Project Report

DIGISIM PS-1



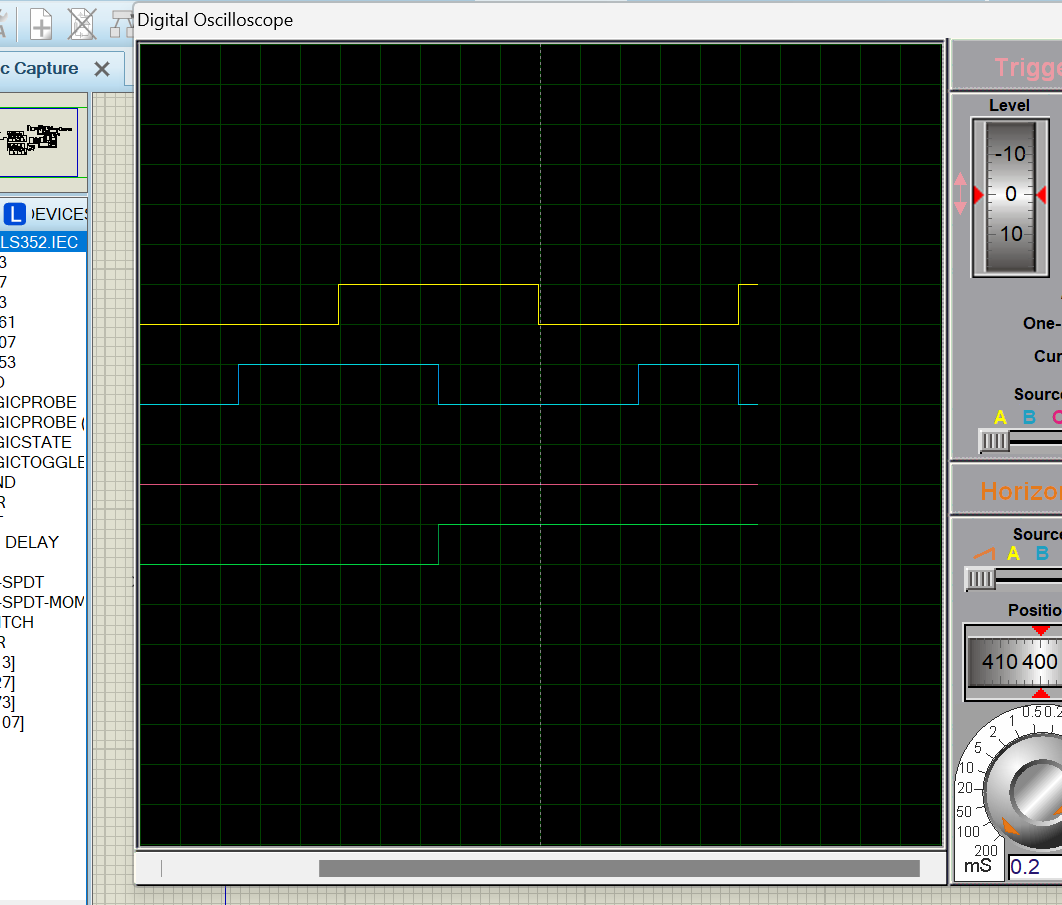
**Hrishabh Deshmukh**

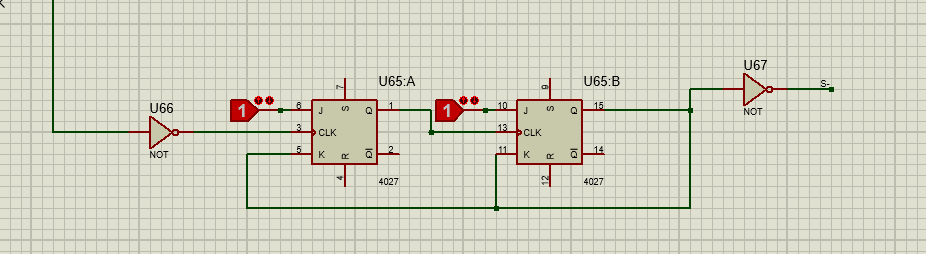
**Sammit Dhar**

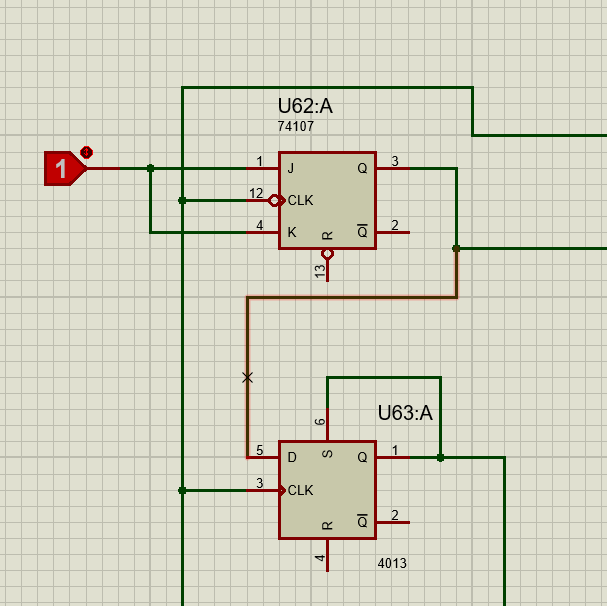
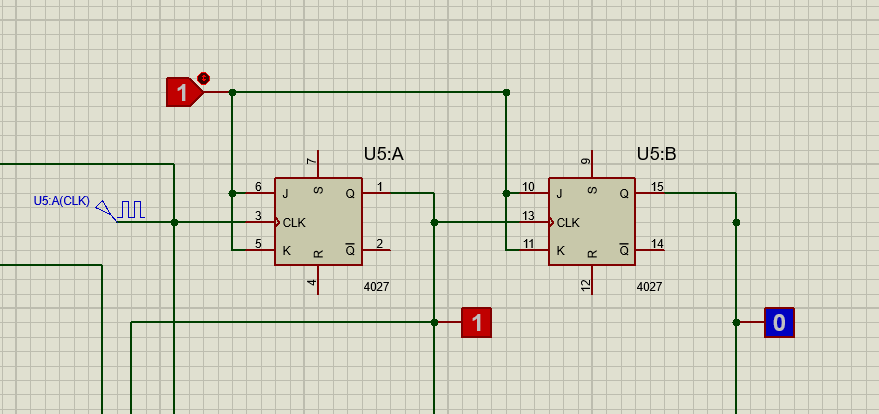
ECE Part 2

# SEQUENTIAL ENCRYPTION (Rs. 75.8)

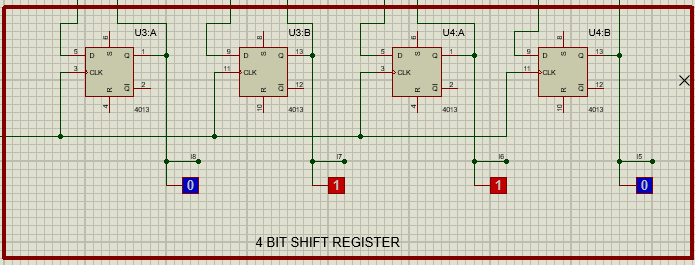
For the Sequential implementation, we had to employ different waveforms obtained from a single clock, for which we used different combinations of Flip-Flops. These waveforms were synchronized in order to begin the required processes in time before the start of the next round.



