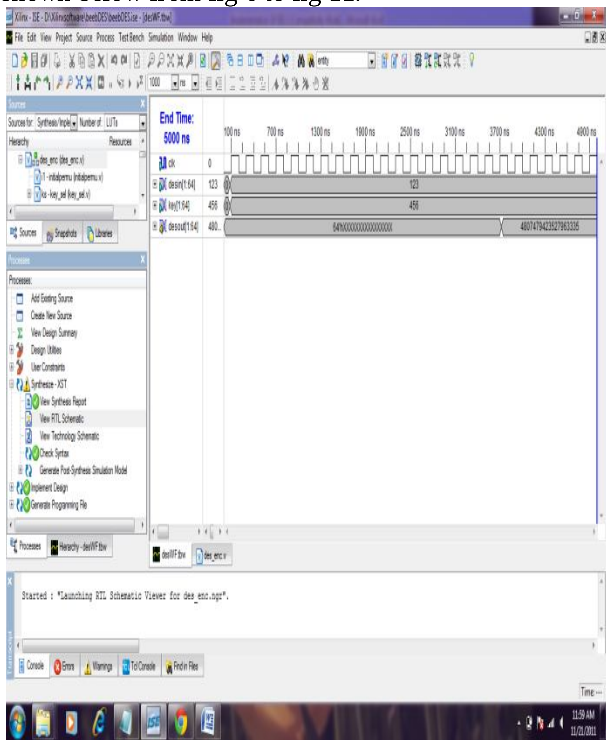
Xilinx XST uses processes [1] to describe FSM in VHDL language. Most of the FPGA synthesis tools use various templates for implementing the FSM. XST has 8 different techniques to describe FSM:

- Auto-State Encoding
- One-Hot State Encoding
- Gray State Encoding
- Compact State Encoding
- Johnson State Encoding
- Sequential State Encoding
- Speed1 State Encoding
- User State Encoding

Auto-State Encoding is used in this study, which tries to select the best suited algorithm for a FSM. FSM modules are implemented in slice logic (LUT) by default. However it can also be implemented into the block RAM. For larger FSM, using block RAM makes FSM faster and leaves the slice logics to its targeted design which causes better utilization in device slice usage. Only Virtex FPGA family uses block RAM feature [7].

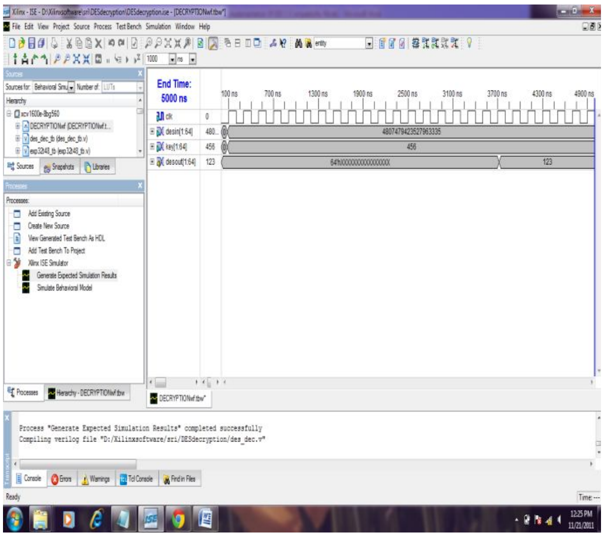
4. RESULTS

The timing diagrams for encryption and decryption and the RTL Schematics for encryption and decryption for a pipelined and non pipelined are also shown below from fig 6 to fig 11.



```
Started : "Launching RTL Schematic Viewer for des_enc.ngr".
```

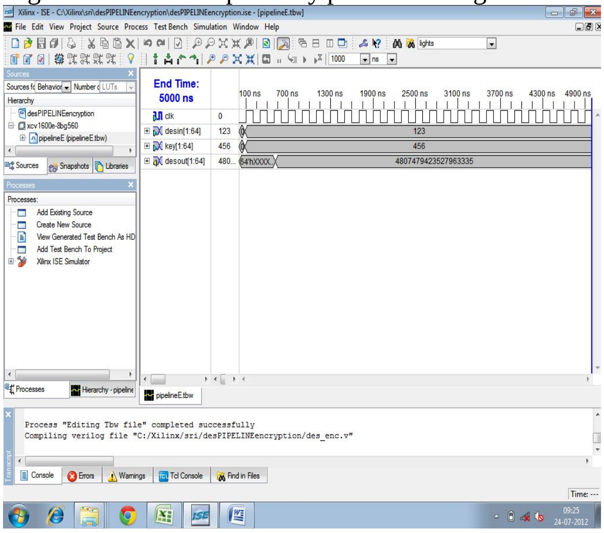
Fig 6 Non Pipelined Encryption Waveform



```
Process "Generate Expected Simulation Results" completed successfully
Compiling verilog file "D:/Xilinxsoftware/sri/DESdecryption/des_dec.v"
```

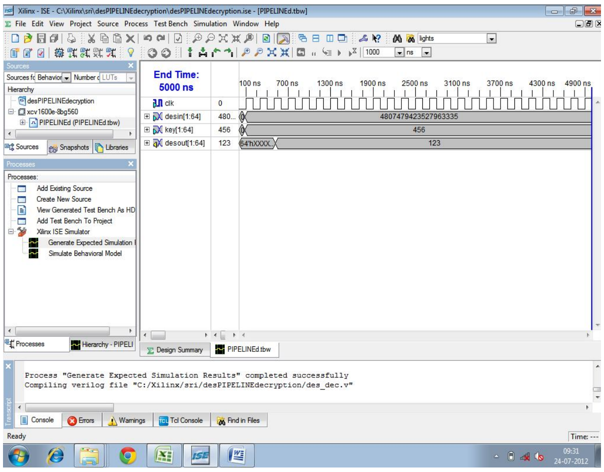
Fig 7 Non Pipelined Decryption Waveform

The translate process merges all of the input netlists and design constraints and generates Xilinx Native Generic Database file (NGC) which describes logical design reduced to Xilinx primitives. The MAP process fits the design to available resources in the targeted device and optionally place the design.



