DIGISIM PS-1

Team Name: The Hawks

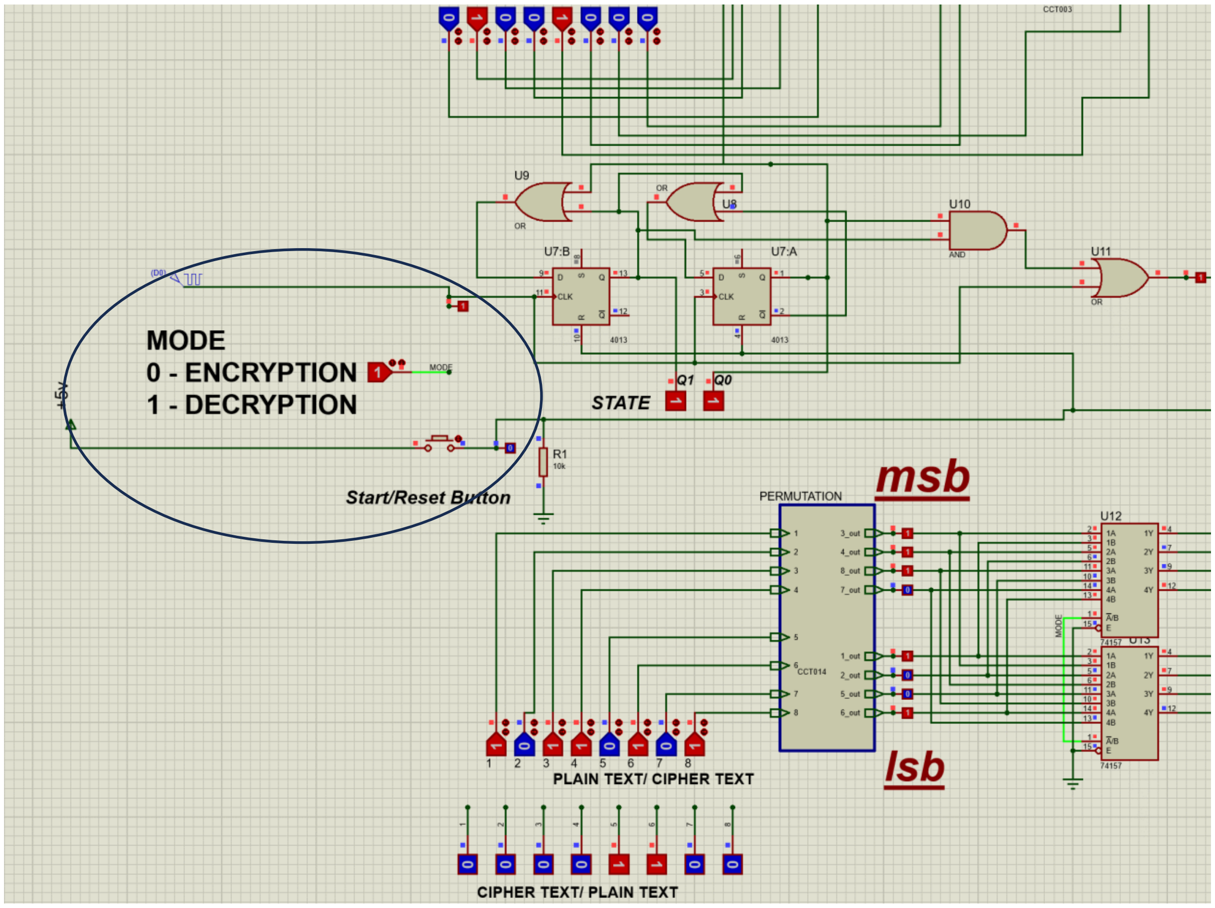
Sequential Circuit:

Part 2(Encryption/Decryption): **Total Cost = 68.5+ 49.2(6- MUX 8x4) = 117.7**

Please read combinational part 1, part 2 and sequential part 1 pdf before reading this.