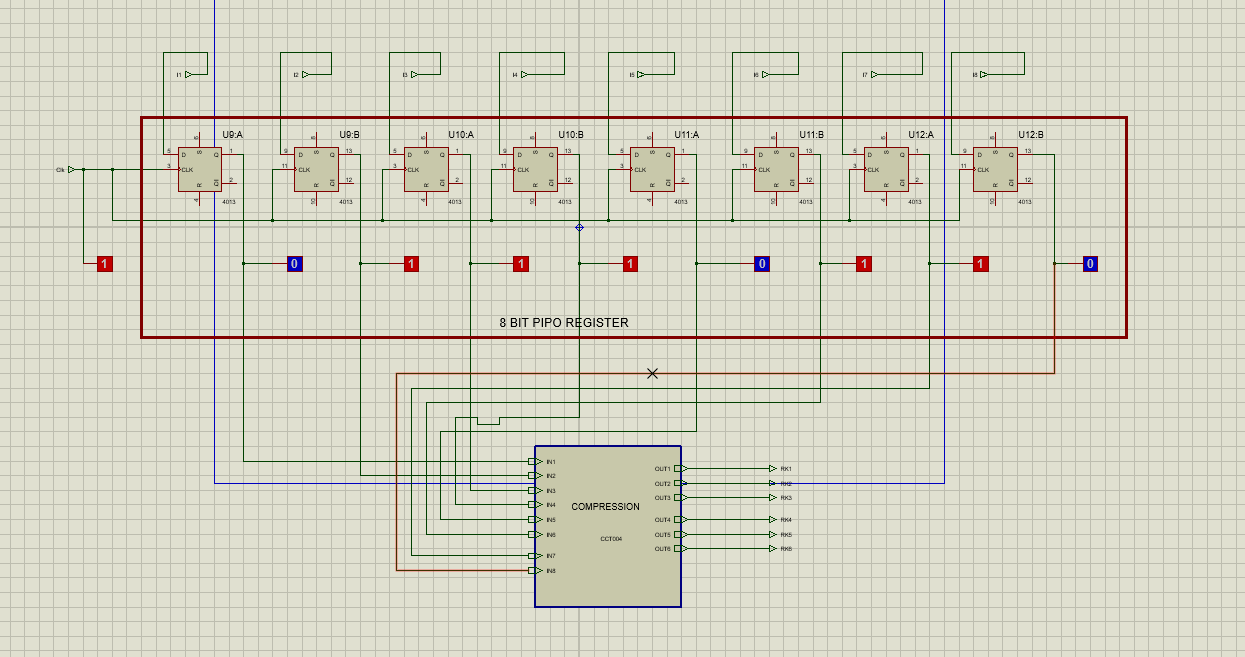


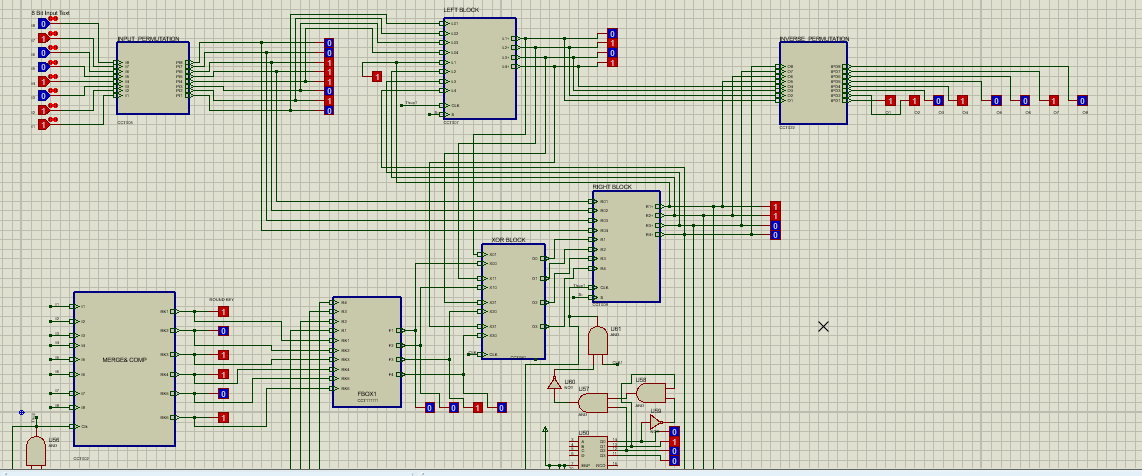
For Round Key Generation, we used the waveform with non-uniform frequency (shown in blue). For shifting, we employed simple Shift Registers. The shift in each round was decided by the waveform given.



In the Merge and Compress box, we used a uniform waveform, such that its output showed the Round Key corresponding to each Round.



We also employed a counter to stop the generation of Round Keys beyond the 4th round. It is however flexible and can be extended to more rounds.

