International Journal of Computer Applications and Technology
journal homepage: www.ijcat.com

Implementation of Pipelined Data Encryption Standard for Security Enhancement through Verilog

U.Ratna Kumari
Department of ECE,
GIT, GITAM University,
Vishakhapatnam, Andhra Pradesh, India

T.K.Rasagna
Department of ECE,
GIT, GITAM University,
Vishakhapatnam, Andhra Pradesh, India

Abstract: This paper specifies a cryptographic algorithm in order to protect sensitive data. To maintain the data confidentially and to protect it, we need to convert the data into different form that differs completely from input and then transmit it. That data has to be again decrypted at the receiver. The algorithm defines the steps needed to encrypt the data and also to decrypt it. The pipelined DES has three modules: DES, pipeline, Control module. This design is programmed in Verilog. By pipelining we can achieve high throughput and by implementing triple DES, the security can be increased.

Key words: DES, Key scheduling, F-function, pipelined DES, triple DES.

1. INTRODUCTION

Most of the information in today's world is in digital format. For example most of the information in the form of photos, music and private information can be transmitted through copper, optical or wireless network to a recipient anywhere in the world. One of the advantages of Internet is the open system architecture. Its flexibility makes Internet developed fast. On the other hand, the lack of privacy in the Internet becomes obstacle to growing further. However, this weakness can be eliminated with the introduction of cryptography. Cryptography is used to transform intelligible information to unintelligible data. The Data Encryption Standard (DES), known as the Data Encryption Algorithm (DEA) by ANSI and the DEA-1 by the ISO, has been a worldwide standard for over 20 years. DES is a block cipher, which takes 64-bit input and 64-bit key. A 64-bit output is produced. The effective key length is 56 bits because 8 bits are used as parity-checking bits. There are a total of 2^56 possible keys available in 56-bit key length. DES is a symmetric algorithm. The same key is used for both encryption and decryption. DES has 16 rounds, meaning the main algorithm is repeated 16 times to produce the cipher text.

2. ALGORITHM

In each basic building block of DES, the input will be split into two, left half and right half. The right half will become left half for the next round. Meanwhile, the right half will go through the function f to produce a key-dependent output and then XOR with left half. The result will become right half for the next round. The basic building block of DES is repeated for 16 times [1]. The only difference between each round of building block is the key used as shown in figure 1. Every eight bit of the 64-bits key is used for parity checking and otherwise ignored. After an initial permutation, the 64-bits input is split into a right and left half each 32 bits in length. DES has 16 iterations or rounds. In each round a function *f* is performed in which the data is combined with a 48-bits permutation of the key. After the 16th iteration, the right and left halves are concatenated and a final permutation, which is the inverse of the initial permutation, completes the algorithm [3].

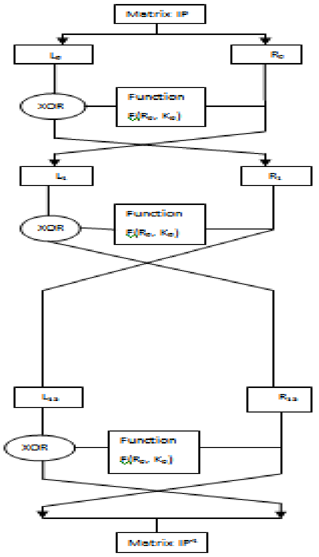


Figure 1. DES algorithm

3. F-FUNCTION

The function *f* of the DES algorithm is made up of four operations. Firstly, the 32-bits right half of the plaintext is expanded to 48-bits and then X-ORed with a 48-bits sub-key K1. The result is fed into eight substitution boxes (s-boxes), which transform the 48-bits input to a 32-bits output [4].Finally, a straight permutation (P-permutation) is performed, the output of which is XORed with the initial left half L, to obtain the new right half R1. The original right half R0 becomes the new left half L1 as shown in Fig 3.

