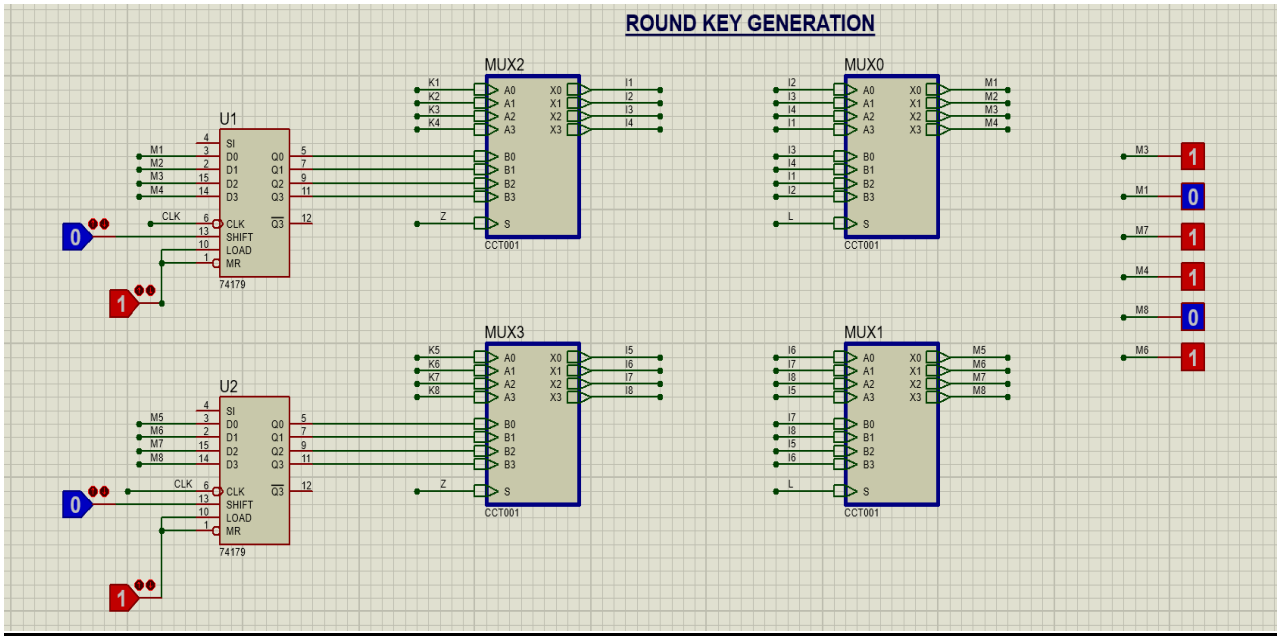
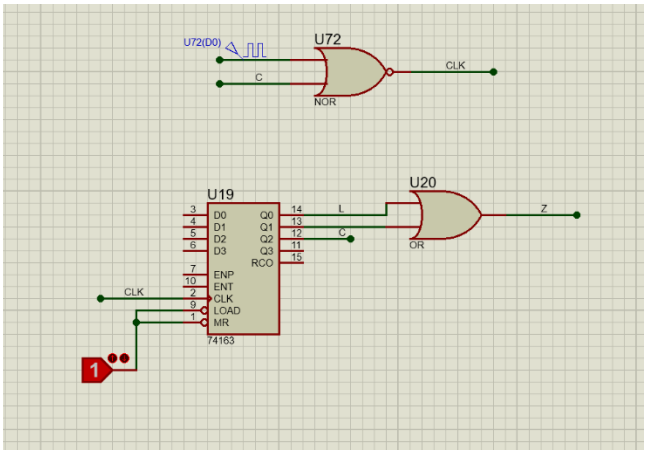
Team Spartans
Ayush Kumar
Ashish Singh