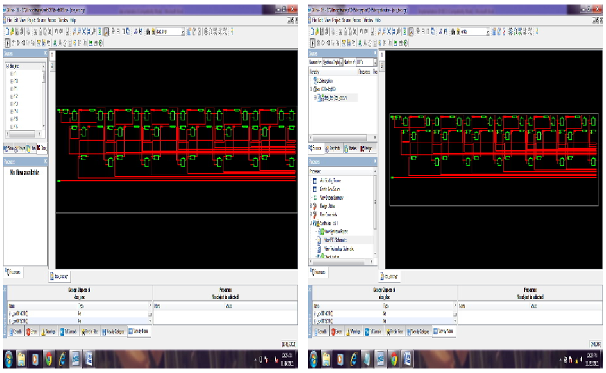
```
Process "Editing Tbw file" completed successfully
Compiling verilog file "C:/Xilinx/sri/desPIPELINEencryption/des_enc.v"
```

Fig 8 Pipelined Encryption Waveform



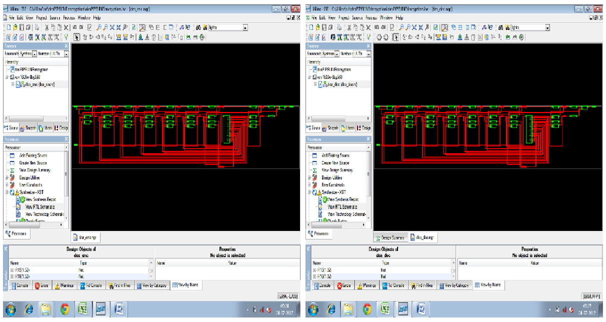
```
Process "Generate Expected Simulation Results" completed successfully
Compiling verilog file "C:/Xilinx/sri/desPIPELINEdecryption/des_dec.v"
```

Fig 9 Pipelined Decryption Waveform



Encryption Decryption

Fig.10 RTL Schematic for Non Pipelined



Encryption Decryption

Fig.11 RTL Schematic for Pipelined

5. CONCLUSION

In this project, we have given 64 bits input and 64 bits security key and observed how it is delivered at the the output with security for both pipeline and for nonpipelined . In this project there is no revealing of the original message to the hackers. The original message can be revealed to only sender and the receiver. The testing of the implemented design shows that it is possible to generate data in 16 clock cycles when non-pipelined approach is employed. When pipelined approach is employed on the other hand, 2 clock signals are required for data generation cycle. So improvement in speed is achieved.

Modern applications of DES cover a wide variety of applications, such as secure internet (ssl), electronic financial transactions, remote access servers, cable modems, secure video surveillance and encrypted data storage. The future scope of our project is to extend 64 bits inputs to n bits(n is any integer value). So, in future, any propriety information can be transmitted securely by extending this design to any number of blocks.(military or banking purposes).

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

REFERENCES:

[1] Fu Li, Pan Ming, “A simplified FPGA implementation based on an Improved DES algorithm,” IEEE Genetic and Evolutionary Computing, 2009. WGEC '09. 3rd International Conference on, pp.227-230.

[2] Ke Wang, “An encrypt and decrypt algorithm implementation on FPGA’s,” IEEE Semantics, Knowledge and Grid, 2009. SKG 2009. Fifth International Conference on, pp.298-301.

[3] L. Floyd, “Digital Fundamental with VHDL,” pp.362-368, ISBN: 0-13-099527-4, Pearson Education, 2003

[4] Mentor Graphics, “Modelsim Data sheet,”2008,Available:http://modelsim.s3.amazonaws.com/modelsim-sedatasheet.pdf .

[5] N. A. Saqib, F. Rodrıguez-Henriquez, and A. Dıaz-Perez, “A compact and efficient fpga implementation of the DES algorithm,” 2004.

[6] Pong P. Chu, “RTL Hardware Design Using VHDL,” pp.293-348, ISBN: 0-471-72092-5, Wiley-Interscience, 2006.

[7] Teo Pock Chueng, Yusoff, Z.M., Sha’ameri, A.Z., “Implementation of Pipelined Data Encryption Standard (DES) Using Ultera CPLD,” IEEE TENCON 2000. Proceedings, publication year 2000, Page 17-21 vol.3.

AUTHORS PROFILE:

P.Santhamma is pursuing M.Tech in VLSI &ES at Chirala Engineering College,Chirala.

B.Raghavaiah,M.Tech in Embedded systems from JNTUK& he is working as HOD of EIE Dept. in CEC,Chirala. He has 6 years of teaching Experience.

Prof.N.Suresh Babu is Vice- Principal & HOD of ECE Dept in CEC,Chirala.He got his M.Tech in Microwave Engineering from Birla Innstitute of technology, Ranchi. He has 13 years of teaching Experience and 2 Years of Industrial Experience in various organisations .

◈◈◈