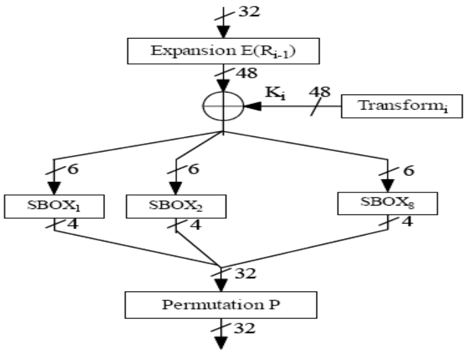


Figure 2. f-function

4. KEY SCHEDULING

Although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. The least significant (right-most) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. The initial step in the in this procedure is to remove the parity check bits in the 64-bit key. Every eighth bit is used for parity checking, leaving 56-bits. A different 48-bits sub key is now generated for each of the 16 rounds of DES [7][8]. The sub-keys are determined by first splitting the 56-bits into two 28- bits lengths of data. Then both halves are shifted left by either one or two bits depending on the round number.The procedure for generating the sub keys - known as key scheduling is fairly simple:

1. Set the round number R to 1.
2. Split the current 56-bit key, K, up into two 28-bit blocks, L (the left-hand half) and R (the right-hand half).
3. Rotate L left by the number of bits specified in the table below, and rotate R left by the same number of bits.
4. Join L and R together to get the new K.
5. Apply Permuted Choice 2 (PC-2) to K to get the final K[R], where R is the round number we are on.
6. Increment R by 1 and repeat the procedure until we have all 16 sub-keys K [1] - K [16].

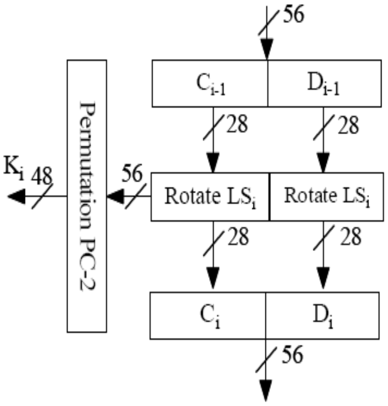


Figure 3. Key Scheduling

5. PIPELINED DES

The figure4 shows the pipelined DES architecture. There are four registers, each 16-bit, to store 64-bit input. Pipelined DES will load 4 different inputs from ISA bus for the first four rounds. This will fill the pipeline with 4 different inputs. After the first four rounds, the result from the fourth segment will be fed back to the first segment and so on. 4 different inputs will be encrypted in the architecture. At the 16th round.the first data is encrypted, and 17th the second data is obtained and so on [6].

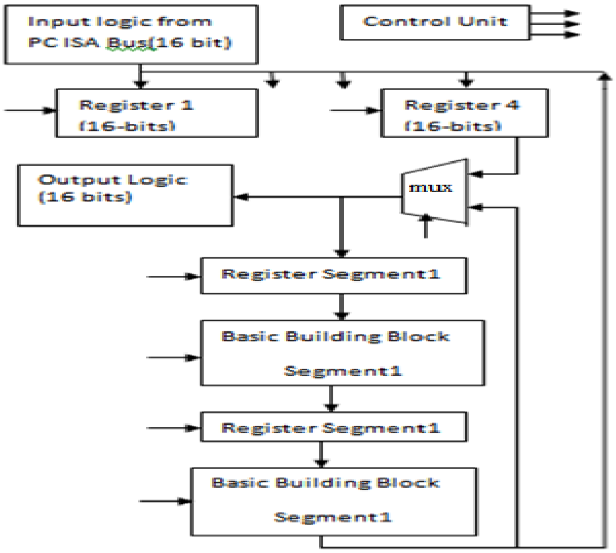


Figure 4. Architecture of Pipelined DES

Control unit is in charge of coordinating operations of components and data flow. The flow chart is shown in Figure **5.**

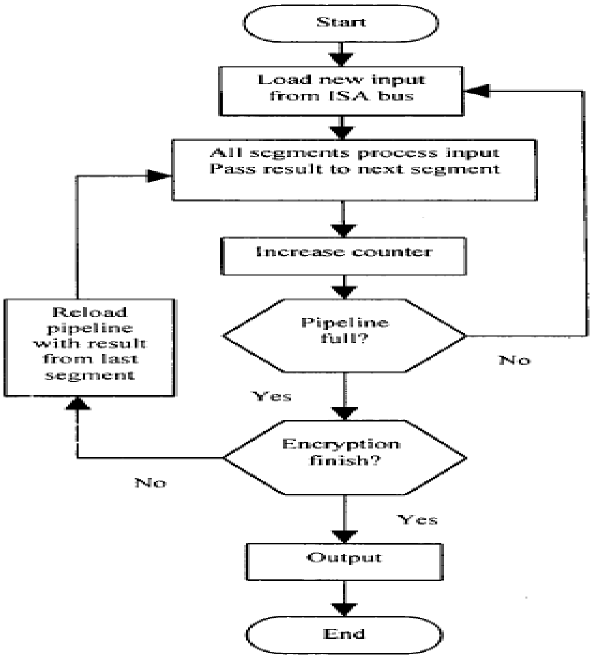

Start
Load new input from ISA bus
All segments process input
Pass result to next segment
Increase counter
Pipeline full?
No
Yes
Reload pipeline with result from last segment
Encryption finish?
No
Yes
Output
End

Figure 5. Flow Chart of Pipelined DES.

