IMPLEMENTATION OF DATA ENCRYPTION STANDARD ALGORITHM USING VERILOG

1) Ashwini.R.Patil **2) Prof.Pramod.Patil**
HIT, Nidasoshi,ECEHIT,Nidasoshi
ashwinirpatil07@gmail.com
9620411759

3) Rajeshwari.Khot4)Laxmi.Patil

HIT,Nidasoshi,ECEHIT,Nidasoshi,ECE

4）**Rajeshwarikhot11@gmail.com**

laxmipatil0217@gmail.com

5) Anusha.Fernandes

HIT,Nidasoshi,ECE

Anushafernandes4@gmail.com

ABSTRACT:

The data encryption standard is a symmetric key algorithm for the encryption of electric data.It is called as symmetric because same algorithm and key are used for encryption and decryption. DES is a block cipher, it encrypts data in 64 bit blocks. 64 bit blocks of plaintext goes in one end of the algorithm and 64 bit block of cipher text comes out other end.The key length is 56 bits. To accomplish encryption , most secret key algorithm use two main techniques known as substitution and permutation .Substitution is a simply a mapping of one value to another. Whereas permutation is a reordering of bit position for each of the inputs. These techniques are used number of times in iteration called rounds. S-boxes are used basically non-linear substitution table,where either the output is smaller than the input. It will be implemented by using the tool Xilinx 13.1.Simulator used is ISE .Language used for this implementation is Verilog.

INTRODUCTION:

There are mainly 2 types of cryptography are there–**symmetric**or **secret key** cryptography and **asymmetric** or **public key** cryptography. Symmetrickeycryptography is the oldest type whereas asymmetric cryptography is only beingused publicly since the late 1970’s. Secret key cryptography goes back to at least Egyptian times and is of concern here. It involves the use of only one key which is used for both encryption and decryption (hence the use of the term symmetric). Figure 2.1 depicts this symmentric key cryptography.

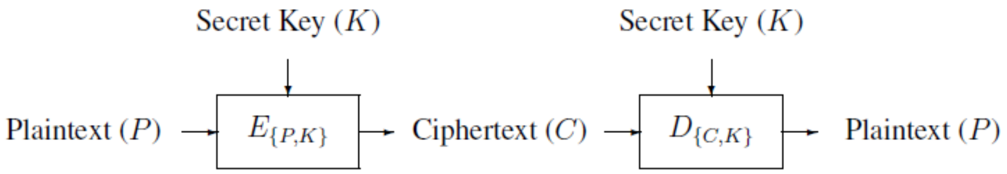

Figure 2.1: Secret key encryption.

In encryption process we have mainly 2 opertions, Substitution and Permutation. Substitution is simply a mapping of one value toanother value whereas permutation is a reordering of the bit positions for each of the inputs.These techniques are used a number of times in iterations called **rounds**. More rounds means more secure algorithm. A non-linearity is alsointroduced into theencryption so that decryption will be computationally infeasible without the secret key. This is achieved with the use of **S-boxes** which are basicallynon-linear substitution tables where either the output is smaller than the input or viceversa.

The main problem arised in secret key is key distribution.For this purpose sender and receiver must know the key. Thiswould have to be communicated over some secure channel which, unfortunately, is notthat easy to achieve. As will be seen later, public key cryptography provides a solutionto this.

BRIEF HISTORY OF DES:

Up till we have a standard algorithm for encryption as Data Encryption Standard Algorithm. But now a days it is replaced with a new standard known as the **Advanced Encryption Standard (AES)**. DES is a 64 bit **block cipher** which means that it encrypts data 64bits at a time. Stream cipher encrypts only one bit at a time.DES was the result of a research project set up by International Business Machines(IBM) corporation in the late 1960’s which resulted in a cipher known as LUCIFER. Inthe early 1970’s it was decided to commercialise LUCIFER and a number of significantchanges were introduced. IBM was not the only one involved in these changes asthey sought technical advice from the National Security Agency

(NSA) (other outsideconsultants were involved but it is likely that the NSA were the major contributorsfrom a technical point of view). The altered version of LUCIFER was put forward asa proposal for the new national encryption standard requested by the National Bureauof Standards (NBS). It was finally adopted in 1977 as the Data Encryption Standard DES (FIPS PUB 46).

Some of the changes made to LUCIFER have been the subject of much controversyeven to the present day. The most notable of these was the key size. LUCIFER useda key size of 128 bits however this was reduced to 56 bits for DES. Even though DESactually accepts a 64 bit key as input, the remaining eight bits are used for paritychecking and have no effect on DES's security. Outsiders were convinced that the 56bit key was an easy target for a brute force attack4 due to its extremely small size. Theneed for the parity checking scheme was also questioned without satisfying answers.