DIGISIM PROBLEM STATEMENT- 1

A) SEQUENTIAL APPROACH

ROUND KEY GENERATION



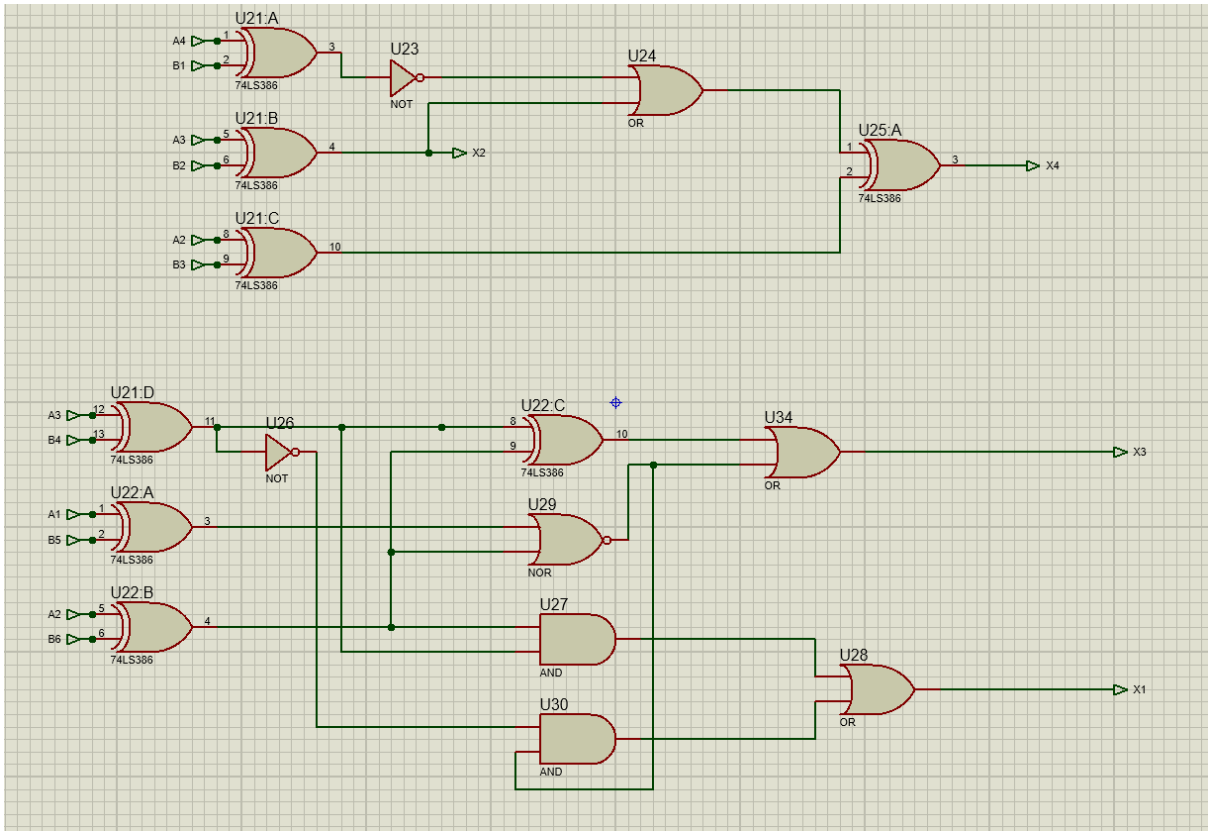


To generate round key using sequential approach, we have used a combination of a clock, multiplexers and shift registers. There are 4 round keys to be generated so we used a counter to count from 0-3 and we stopped the clock when the 2rd bit of the counter becomes 1. We did this using a nor gate with 2rd bit of counter and a clock as inputs and generated a new clock which stopped after 4 cycles.

Mux 0 and Mux 2 are used for the left 4 bits operationand Mux 1 and Mux 3 are used for right 4 bits. Mux 2 and 3 have a select line Z which is 0 when the counter is at 0 and 1 otherwise, achieved using taking OR of bit 0 and 1 of counter. So initially these 2 muxes pass on the given key K_0,K_1.....K_8 and then the second input. Now the output of these muxes go to Mux 0 and 1 respectively where circular left shift operation takes place. Here we need to shift left by 1 when the counter is at 0 and 2, and shift left by 2 when the counter is at 1 and 3. So the select line of these muxes is L which is the 0th bit of the counter. So the first input of the muxes are 1 time left shifted output of mux 2 and 3 and the second inputs are 2 time left shifted outputs of the same.

Now the outputs of these muxes are merged as given in ps and also given as inputs for the next cycle of round key generation through shift registers set in parallel load mode. In the next cycle the output of these registers are the second inputs of the muxes 2 and 3 and now since Z is 1 these inputs proceed further in the chain. This is how the round keys are generated in each cycle.