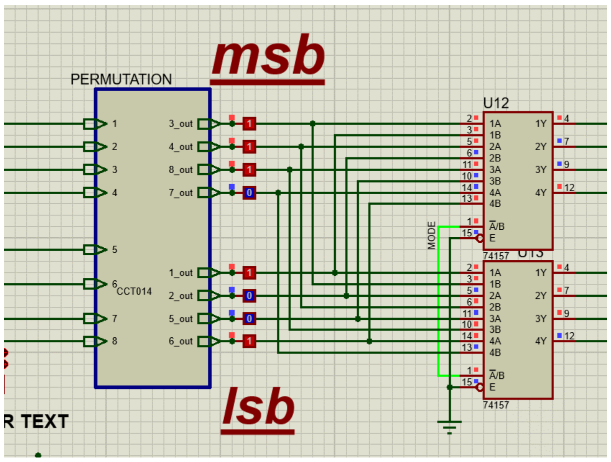
Everything is same as of Part 1 except the following things: -

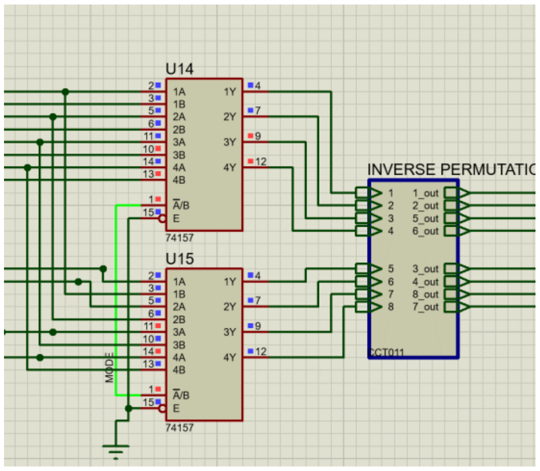
1. **INPUT METHOD:**



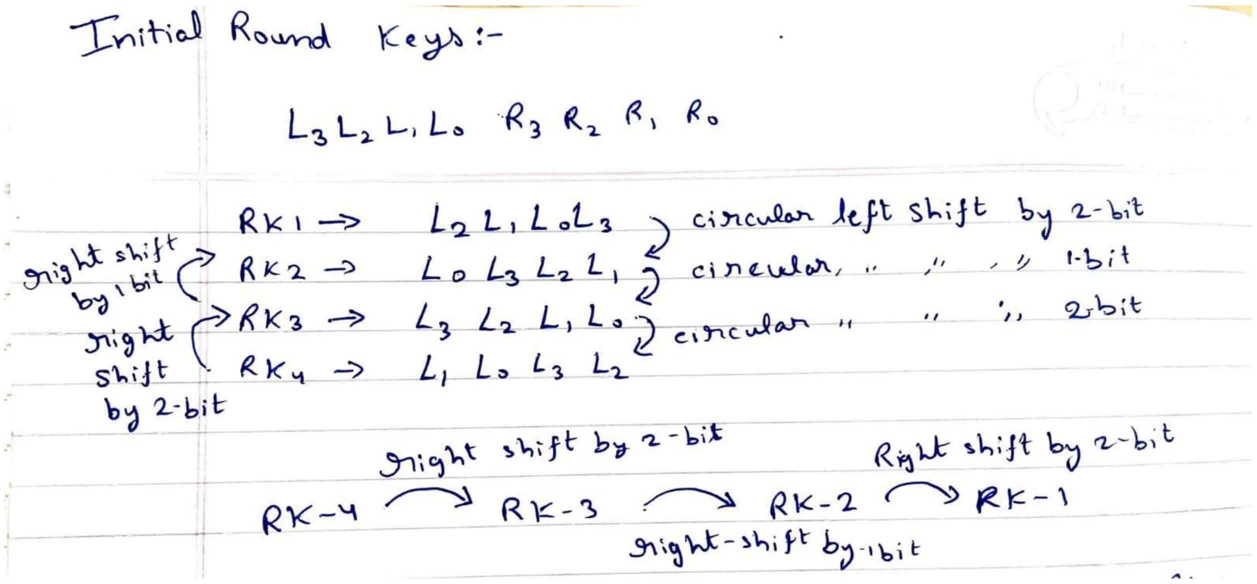
After you start the simulation, enter the plain text/cipher text, initial key as well as MODE and then trigger the start button once. After the state 11 is reached through FSM, the cipher text/plain text is generated. On triggering the PUSH button, the state is set to 00.