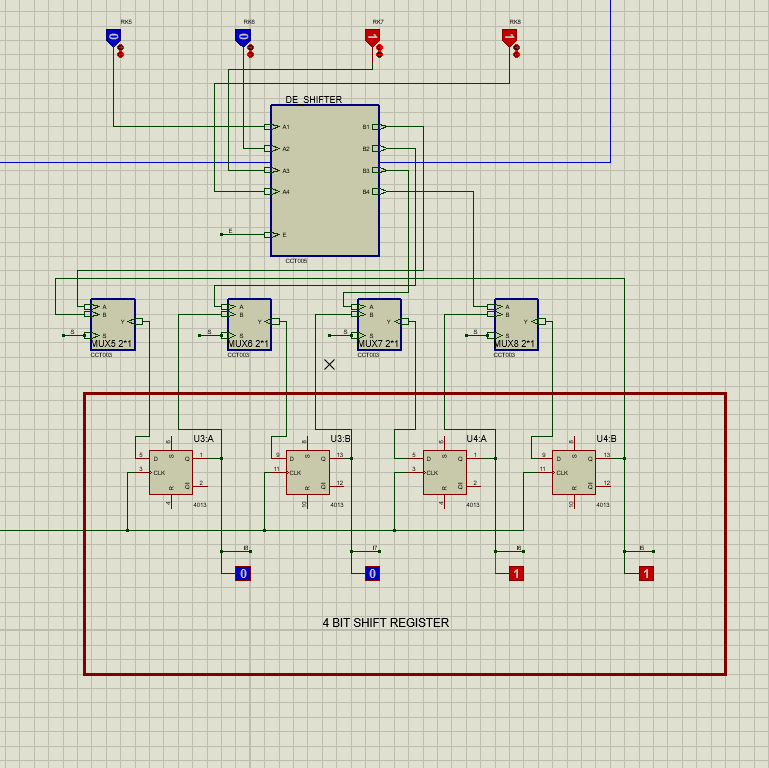
For the shuffling of the input, we used a separate subcircuit. 

We divided each round into Left and Right Blocks.

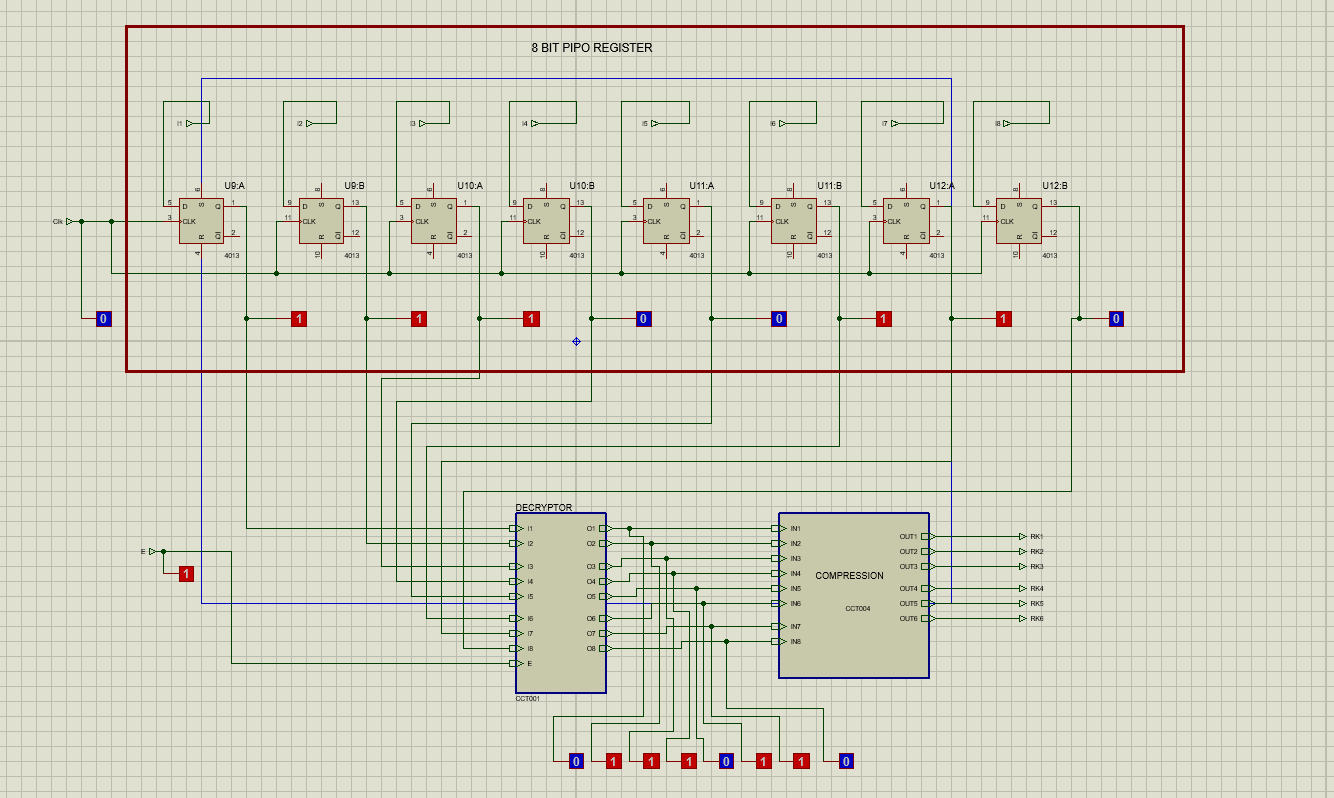
In the Left Block, we employed a 4 bit PIPO register (for processing) and MUXes (to take input). The Right Block was constructed in a similar fashion. The XOR block also used a PIPO register to do the required calculations at the exact time they are required.