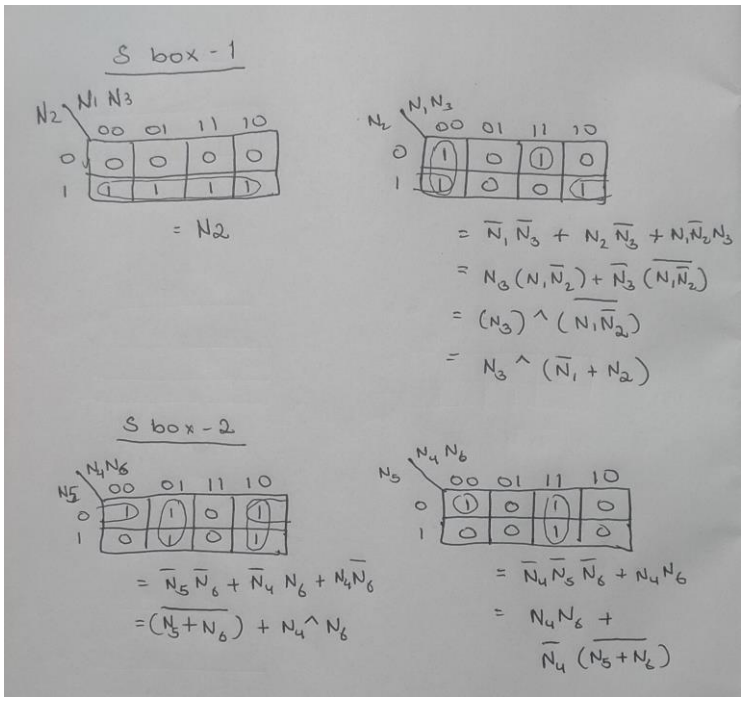
FUNCTION F



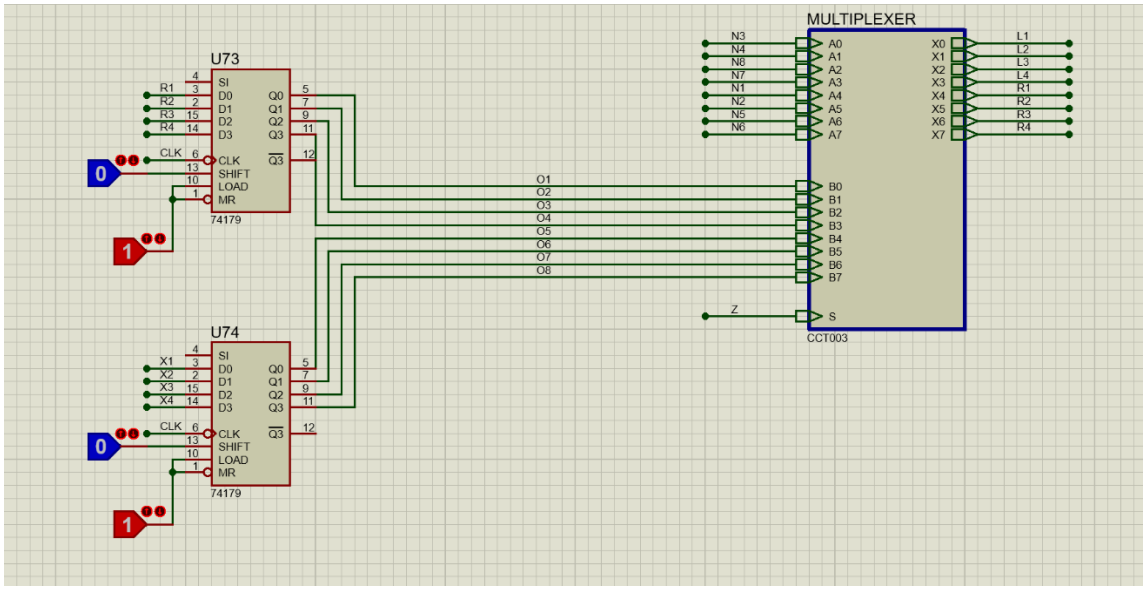
In the function F we took the 4 right bits and the round key are taken as inputs. First the right 4 bits are expanded as given and then xored with the round key for the given round. Next the S- boxes for the left and right 3 bits are solved using K- maps to obtain the following expressions which are merged and permuted: -

a) $X_1 = (N_1' + N_2)\ \hat{}\ N_3$
b) $X_2 = N_2$
c) $X_3 = (N_4 \hat{} N_6) + (N_5 + N_6)'$
d) $X_4 = N_4.N_6 + N_4'.(N_5 + N_6)'$