2. **MUX(8x4) used after permutation and before initial permutation.**
(reason is combinational part 2 in previous pdf)





3. **Round Key Generation: -**
To generate round keys in reverse order using same circular left shift flip flops, few modification is done. First we need to observe the following (only consider left 4 bit of initial key. It will be same for the right keys)



Initial Round Keys :-

$L_3 L_2 L_1 L_0$ $R_3 R_2 R_1 R_0$

RK1 → $L_2 L_1 L_0 L_3$ circular left shift by 2-bit

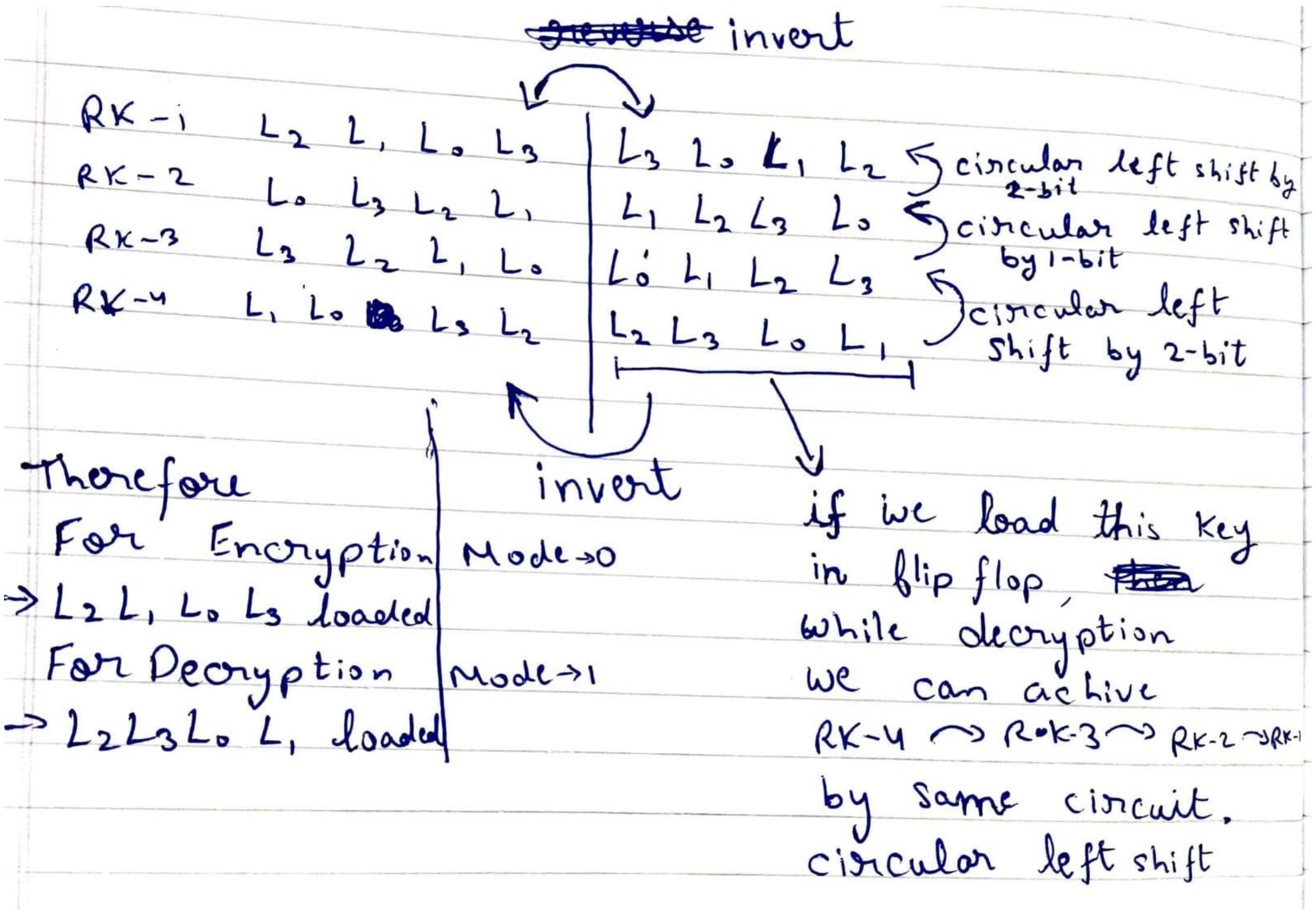
right shift by 1 bit — RK2 → $L_0 L_3 L_2 L_1$ circular " " " 1-bit

right shift by 2-bit — RK3 → $L_3 L_2 L_1 L_0$ circular " " " 2-bit

RK4 → $L_1 L_0 L_3 L_2$

$$\begin{tikzcd} RK\text{-}4 \arrow[r, "\text{right shift by 2-bit}"] & RK\text{-}3 \arrow[r, "\text{right-shift by 1-bit}"'] & RK\text{-}2 \arrow[r, "\text{Right shift by 2-bit}"] & RK\text{-}1 \end{tikzcd}$$

For Encryption Mode, **Rk-1** is loaded in the left 4 flip flops using **set and reset of flip flops**.
For Decryption Mode, **Rk-4** is loaded in the left 4 flip flops using **set and reset of flip flops.**
And we need to do circular right shift circuit in decryption mode. But if we invert the keys and then load it, we can generate the keys using circular left shift circuit.



