DIGISM-PS1

Team Name: Silicon Sages
Team Members: Ayushi Kumari (22095020)
Allu.Deekshita(22095009)

ENCRYPTION AND DECRYPTION USING COMBINATIONAL CIRCUITS

Components Used:

- Quad 2:1 mux
- Logic gates
- wires

Cost:

- Encryption - 10.8rs/-
- Encryption + Decryption- 90.8rs/-

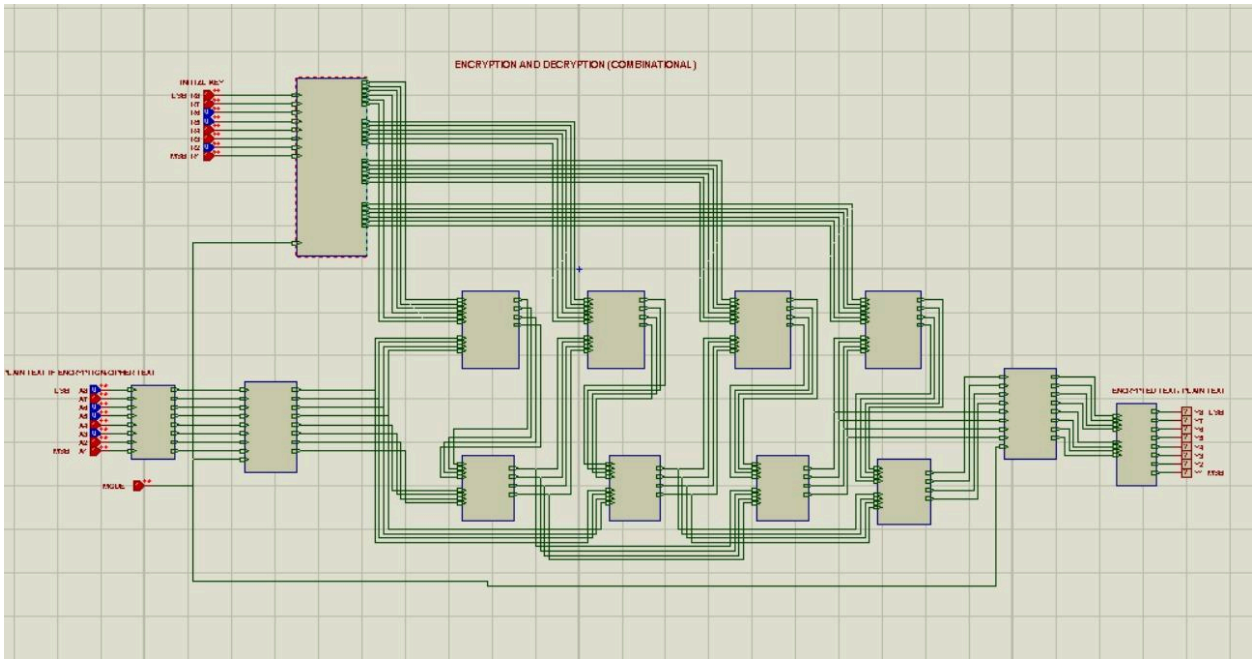
PROBLEM STATEMENT

To design and simulate a digital circuit that encrypts a given digital input using a simplified version of the DES algorithm.

DES ALGORITHM:

Data Encryption Standard (DES) is a block cipher with a 56-bit key length that has played a significant role in data security.Here in our PS, we have to deal with just
8 bit binary input.

CIRCUIT IMPLEMENTATION ON PROTEUS:



APPROACH:

**ENCRYPTION **

=> ROUND KEY GENERATION:

- The given initial 8 bit key is sent into a round key generator which generates four outputs with each 6 bit. We take those outputs as our round keys. into two parts Left part (4 bits) and Right part(4 bits).Now both the parts were sent into 1 bit Left Shift block which performs the left shift of four bit binary number. It is done just by exchanging wires .

  For example if the input is L3L2L1L0 then the output of left shift block is

  L2L1L0L3.

- The 8 bit number resulted from combining both left and right parts obtained above is compressed into 6 bit number as given in ps by exchanging wires . The 6 bit number is our ROUNDKEY1.
- Now the left and right parts after 1 bit shifting were sent into 2 bit Left Shift block which performs 2 bit shift and the results were again compressed as mentioned above which results in ROUND KEY2.
- Similarly ROUND KEY 3 is generated by 1 bit shift left shift.And ROUND KEY 4 is generated by 2 bit left shift
- Hence all the round keys were generated.

=> FUNCTION GENERATION:

- A block is generated for this which taken two inputs 6 bit round key and 4 bit binary number
- The 4 bit number is expanded into 6 bit by using the approach given.
- Now the two 6 bit numbers were given to a XOR gate. The output is divided into two 3 bit numbers which were given as input two S BOXES created.
- Left 3 bits were given to S BOX1 which outputs a 2 bit number. The structure of S BOX1 is created by solving kmap which is made according to the table given in ps and writing logic expressions eventually building circuits with corresponding logic gates
- Right most 3 bits were sent into S BOX2 which outputs 2 bit number. construction of SBOX2 is made similarly as above according to the information given in ps for SBOX2.

- The output of two boxes were merged and sent into a permutation box as input which functions according to the logics given by exchanging the wires.
- The result came after permutation is the 4 bit output of the function.

=> MAIN PROCESS:

- The given 8 bit plain text is initially sent into a permutation box which functions according to the logic given in ps.
- The output of permutation box (8 bits) is divided into two parts each of 4 bits.
- The right most 4 bits is given as input to the function generator along with ROUND KEY1. The output of function box is given as input to XOR gate along with left most 4 bits.
- The output of XOR gate becomes right most part for next round And the right most part of previous round becomes becomes left most part for next round.
- The same process will repeat for 4 rounds with . The each generated round keys were given as input in each round.
- The final outputs left part and right parts were merged and inverse permutated to get the final Encrypted message of 8 bits

** DECRYPTION **

- For decryption, we will use a mode control, mode 0 is encryption and mode 1 is decryption.
- we have made an IC with 6 2x1 mux. when connected between ist and 4th round keys generated.it will produce 1st round key when mode is 0 and 4th output when mode is 1. same is done for 2nd and 3rd round keys. a total of 4 such ICs are used for

each round key. so that when mode is 1, the order of round is reversed. Along with it, we have introduced an ic (EN/DE). It uses mux to swap the left and right half part when the mode is 1 and passes the output unchanged in mode is 0.

- This way when mode is 1, input encrypted text is converted to plain text and when mode is 0, input plain text is encrypted.