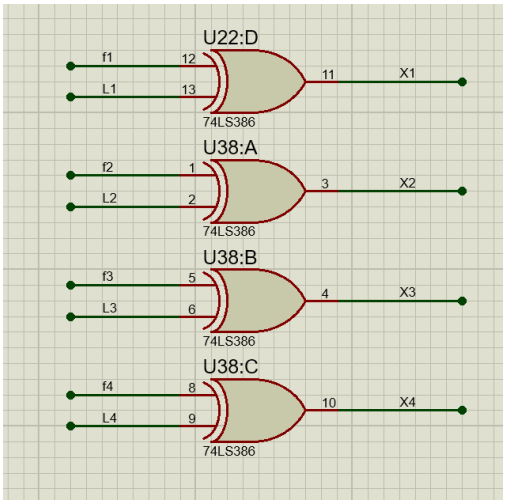
(N1, N2, N3, N4, N5 and N6 are the 6 xored outputs of right 4 bits and the round key.)



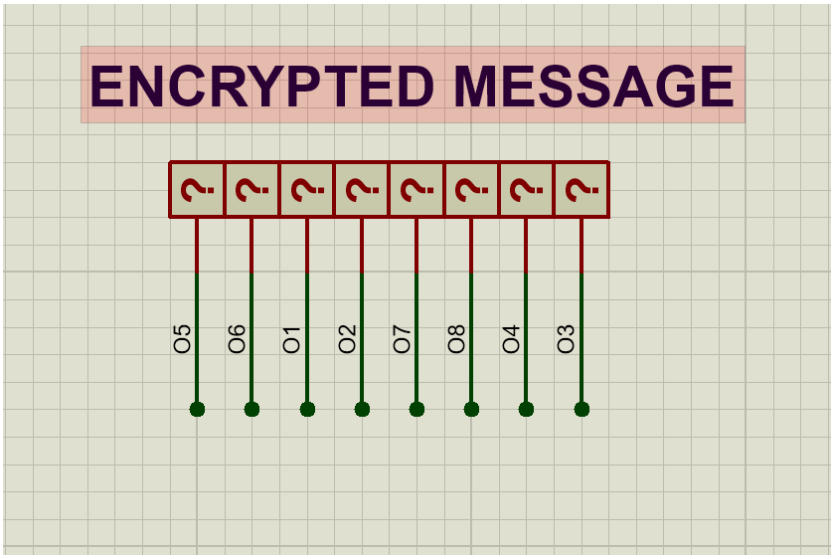
LEFT AND RIGHT BITS GENERATION



This is the part of solution where the left and right bits are updated for each round. The select line to this multiplexer is also Z, 0 initially and 1 otherwise. So initially the given plain text is passed on after applying given permutation. While from next cycle the left 4 bits of output are made to be the right four bits of previous round and the right four bits are the xored output of the previous left bits and the output of function F. This process is done by storing these values in a shift register in parallel load mode, so when next clock edge comes then these values are updated as required.



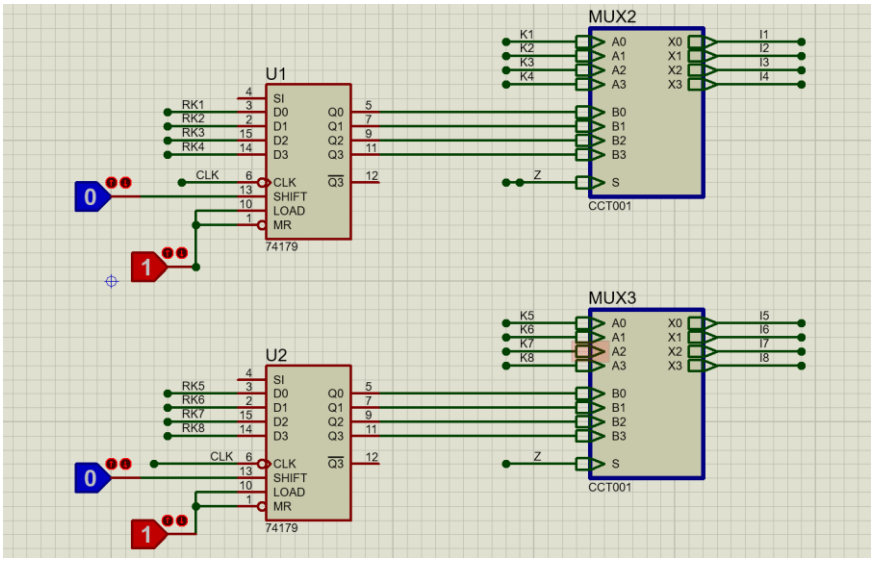
These 8 bits (the second input to this multiplexer) are also the final encrypted output at the end of 4th round and are then inverse permuted and displayed through logic probes.

