# **Lab 10**

# **Implementing Data Encryption Standard (DES)**

|  |  |
| --- | --- |
| **Name:** | **Student ID:** |

## **10.1 Data Encryption Standard**

## **10.1 Objective**

The Objectives of this lab are:

* Perform the initial permutation and final permutation.
* Constructing the DES function
* Designing the Key Generator function
* Implementing the Cipher algorithm
* Encrypt data using the Data Encryption Standard (DES) algorithm.
* Validate the DES implementation with various inputs.

**10.2 Background/Scenario**

The Data Encryption Standard (DES) is a historically significant symmetric-key block cipher encryption algorithm that played a pivotal role in the early days of computer cryptography. Developed by IBM in the 1970s, DES was adopted as a federal standard in the United States and became one of the most widely used encryption schemes in the world for several decades. In this lab, you will implement Data Encryption Standard (DES) using Verilog HDL, and subsequently employ it to encrypt your data. or the submission, ensure that you provide all your modules and results where applicable under the relevant task headings.

**Task 1 – Initial and Final Permutation Tables**

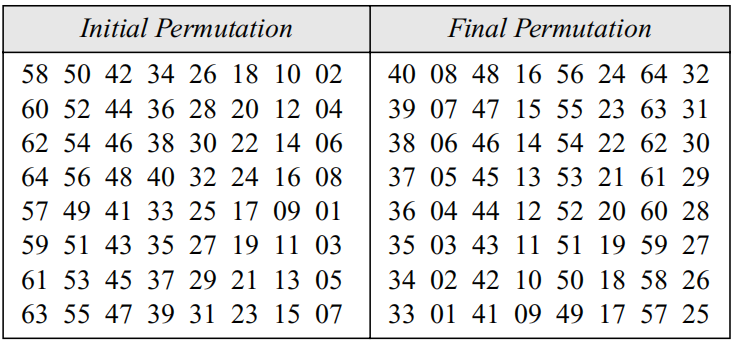


Figure 1: Initial and Final Permutation tables

1. Create a module that returns as output the initial permutation table consisting of 64 bits. Refer to Figure 1 for the change in position of each bit.
2. Create a module that returns as output the final permutation table consisting of 64 bits. Refer to Figure 1 for the change in position of each bit.

**Task 2 – Constructing the DES function**

The DES function applies a 48-bit key to the rightmost 32 bits (RI−1) to produce a 32-bit output. This function is made up of four sections: an expansion P-box, a whitener (that adds key), a group of S-boxes, and a straight P-box

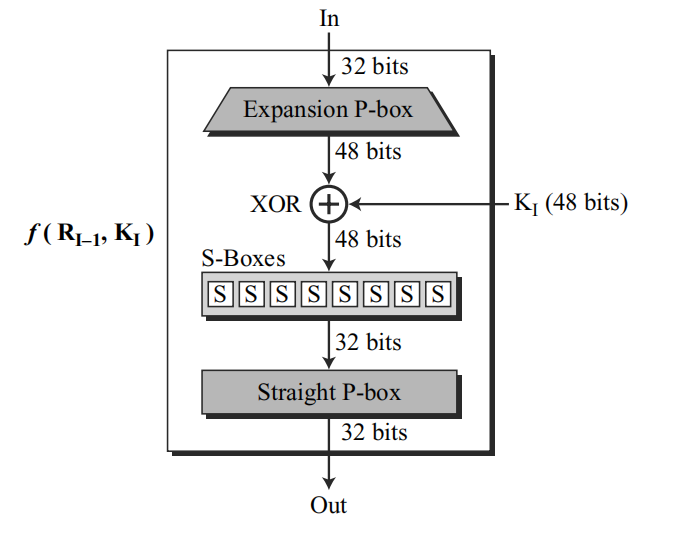


Figure 2: DES function

**Task 2.1 - Creating the Expansion P-Box Module**

Use Figure 3 to create a module that takes as input 32-bit and returns as output a 48-bit text.