For the Inverse Permutation of the output, we again employed a separate subcircuit.

# SEQUENTIAL DECRYPTION (Rs. 89.8)

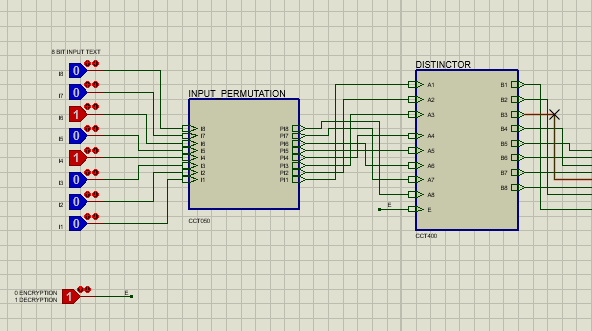
Here, we added a De-Shifter block in the Round Key Generation circuit, as this time we had to do right shifts instead. This was achieved by employing MUXes inside. 

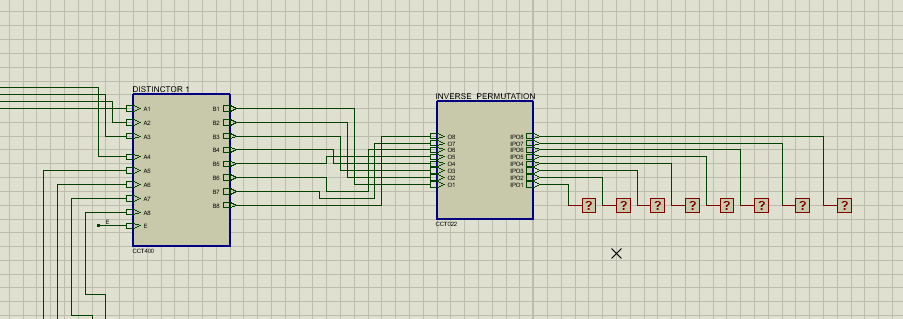
The purpose of the Decryptor block inside the Merge & Compress block is to invert half of the input in order to introduce ease of logic formation for encryption.



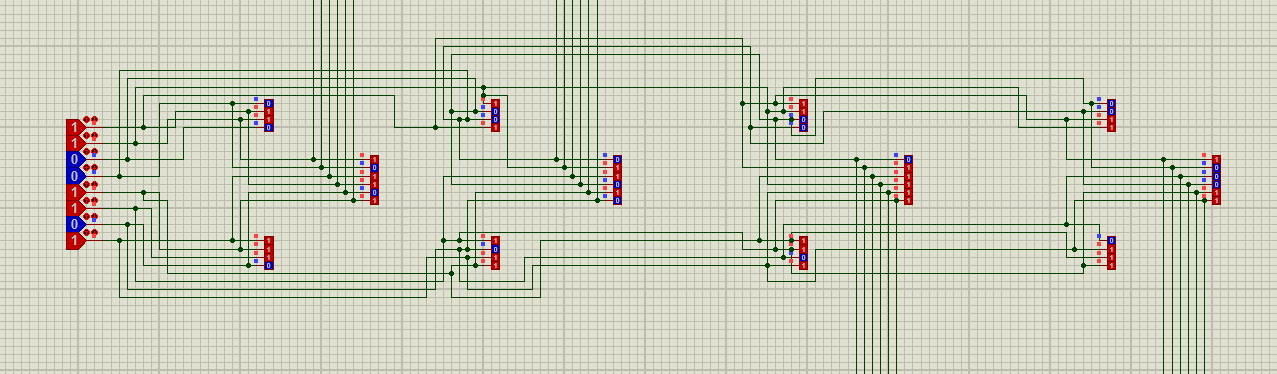
The purpose of these blocks is to generate the desired Round Keys in the opposite order of Encryption.