6. TRIPLE DES

The following figure 6 shows the triple DES where three keys are used to encrypt the data. The same DES is repeated thrice by using three different keys K1, K2, K3. The plain text is encrypted using key K1 and it is again decrypted using key K2 and again encrypted using K3.

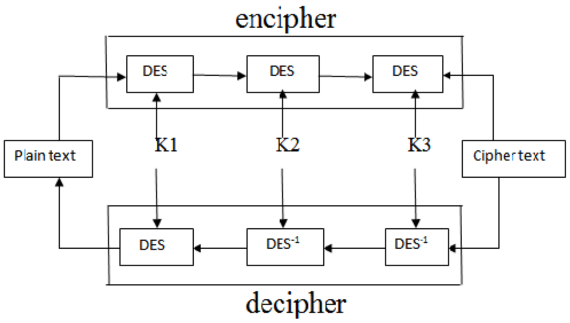


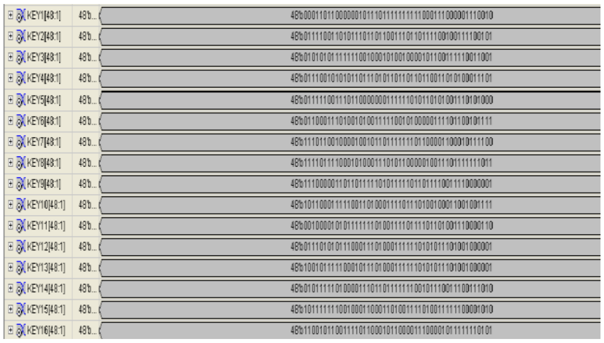
Figure 6. Block Diagram of Triple DES

7. APPLICATIONS

Cryptographic services are required across variety of platforms in a wide range of applications such as secure access to private networks, electronic commerce and health care. The security of conventional encryptions depends on several factors. DES can be used in intensive cryptographic computer application. Applications such as electronic commerce, internet banking sand electronic fund transfer, secure and private communication require better performance cryptographic system.

8. RESULTS

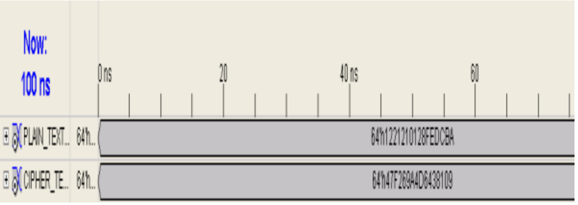
Implementation of DES algorithm was accomplished using Xilinx 8.1 as simulation tool. The design was coded in Verilog. The design achieves a frequency of 111.882 MHz It takes 16 clock cycles latency first time only then encrypts one data block (64-bits) per clock cycle. Initially let the key be **K** = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001. From this key, the sub-keys are generated. The sub-keys generated are shown in fig 7:



```
KEY1[48:1]   48'b...  48'b000110110000001011101111111111000111000001110010
KEY2[48:1]   48'b...  48'b011110011010111011011001110110111100100111100101
KEY3[48:1]   48'b...  48'b010101011111110010001010010000101100111110011001
KEY4[48:1]   48'b...  48'b011100101010110111010110110110110011010100011101
KEY5[48:1]   48'b...  48'b011111001110110000000111111010110101001110101000
KEY6[48:1]   48'b...  48'b011000111010010100111110010100000111101100101111
KEY7[48:1]   48'b...  48'b111011001000010010110111111101100001100010111100
KEY8[48:1]   48'b...  48'b111101111000101000111010110000010011101111111011
KEY9[48:1]   48'b...  48'b111000001101101111101011111011011110011110000001
KEY10[48:1]  48'b...  48'b101100011111001101000111101110100100011001001111
KEY11[48:1]  48'b...  48'b001000010101111111010011110111101101001110000110
KEY12[48:1]  48'b...  48'b011101010111000111010001111110101011101001000001
KEY13[48:1]  48'b...  48'b100101111100010111010001111110101011101001000001
KEY14[48:1]  48'b...  48'b010111110100001110110111111100101110011100111010
KEY15[48:1]  48'b...  48'b101111111001000110001101001111010011111100001010
KEY16[48:1]  48'b...  48'b110010110011110110001011000011100001011111110101
```