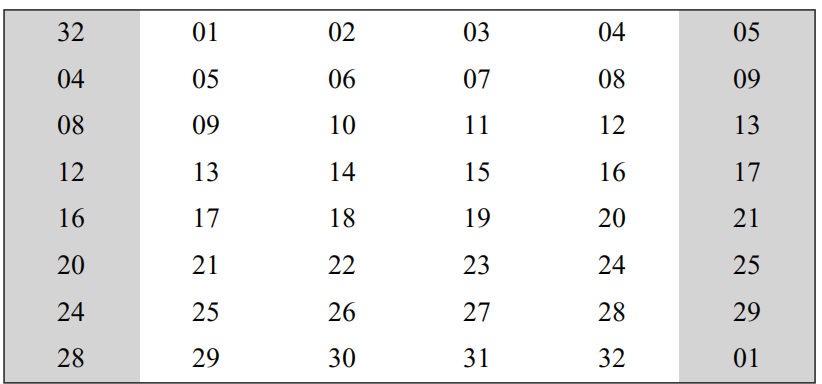


Figure 3: Expansion P-box Table

**Task 2.2 - Whitener (XOR)**

After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the keys are 48-bits in length. Also note that the round key is used only in this operation.

Create a module for XOR that implements this functionality.

**Task 2.3 - Construction of S-boxes.**

DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box. The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text.