Another controversial issue was that the S-boxes used were designed under classifiedconditions and no reasons for their particular design were ever given. This led people to assume that the NSA had introduced a "trapdoor" through which they could decryptany data encrypted by DES even without knowledge of the key. One startling discoverywas that the S-boxes appeared to be secure against an attack known as **DifferentialCryptanalysis** which was only publicly discovered by Biham and Shamir in 1990.This suggests that the NSA were aware of this attack in 1977; 13 years earlier! In factthe DES designers claimed that the reason they never made the design specifications forthe S-boxes available was that they knew about a number of attacks that weren't public knowledge at the time and they didn't want them leaking - this is quite a plausibleclaim as differential cryptanalysis has shown. However, despite all this controversy, in1994 NIST reaffirmed DES for government use for a further five years for use in areasother than "classified".

DES of course isn't the only symmetric cipher. There are many others, each with varyinglevels of complexity. Such ciphers include: IDEA, RC4, RC5, RC6 and the newAdvanced Encryption Standard (AES). AES is an important algorithm and was originallymeant to replace DES (and its more secure variant triple DES) as the standardalgorithm for non-classifiedmaterial. However as of 2003, AES with key sizes of 192and 256 bits has been found to be secure enough to protect information up to top se-cret. Since its creation, AES had underdone intense scrutiny as one would expect foran algorithm that is to be used as the standard. To date it has withstood all attacks butthe search is still on and it remains to be seen whether or not this will last. We willlook at AES later in the course.

INNER WORKINGS OF DES:

DES (and most of the other major symmetric ciphers) is based on a cipher known as the **Feistel block cipher**. This was a block cipher developed by the IBM cryptographyresearcher Horst Feistel in the early 70's. It consists of a number of rounds whereeach round contains bit-shuffling, non-linear substitutions (S-boxes) and exclusive ORoperations. Most symmetric encryption schemes today are based on this structure(known as a **feistel network**).

As with most encryption schemes, DES expects two inputs - the plaintext to be encryptedand the secret key. The manner in which the plaintext is accepted, and the keyarrangement used for encryption and decryption, both determine the type of cipher itis. DES is therefore a symmetric, 64 bit **block cipher** as it uses the same key for bothencryption and decryption and only operates on 64 bit blocks of data at a time5 (be theyplaintext or ciphertext). The key size used is 56 bits, however a 64 bit (or eight-byte)key is actually input. The least significant bit of each byte is either used for parity (oddfor DES) or set arbitrarily and does not increase the security in any way. All blocks arenumbered from left to right which makes the eight bit of each byte the parity bit.

Once a plain-text message is received to be encrypted, it is arranged into 64 bit blocksrequired for input. If the number of bits in the message is not evenly divisible by 64, then the last block will be padded. Multiple permutations and substitutions are incorporatedthroughout in order to increase the difficulty of performing a cryptanalysis onthe cipher. However, it is generally accepted that the initial and final permutations offerlittle or no contribution to the security of DES and in fact some software implementations omit them (although strictly speaking these are not DES as they do not adhere to the standard).

OVERALL STRUCTURE:

Figure 2.2 shows the sequence of events that occur during an encryption operation. DES performs an initial permutation on the entire 64 bit block of data. It is then splitinto 2, 32 bit sub-blocks, Li and Ri which are then passed into what is known as a**round** (see figure 2.3), of which there are 16 (the subscript i in Li and Ri indicatesthe current round). Each of the rounds are identical and the effects of increasing theirnumber is twofold - the algorithms security is increased and its temporal efficiencydecreased. Clearly these are two conflicting outcomes and a compromise must bemade. For DES the number chosen was 16, probably to guarantee the elimination ofany correlation between the ciphertext and either the plaintext or key6. At the end of the16th round, the 32 bit Li and Ri output quantities are swapped to create what is knownas the **pre-output**. This [R16, L16] concatenation is permuted using a function whichis the exact inverse of the initial permutation. The output of this final permutation isthe 64 bit ciphertext.