~~reverse~~ invert

RK-1	$L_2\ L_1\ L_0\ L_3$	$L_3\ L_0\ L_1\ L_2$
RK-2	$L_0\ L_3\ L_2\ L_1$	$L_1\ L_2\ L_3\ L_0$
RK-3	$L_3\ L_2\ L_1\ L_0$	$L_0\ L_1\ L_2\ L_3$
RK-4	$L_1\ L_0\ L_3\ L_2$	$L_2\ L_3\ L_0\ L_1$

circular left shift by 2-bit

circular left shift by 1-bit

circular left shift by 2-bit

invert

Therefore

For Encryption Mode→0

→ $L_2 L_1 L_0 L_3$ loaded

For Decryption Mode→1

→ $L_2 L_3 L_0 L_1$ loaded

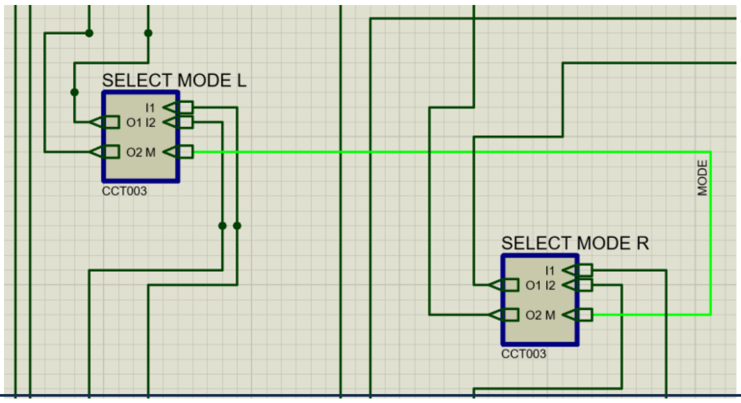
if we load this key in flip flop, ~~then~~ while decryption we can achive RK-4 → RK-3 → RK-2 → RK-1 by same circuit. circular left shift

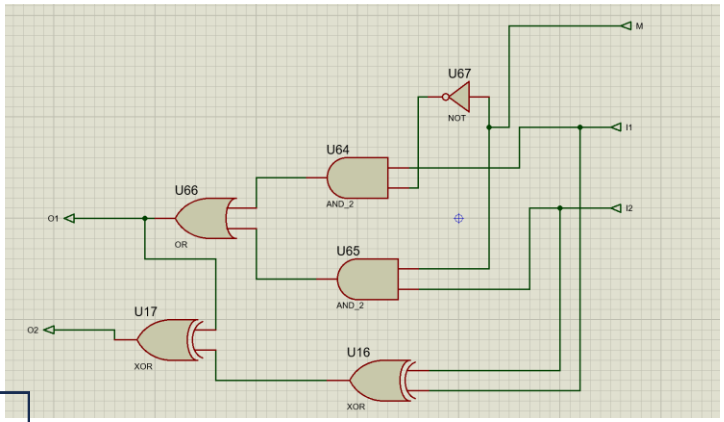
Therefore For Decryption, circular left shift circuit is used. (explanation of circular left shift circuit is given is sequential part-1 pdf**)**

When Mode = 0 ---- L2 L1 L0 L3 is loaded

When Mode = 1 ---- L2 L3 L0 L1 is loaded

Therefore only L1 , L3 are flipped. We have done this using following circuit.