1. Construct the table for S-box 1 using the table below.

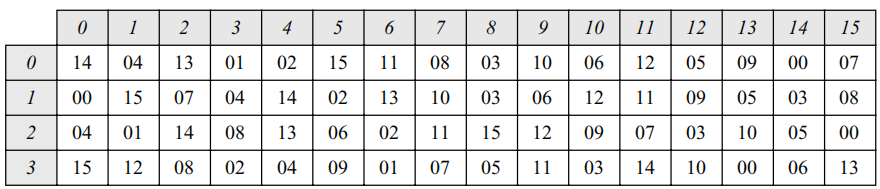


Table 1: S-box 1

1. Construct the table for S-box 2 using the table below.

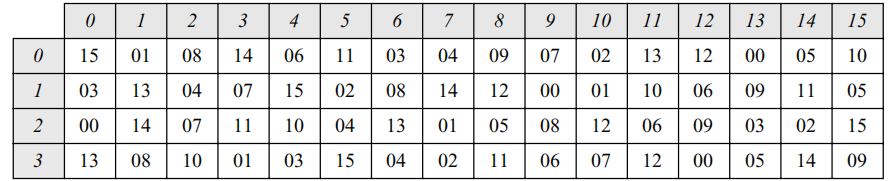


Table 2: S-box 2

1. Construct the table for S-box 3 using the table below.

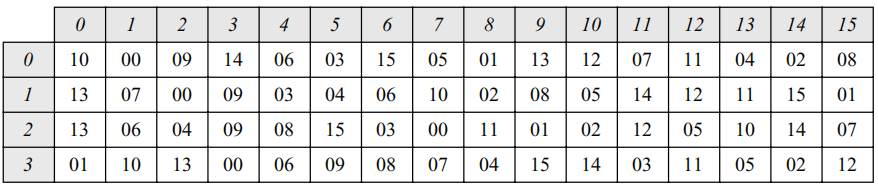


Table 3: S-box 4

1. Construct the table for S-box 4 using the table below.

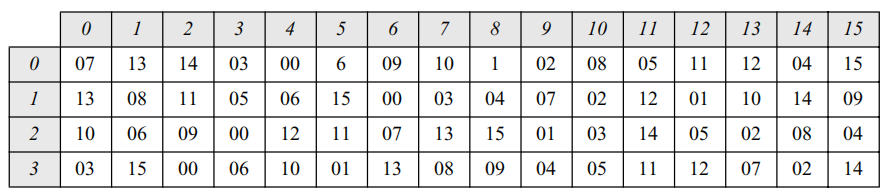


Table 4: S-box 4

1. Construct the table for S-box 5 using the table below.

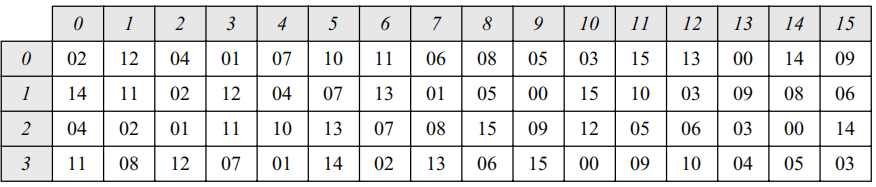


Table 5: S-box 5

1. Construct the table for S-box 6 using the table below.

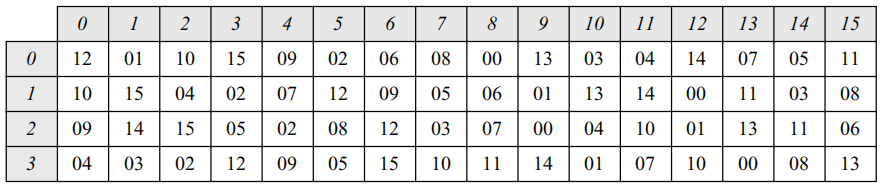


Table 6: S-box 6

1. Construct the table for S-box 7 using the table below.

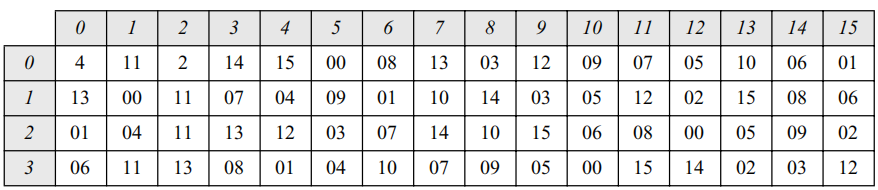


Table 7: S-box 7

1. Construct the table for S-box 8 using the table below.

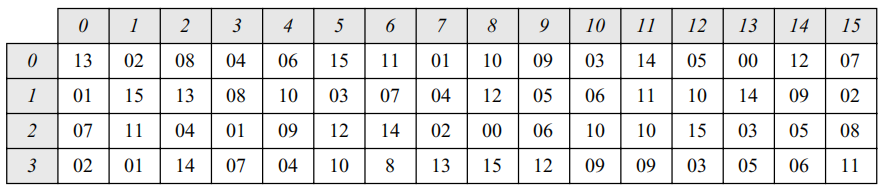


Table 8: S-box 8

**Task 2.4 - Performing Straight Permutation**

Create a module that takes as input a 32-bit text and returns another 32-bit text. The input/output relationship follows the table below. For example, the seventh bit of the input becomes the second bit of the output.

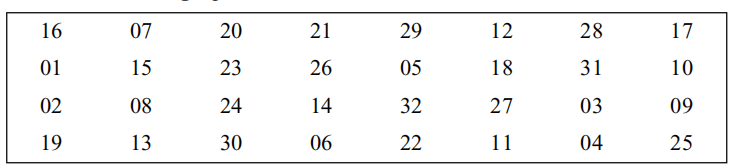


Figure 4: Straight Permutation Table

**Task 2.5 - Designing the DES function**

Integrate the functions previously designed in this task to assemble the complete DES function, mirroring the structure illustrated in Figure 2.

**Task 3 – Creating the DES Cipher**

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds. The cipher is used at the encryption site; the reverse cipher is used at the decryption site. In this lab, we will only be creating the cipher for encryption.

Note that there are two approaches available for constructing the cipher in your DES implementation. In the first approach, illustrated in Figure 5, the final round does not include a swapper. Alternatively, you can opt for the second approach, where all rounds are kept uniform, and an additional swapper is introduced at the end to counteract the effect of the swapper in the 16th round. You may choose either approach for your DES implementation.