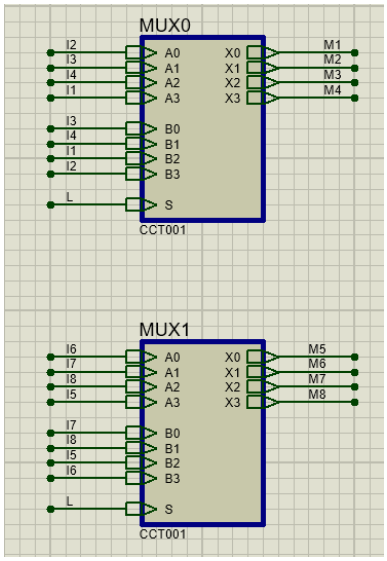


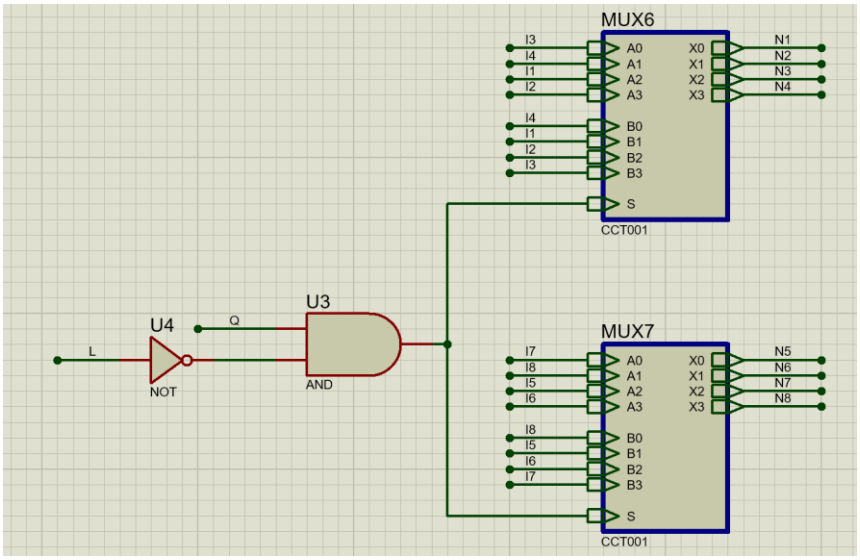
DECRYPTION: -

ROUND KEY GENERATION

For this part round keys have to be generated for both encryption and decryption part. Round key generation for encryption part is the same as before. For the decryption part a few extra multiplexers have been added. The initial part of this process is same as in previous solution where muxes 2 and 3 initially forward the given Key and later on they forward the values of the register.