When M=0 --- O2= L1 and O1= L3

When M=1--- O2= L3 and O1= L1

I1 = L3

I2 = L1

We have done some Mathematical juggling to minimize gate usage

$$\boxed{O_1 = \bar{M} I_1 + M I_2}$$

		output
When	Mode = 0	$O_1 = I_1$
	Mode = 1	$O_1 = I_2$

$$\boxed{O_2 = \bar{M} I_2 + M I_1}$$

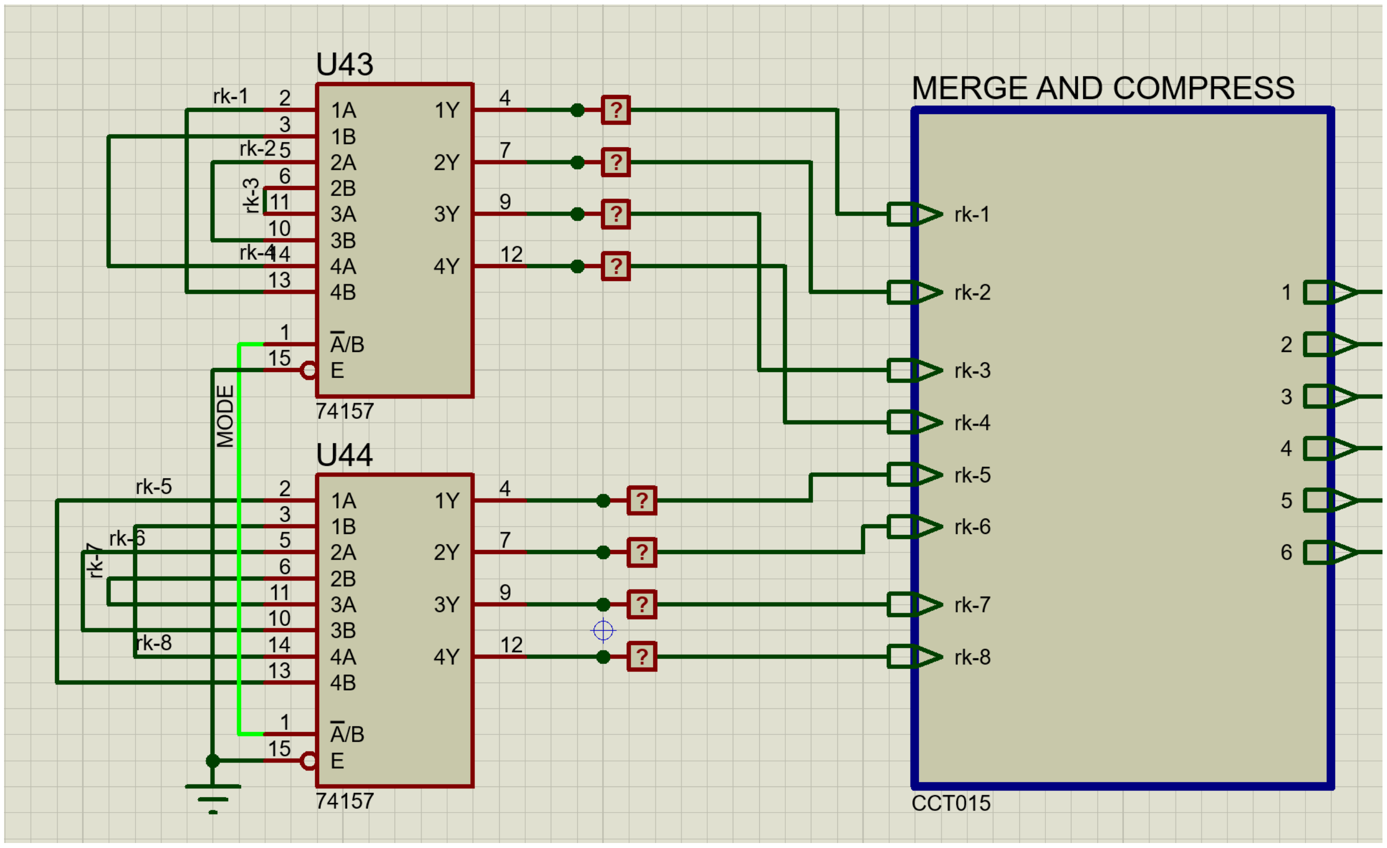
		output
When	Mode = 0	$O_2 = I_2$
	Mode = 1	$O_2 = I_1$

We have done some juggling to minimize gate usage.

i.e. $O_1 \oplus O_2 = I_1 \oplus I_2 \rightarrow$ observed.

$\therefore$ $\underline{O_1 \oplus I_1 \oplus I_2 = O_2}$

4. **Before Merge and Compress box:**



As the keys were inverted while decryption,

MUX are used to invert the round keys when Mode is 1 .

Rest Everything is same as of sequential part-1