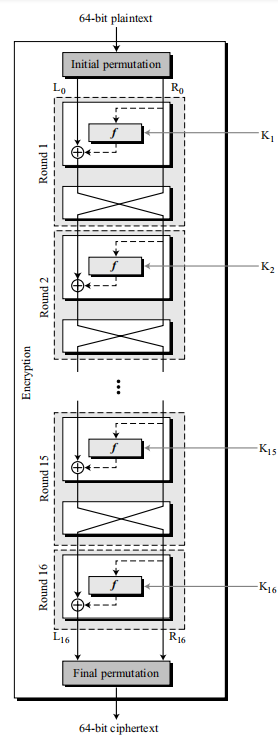


Figure 5: First Approach

Below is the pseudocode for the DES cipher:

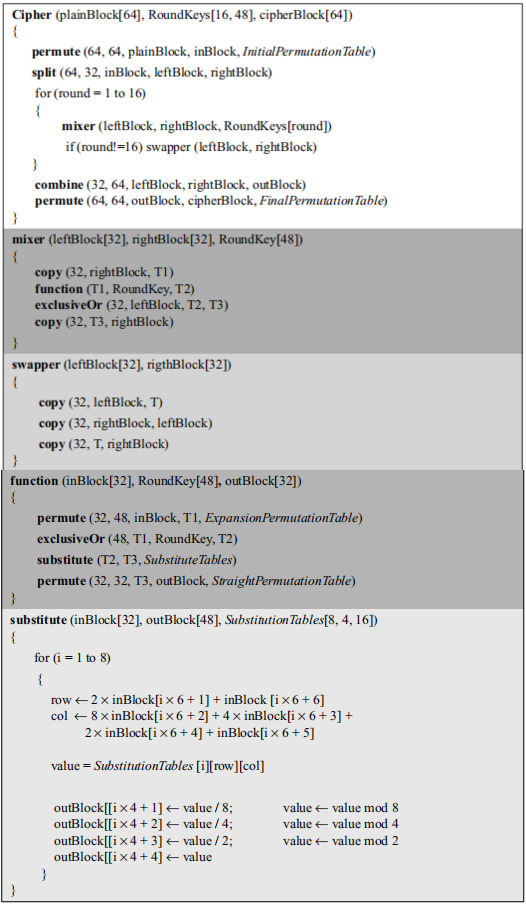


Figure 6: DES Algorithm

**Task 4 – Key Generation**

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. However, the cipher key is normally given as a 64-bit key in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.

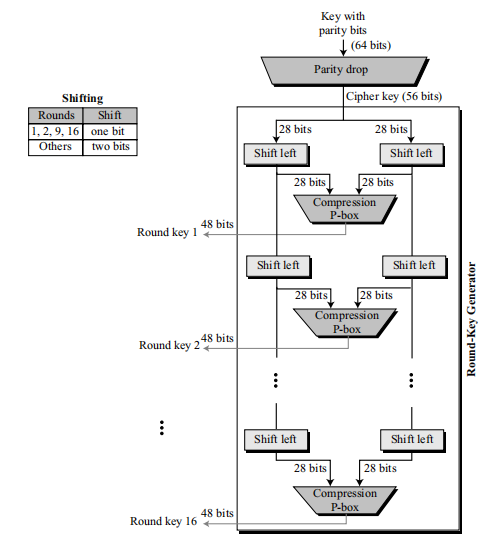


Figure 7: Key Generation

**Task 4.1 - Constructing the P-Box.**

Create a function that takes 64 bits as input and uses the parity drop table to return a 56-bit cipher key. The parity drop permutation table is given below.

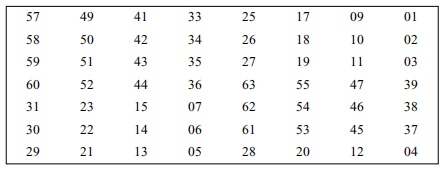


Figure 8: Parity Drop Table

**Task 4.2 - Constructing the shift-left function.**

After the straight permutation, the key is divided into two 28-bit parts. Each part is shifted left (circular shift) one or two bits. In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits. The two parts are then combined to form a 56-bit part.

Design a function that employs the algorithm depicted in Figure 9. This function should accept a 28-bit block and the specific number of shifts needed, returning the resulting 28-bit block. Refer to Table 9 for the exact number of shifts required for each round.