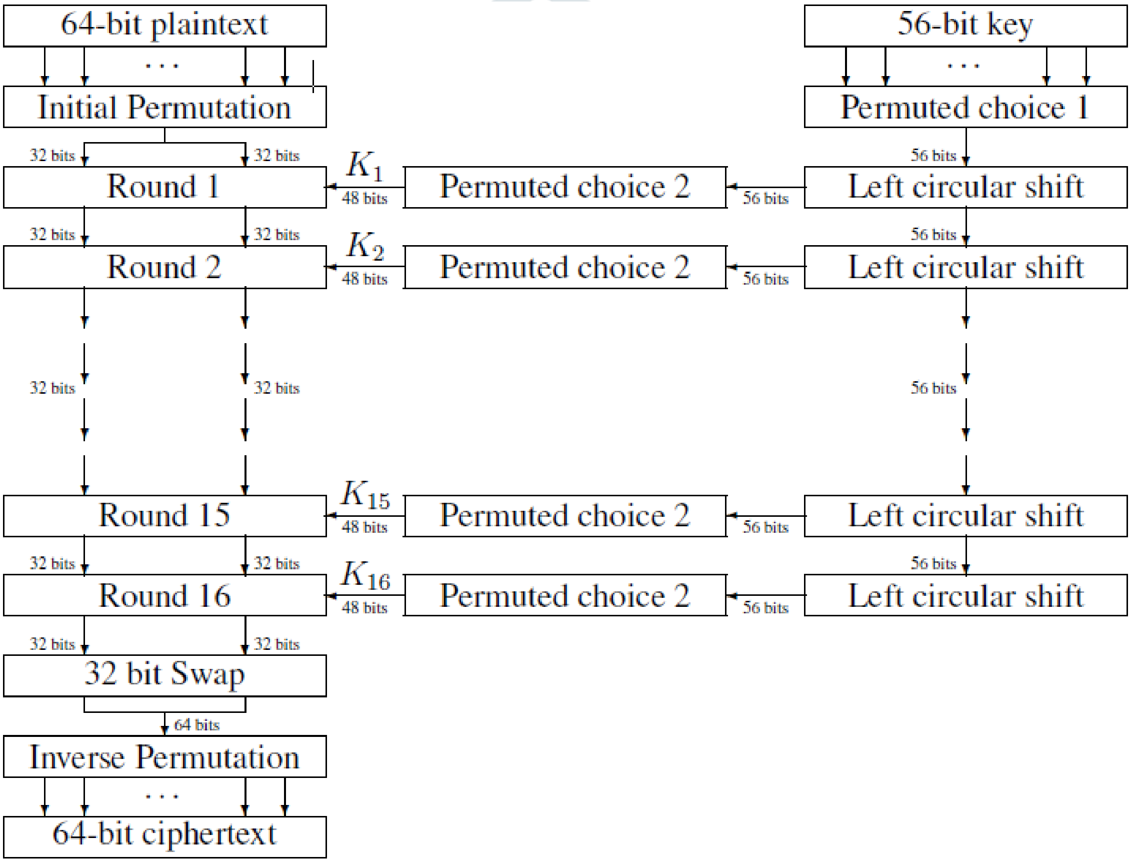

Figure 2.2: Flow Diagram of DES algorithm for encrypting data.

So in total the processing of the plaintext proceeds in three phases as can be seen from the left hand side of figure 2.2:

1. Initial permutation (**IP** - defined in table 2.1) rearranging the bits to form the“permuted input”.

2. Followed by 16 iterations of the same function (substitution and permutation).The output of the last iteration consists of 64 bits which is a function of theplaintext and key. The left and right halves are swapped to produce the preoutput.

3. Finally, the preoutput is passed through a permutation (**IP**−1 - defined in table2.1) which is simply the inverse of the initial permutation (**IP**). The output of**IP**−1 is the 64-bit ciphertext.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Initial Pemutation

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Expansion Permutation Table(E)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Permutation Function (P)

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Inverse Initial permutation

Table 2.1: Permutation Tables Used in DES

As figure 2.2 shows, the inputs to each round consist of the Li,Ri pair and a 48 bit**subkey**which is a shifted and contracted version of the original 56 bit key. The use of the key can be seen in the right hand portion of figure 2.2:

- Initially the key is passed through a permutation function (**PC**1 - defined in table 2.2)
- For each of the 16 iterations, a subkey (**K**i) is produced by a combination of a leftcircular shift and a permutation (**PC**2 - defined in table 2.2) which is the same for each iteration. However, the resulting subkey is different for each iteration because of repeated shifts.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Input Key

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Permuted Choice-1(PC-1)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Permuted Choice-2 (PC-2)

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of bit shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Schedule Of Left Shift

DETAILS OF INDIVISUAL ROUNDS:

Details of an individual round can be seen in figure 2.3. The main operations on thedata are encompassed into what is referred to as the **cipher function** and is labeled ***F***. This function accepts two different length inputs of 32 bits and 48 bits and outputs a single 32 bit number. Both the data and key are operated on in parallel, however the operations are quite different. The 56 bit key is split into two 28 bit halves Ci and Di (C and D being chosen so as not to be confused with L and R). The value of the key used in any round is simply a left cyclic shift and a permuted contraction of that used in the previous round. Mathematically, this can be written as

- Ci= Lcsi($C_{i-1)}$ $D_{i=}$ Lcsi($D_{i-1)}$ $K_{i=}$ PC2(Ci,Di)

whereLcsi is the left cyclic shift for round i, Ci and Di are the outputs after the shifts,PC2(.) is a function which permutes and compresses a 56 bit number into a 48 bit number and Ki is the actual key used in round i. The number of shifts is either one or two and is determined by the round number i. For i = {1, 2, 9, 16} the number of shifts is one and for every other round it is two (table 2.2).

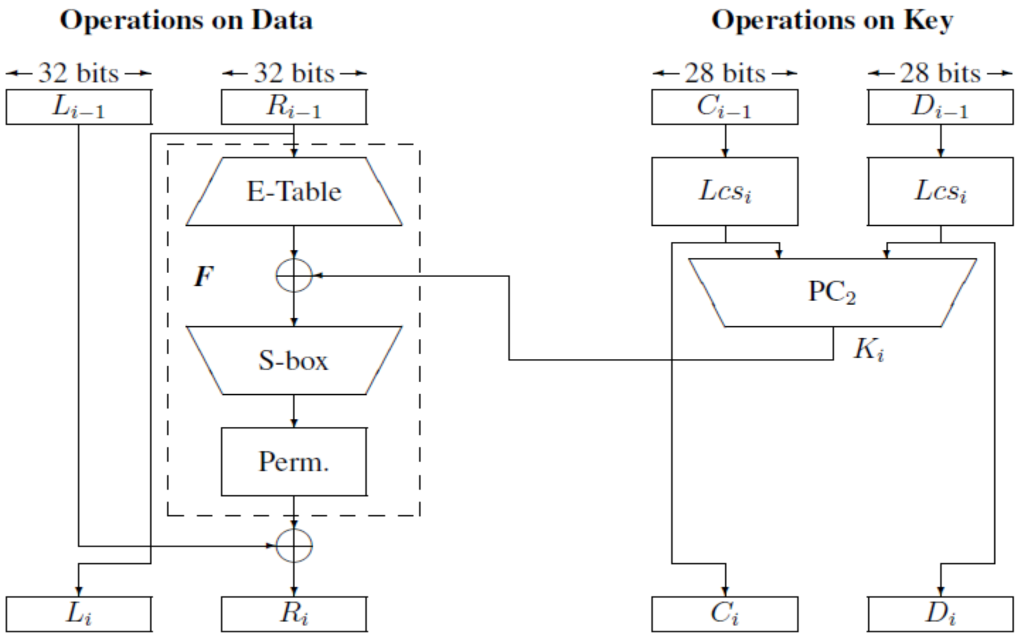

Fig 2.3: Details OfIndivisual Rounds

The common formulas used to describe the relationships between the input to one round and its output (or the input to the next round) are:

Li= Ri-1 , Ri=Li-1 +F(Ri-1,Ki)

where L and R have their usual meaning and ***F***(.) is the cipher function. This function ***F*** is the main part of every round and consists of four separate stages (see figure 2.4):

1. The E-box expansion permutation - here the 32-bit input data from Ri−1 is expanded and permuted to give the 48 bits necessary for combination with the 48 bit key (defined in table 2.1). The E-box expansion permutation delivers a larger output by splitting its input into 8, 4-bit blocks and copying every first and fourth bit in each block into the output in a defined manner. The security offered by this operation comes from one bit affecting two substitutions in the S-boxes This causes the dependency of the output bits on the input bits to spread faster,and is known as the avalanche affect.

2. The bit by bit addition modulo 2 (or exclusive OR) of the E-box output and 48bit subkey Ki.

3. The S-box substitution - this is a highly important substitution which accepts a48-bit input and outputs a 32-bit number (defined in table 2.3). The S-boxes arethe only non-linear operation in DES and are therefore the most important partof its security. They were very carefully designed although the conditions theywere designed under has been under intense scrutiny since DES was released.The reason was because IBM had already designed a set of S-boxes which werecompletely changed by the NSA with no explanation why.