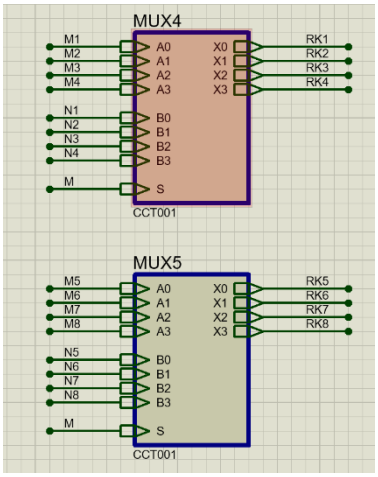




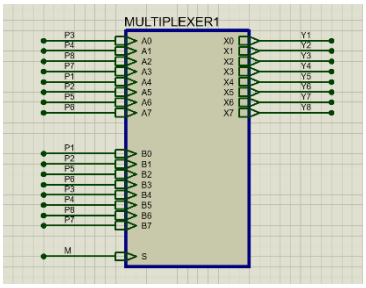


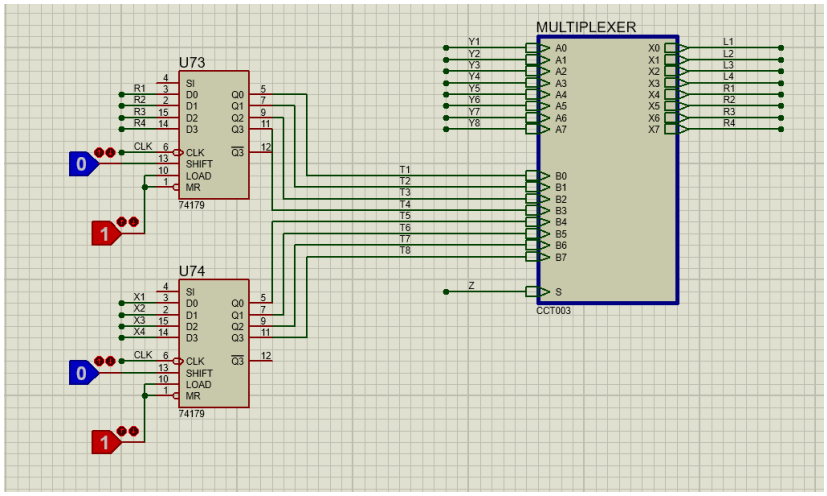
Now mux 0 and 1 pr=erform the left shift operation for the encryption part similar to what was done before. Mux 6 and 7 are for shifting operations of decryption part where the input is right shifted by 2 bits for 0^{th}, 1^{st} and 3^{rd} cycle and is shifted once for 2^{nd} cycle of round key generation. For the select line of these muxes we make select line 1 when the counter output is 2 using the 0^{th} and 1^{st} bit of counter.

Mux 4 and 5 give out the final round key based on the selected mode M, 0 for encryption and 1 for decryption. The output of these are the keys for each round and are also fed as inputs to the shift registers at the start of module.



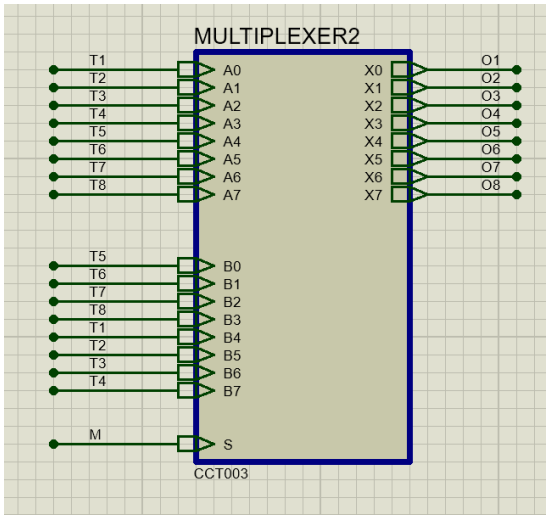
First step of decryption process is to swap the left and right 4 bits. This task has been done through multiplexer 1.





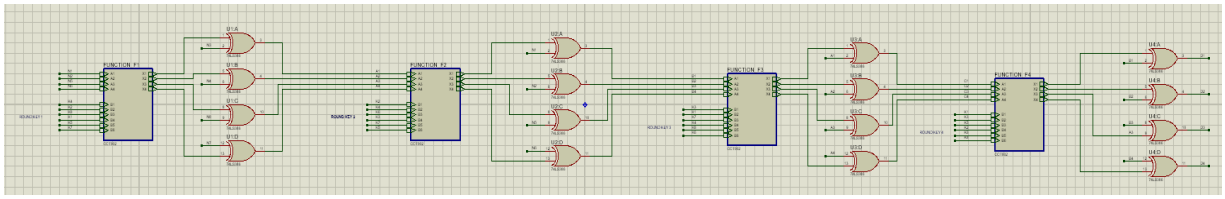
Now this multiplexer is used to initailly pass on the given plain text (after permutation) and later to pass on the generated values further down the chain, same as done previously.

Finally through multiplexer2 we again swap the 4 left and right bits and do inverse permutation to get the final decrypted message.



B) COMBINATIONAL APPROACH

In this approach, the round keys have been generated manually by shifting and are directly fed as inputs to the corresponding Function F block, whose output is then xored with the corresponding left 4 bits. Then such blocks are connected consecutively to perform operations of successive rounds. At the end of 4th round the wires are labelled to perform inverse permutation task.



DECRYPTION: -

The decryption process using combinational approach has the same changes as sequential approach did with 2 multiplexers in place to swap the 4 left and right bits, once at the start and once at the end. Also at the input of each function F there is a multiplexer in place to send the respective round keys for encryption and decryption processes, decided by it's select line M (Mode).