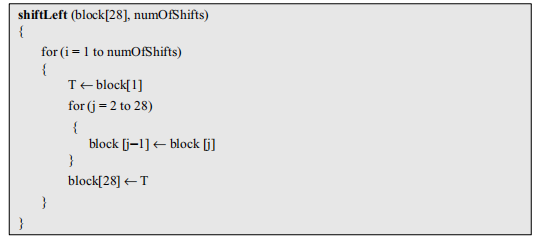


Figure 9: Shift Left Algorithm



Table 9: Number of bits shifts for each round

**Task 4.3 - Key Compression Table.**

The compression permutation (P-box) changes the 58 bits to 48 bits, which are used as a key for a round.

Create a module that accepts a 58-bit input and based on the provided permutation table, generates a 48-bit output.

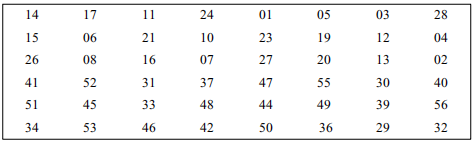


Figure 10: Key Compression Table

**Task 4.4 - Designing the Key Generator.**

Create a function that generates the keys required for all 16 rounds following the algorithm presented in Figure 11. To better grasp the operation of this function, you can also refer to Figure 7.

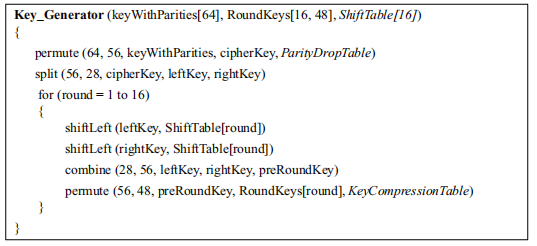


Figure 11: Round Key Generator Algorithm

**Task 5 – Validating the DES cipher.**

1. Run the key generator function for “AABB09182736CCDD” to obtain the round keys for all 16 rounds.
2. Run the DES cipher for “123456ABCD132536”.
3. Verify your output. Your results should match the values shown in Figure 12.

Note: For step 1 and 2, you will have to convert the plaintext into binary first.