The input to the S-boxes is 48 bits long arranged into 8, 6 bit blocks (b1, b2, . . . , b6). There are 8 S-boxes (S1, S2, . . . , S8) each of which accepts one of the 6 bit blocks. The output of each S-box is a four bit number. Each of the S-boxes can be thought of as a 4 × 16 matrix. Each cell of the matrix is identified by a coordinate pair (i, j), where 0 _ i _ 3 and 0 _ j _ 15. The value of i is taken as the decimal representation of the first and last bits of the input to each S-box, i.e. Dec(b1b6) = i and the value of j is take from the decimal representation of the inner four bits that remain, i.e. Dec(b2b3b4b5) = j. Each cell within the S-box matrices contains a 4-bit number which is output once that particular cell is selected by the input.

4. The P-box permutation - This simply permutes the output of the S-box without changing the size of the data (defined in table 2.1). It is simply a permutation and nothing else. It has a one to one mapping of its input to its output giving a 32 bit output from a 32 bit input.

OTHER POINTS OF VIEW:

Having looked at DES in some detail a brief look at some other points is in order.These include decryption, modes of operation, security etc.

MODES OF OPERATION:

The DES algorithm is a basic building block for providing data security. To applyDES in a variety of applications, five modes of operation have been defined which cover virtually all variation of use of the algorithm and these are shown in table 2.4.

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7		9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2		8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	1	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	8	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	20	8	13	15	12	9	0	3	5	6	11

Fig 2.3: S-Box Table

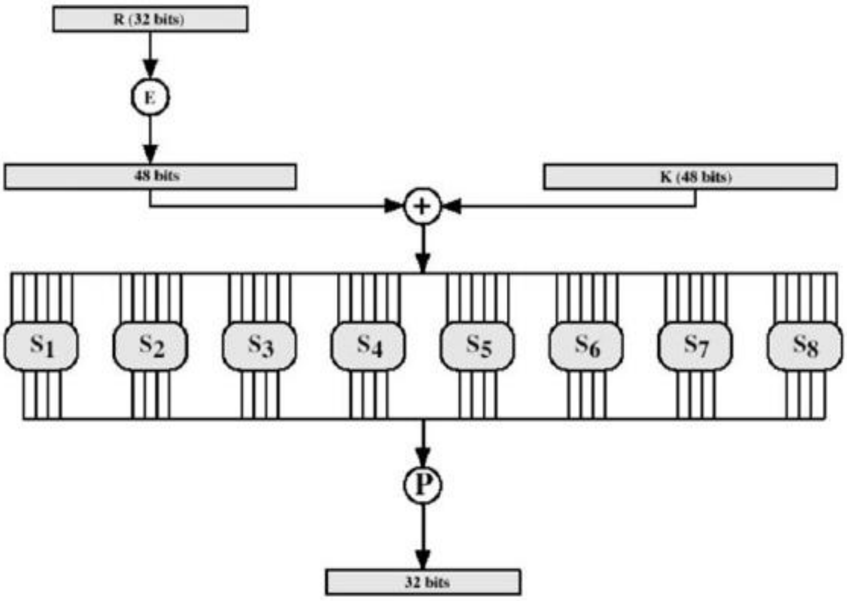

Fig 2.4: The Complex F Function Of the DES algorithm

Modes Of Operations:

Mode	Description	Typical Application
Electronic Code Book(ECB)	Each block of 64 plaintext bits is encoded independently using the same key	Secure Transmission of single values.
Cipher Block Chaining(CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of cipher text.	Authentication
Cipher Feedback(CFB)	Input is processed J bits at a time.	General purpose stream oriented transmission.
Output feedback(OFB)	Similar to CFB except that the input to the encryption	Stream oriented transmission over noisy

	algorithm is the preceding DES output.	channel.
Counter(CTR)	Each block of plain text is XORed with an encrypted counter.The counter is incremented for each subsequent block.	Useful for high speed requirements.

Table 2.4: DES Modes Of Operations

DES DECRYPTION:

The decryption process with DES is essentially the same as the encryption process and is as follows:

Use the ciphertext as the input to the DES algorithm but use the keysKi in reverse order. That is, use K16 on the first iteration, K15 on the second until K1which is used on the 16th and last iteration.

CONCLUSION:

In this paper, we have discussed the High Level Language implementation of DES. Our design is efficient in comparison to other software implementations and it utilizes less hardware resource on FPGA and takes less development time.

REFERENCES:

1) Journal of Computer Science of Newports Institute of Communications and Economics Volume 5, Issue-2014, ISSN: 2226-3683.
2) Cunsheng Ding Department of Computer Science Hong Kong University of Science and Technology Clearwater Bay, Kowloon, Hong Kong, CHINA.