Moving on to the Plain Text Processing Circuit, we had to add 2 Distinctor blocks again in order to swap the left and right halves of the input if Decryption was needed.



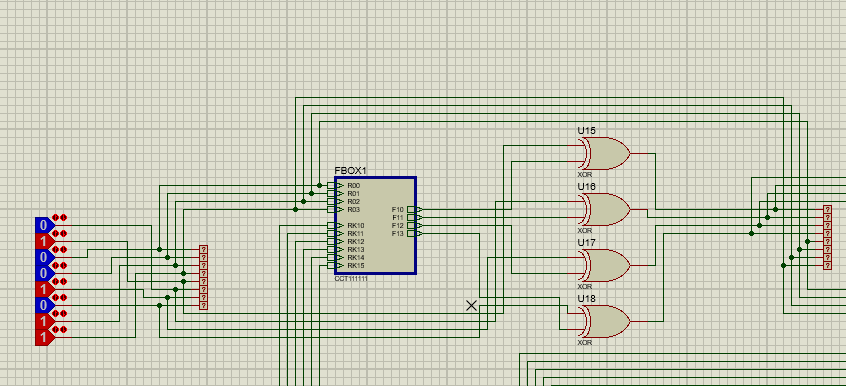


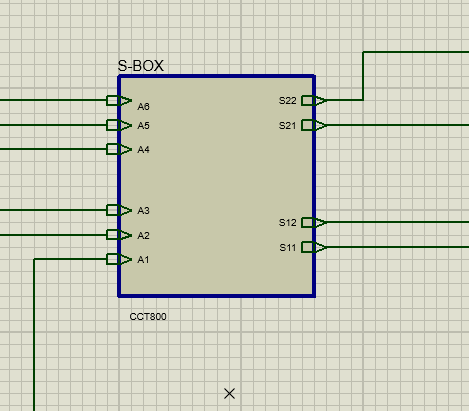
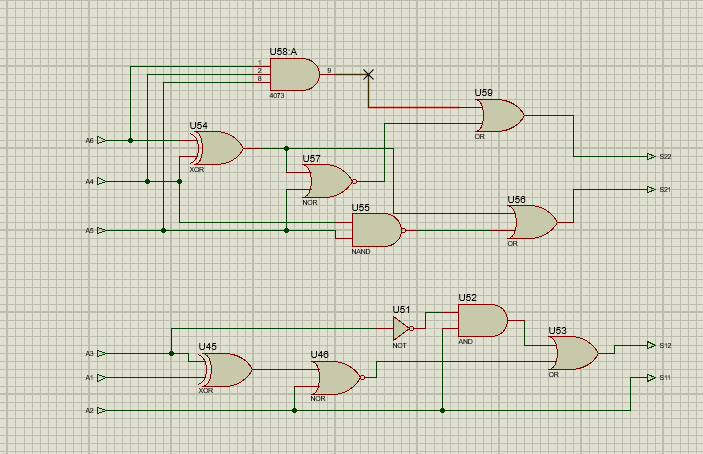
# COMBINATIONAL ENCRYPTION (Rs. 8.8)

For Round Key Generation, we employed the shuffling of wires. 

Left Shift and Right Shift was done accordingly.