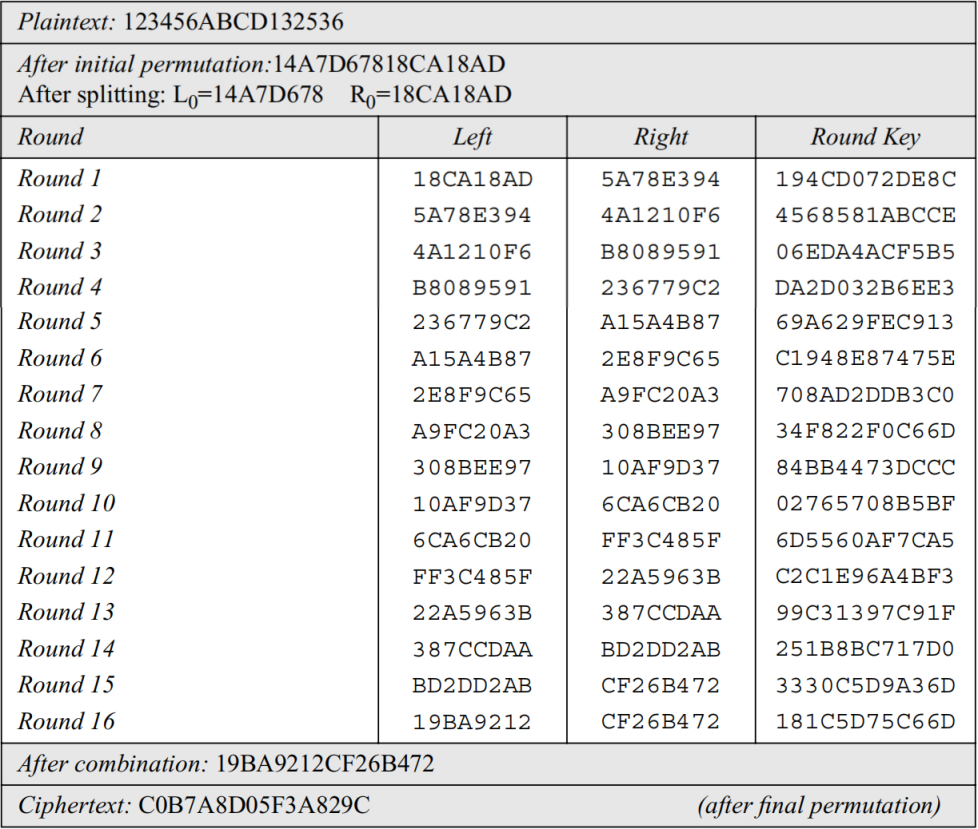


Figure 12: Trace of Data

## **Assessment Rubric**

**Lab 10**

**Data Encryption Standard (DES)**

|  |  |
| --- | --- |
| **Name:** | **Student ID:** |

**Points Distribution**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **LR 2**  **Code** | **LR5**  **Results** | **LR9**  **Report** |
| Task 1 | 5 |  |  |
| Task 2.1 | 5 |  |  |
| Task 2.2 | 5 |  |  |
| Task 2.3 | 5 |  |  |
| Task 2.4 | 5 |  |  |
| Task 2.5 | 5 |  |  |
| Task 3 | 15 |  |  |
| Task 4.1 | 5 |  |  |
| Task 4.2 | 5 |  |  |
| Task 4.3 | 5 |  |  |
| Task 4.4 | 10 |  |  |
| Task 5 | 5 | 10 |  |
| Total | /75 | /10 | /5 |
| **CLO Mapped** | CLO 2 | CLO 2 | CLO2 |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Affective Domain Rubric** | | **Points** | **CLO Mapped** |
| AR 7 | Report Submission | /10 | CLO 2 |

|  |  |  |
| --- | --- | --- |
| **CLO** | **Total Points** | **Points Obtained** |
| 2 | 90 |  |
| 2 | 10 |  |
| **Total** | **100** |  |

*For description of different levels of the mapped rubrics, please refer the provided Lab Evaluation Assessment Rubrics and Affective Domain Assessment Rubrics.*

**Lab Evaluation Assessment Rubric**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Assessment Elements** | **Level 1: Unsatisfactory**  **Points 0-1** | **Level 2: Developing**  **Points 2** | **Level 3:Good**  **Points 3** | **Level 4:Exemplary**  **Points 4** |
| **LR2** | **Program/Code/ Simulation Model/ Network Model** | Program/code/simulation model/network model does not implement the required functionality and has several errors. The student is not able to utilize even the basic tools of the software. | Program/code/simulation model/network model has some errors and does not produce completely accurate results. Student has limited command on the basic tools of the software. | Program/code/simulation model/network model gives correct output but not efficiently implemented or implemented by computationally complex routine. | Program/code/simulation /network model is efficiently implemented and gives correct output. Student has full command on the basic tools of the software. |
| **LR5** | **Results & Plots** | Figures/ graphs / tables are not developed or are poorly constructed with erroneous results. Titles, captions, units are not mentioned. Data is presented in an obscure manner. | Figures, graphs and tables are drawn but contain errors. Titles, captions, units are not accurate. Data presentation is not too clear. | All figures, graphs, tables are correctly drawn but contain minor errors or some of the details are missing. | Figures / graphs / tables are correctly drawn and appropriate titles/captions and proper units are mentioned. Data presentation is systematic. |
| **LR9** | **Report** | All the in-lab tasks are not included in report and / or the report is submitted too late. | Most of the tasks are included in report but are not well explained. All the necessary figures / plots are not included. Report is submitted after due date. | Good summary of most the in-lab tasks is included in report. The work is supported by figures and plots with explanations. The report is submitted timely. | Detailed summary of the in-lab tasks is provided. All tasks are included and explained well. Data is presented clearly including all the necessary figures, plots and tables. |