Figure 7. Output of Key-Generation

After the generation of sub-keys, the initial text (here M=1221210128FEDCBA) is divided into two halves and then the right half is applied to the s-box. The outputs are xored with function and then the right and left halves are swapped and finally applied to the inverse permutation. The final output i.e. the cipher text is shown in fig 8.



```
PLAIN_TEXT...  64'h...  64'h1221210128FEDCBA
CIPHER_TE...   64'h...  64'h47F269A4D6438109
```

Figure 8. ENCRYPTED DATA

So the initial data is encrypted. Now the data encrypted is again decrypted at the receiver side to obtain the original data being transmitted. So in order to obtain the original data, we should again repeat the same algorithm with the key16 for the round1, key15 for the round2 and so on. The cipher text for the different inputs is shown in the figure9.

Plain Text	Cipher Text
64’h12212101 28FEDCBA	64’h47F269A4 D6438109
64’h01234567 89ABCDEF	64’h85E81354 0F0AB405.
64’h97530281 9A7D4B7C	64’hF8C675DA 6904AB21

Figure 9 Various Plain and cipher text obtained by DES

The pipelined data with 4 segments is implemented. The output of sbox1 is shown in the figure10.

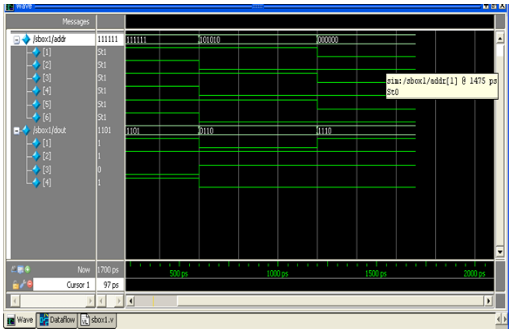


Figure 10. Pipelined output of Sbox1

First input is first fed into segment 1. After being processed by basic building block of segment 1, the output is fed into register of segment 2. At the same time, the second input is fed into segment I. There are two inputs now, first input in segment 2, second input in segment 1, in pipeline now. The final pipelined output is shown in Fig 11.

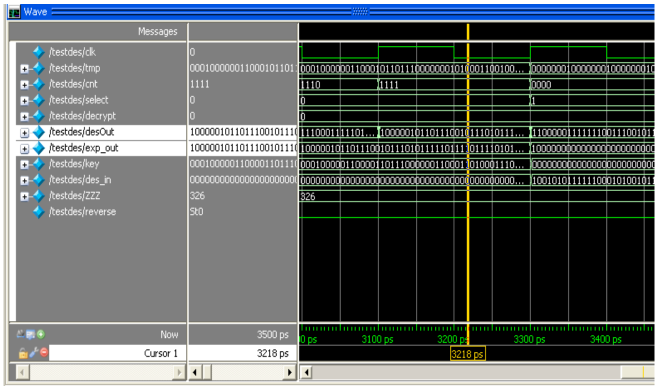


Figure 11. Output of Pipelined Des

During the triple DES, the text is encrypted using K1 and then decrypted using K2 and again encrypted using K3 as shown in figure 12.

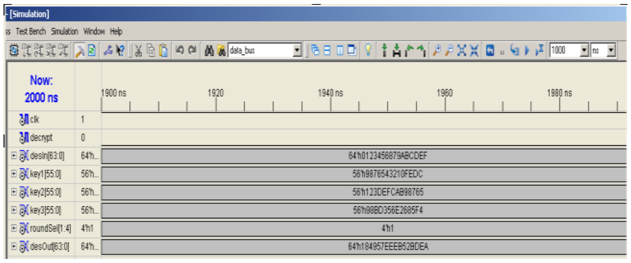
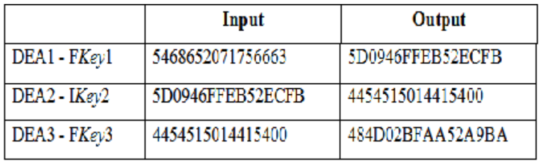