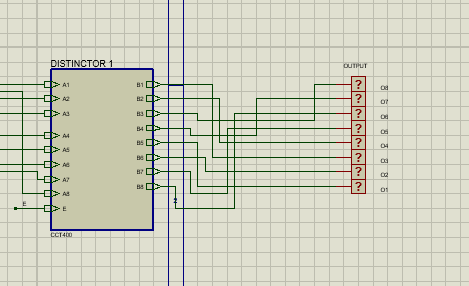
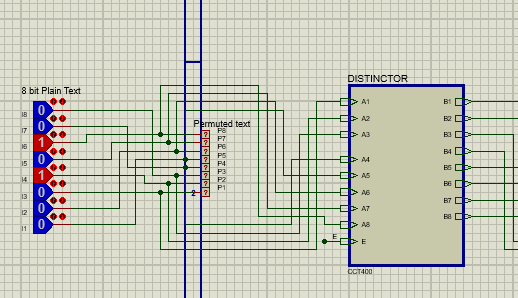
For each round, we made a subcircuit, for the F-box and XOR-ed each bit of the output with L(n-1), n being the round number. We solved the S-boxes using K-maps and applied the relevant gates, employing a single subcircuit for the same.

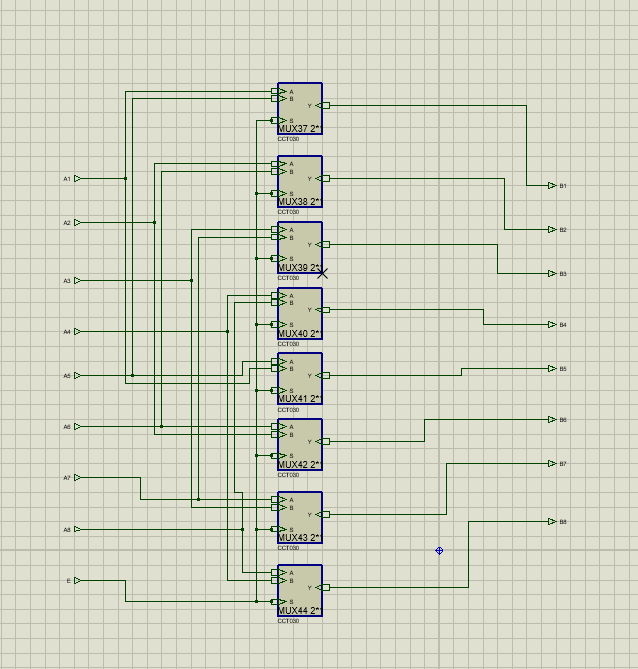


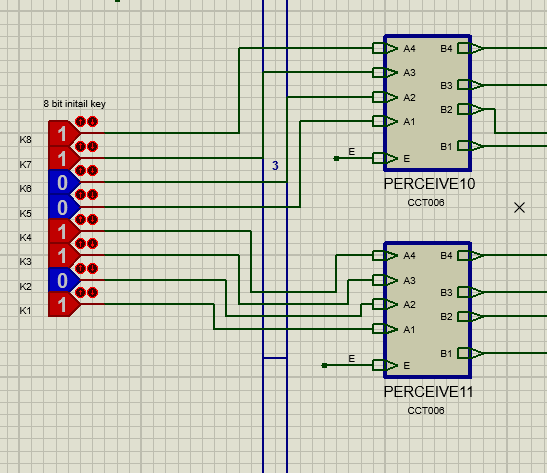
# COMBINATIONAL DECRYPTION (Additional Rs. 16 after Encryption)

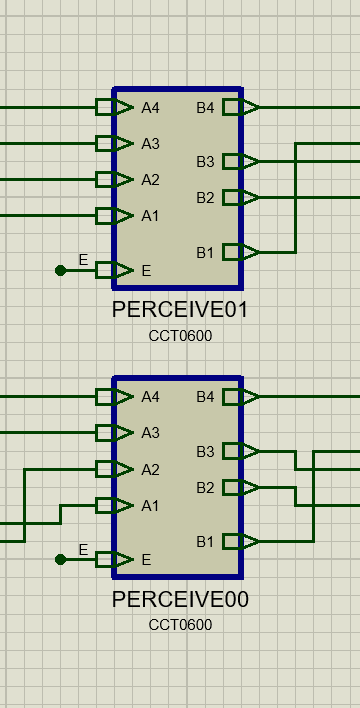
We discovered that the processes of encryption and decryption are mostly the same, with decryption having a few more steps. For that, we employed 2 Distinctor boxes, which swapped the left and right halves of the input if Decryption was needed.





In the Round Key Generation circuit, we had to employ 2 subcircuits name Perceive0 and Perceive1 in order to generate the Round Keys in the opposite order.

PERCEIVE1 

PERCEIVE0 

# 