Figure 12. Output of Triple DES

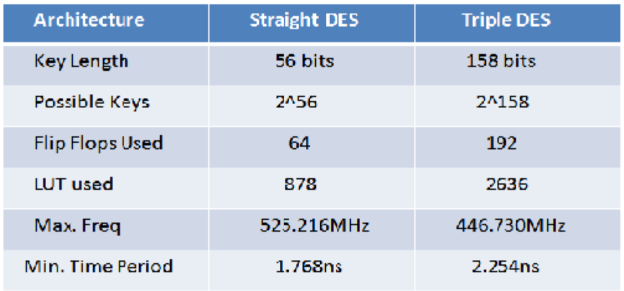
For triple DES, at step 1, the input to DEA1 is *“The quic” i.e.,* P1= 5468652071756663, and the output of DEA1 is “5D0946FFEB52ECFB”. At step 2, the input to DEA2 is the output of DEA1, and the output of DEA2 is “4454515014415400”. At step 3, the input to DEA3 is the output of DEA2, and the output of DEA3 is “484D02BFAA52A9BA”. The output of DEA3 is the cipher text *C*1. The tabular representation of the outputs at various stages of encryption is shown in the figure 13.



	Input	Output
DEA1 - F*Key*1	5468652071756663	5D0946FFEB52ECFB
DEA2 - I*Key*2	5D0946FFEB52ECFB	4454515014415400
DEA3 - F*Key*3	4454515014415400	484D02BFAA52A9BA

Figure 13. Tabular output of Triple DES

The comparison of the straight forward architecture and triple DES is shown in the figure 14. The time taken for encryption is less in pipelined DES.



Architecture	Straight DES	Triple DES
Key Length	56 bits	158 bits
Possible Keys	2^56	2^158
Flip Flops Used	64	192
LUT used	878	2636
Max. Freq	525.216MHz	446.730MHz
Min. Time Period	1.768ns	2.254ns

Figure 14. Comparison of DES and triple DES

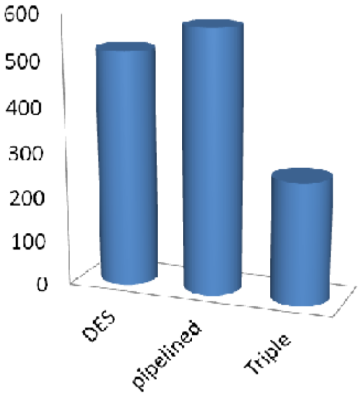


Figure 15. Maximum Freq comparison

The above graph in figure 15 portrays the comparison between the maximum operating frequency and throughput of the existing work and the results obtained. As known pipelining increases the operating frequency as well as the area but the triple DES increases the security of the system.

9. CONCLUSION

Developed a Straight forward DES architecture using the 64-bit key. The sub- key is generated and then the data which is to be sent is encrypted and converted into cipher text. Later, the data is encrypted using a 4 segment pipelined DES and the encrypted time for both is observed & the triple DES is implemented using three independent keys.

The pipelined DES consumes less hardware resource than fully pipelined DES does, and provides more throughput than practical DES. The Triple DES consumes more hardware and also the frequency of operation is low compared to straight DES. But the basic advantage of using triple DES is that it is more secured than that of DES and pipelined DES.

10. REFERENCES

[1] Teo Pock Cheung, "*Implementation of Pipelined Data Encryption Standard (DES) Using Altera CPLD*" in Proc. IEEE Trans circuits syst vol.74, no.13, 1759-1763, 2007.

[2] The DES Algorithm Illustrated by J. Orlin Grabbe.

[3] Schneier, B. "*Applied Cryptography, Protocols, Algorithms, and Source Code*" in Proc. IEEE Int Symp circuits syst (ISCAS), 2005 pp.592-595,2003.

[4] Ahmed Zure Sha'meri, "*DES Cryptographic System for Information Security*" in Proc. IEEE Trans circuits syst vol.53, no.11,1165-1169, 2002.

[5] Wong, K., Wark, M., Dawson, E.: A Single-Chip FPGA *Implementation of the Data Encryption Standard (des) Algorithm*. In: IEEE Globecom Communication Conf., Sydney, Australia (2002) 827–832.

[6] FPGA implementation of des using pipelining concept with skew core key-scheduling By vishwanath patel, r. c. joshi, a. k. saxena.

[7] J Wilcox, D., Pierson, L., Robertson, P., Witzke, E.L., Gass, K.: *A DES basic suitable for network encryption at 10 Gbs and beyond*. In: CHESS 99, LNCS 1717 (1999) 37–48.

[8] Kaps J, Fast DES implementations for FPGAs and its application to a Universal key-search machine. In: Proc. 5th Annual Workshop on selected areas in cryptography.