Mini Project - Digital System Design EE311

Roll Number - 16EE230

Name - M NAVEEN KUMAR

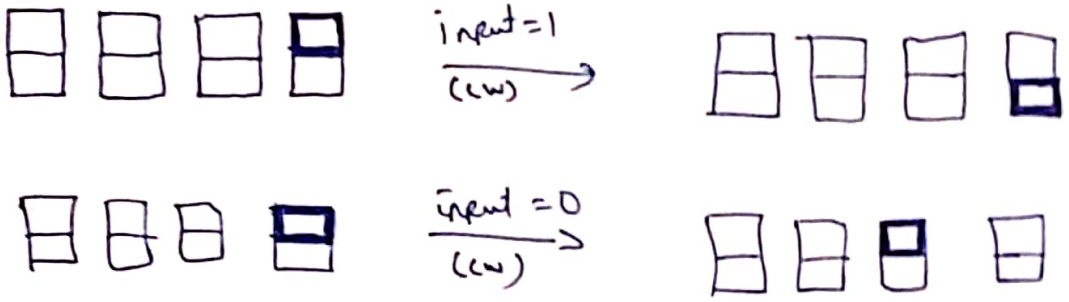
Question # 8

---

Objective :-

Given a 7-segment LED display (4 in number), circulate a square pattern based on the input.

eg :- Thick edges indicate 'ON'.




---

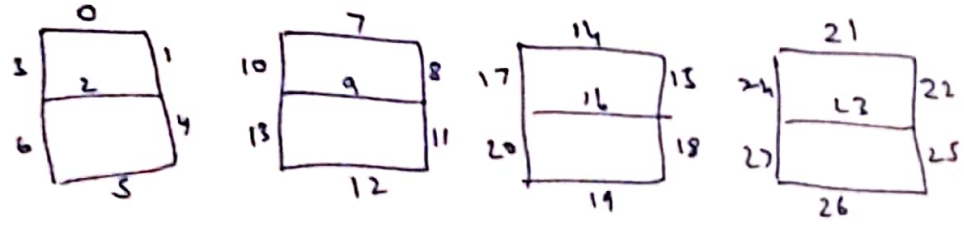
Solution :-

Input : cw (001) indicating if clockwise or not

Number of States = 8 (Can be observed from the diagram)

Output :-

Since we have a 7-segment led display and 4 such displays, number of outputs = 28. Let us number them below :-



But there are only 8 possible outputs and we can clearly eliminate some states based on observation that the output is always a square :-

Therefore, Let us rename the states

$$z_0 = z_1 = z_3 = z_{000}$$

$$z_4 = z_5 = z_6 = z_{001}$$

$$z_2 = z_{00}$$

$$z_7 = z_8 = z_{10} = z_{010}$$

$$z_{11} = z_{12} = z_{13} = z_{011}$$

$$z_9 = z_{01}$$

$$z_{14} = z_{15} = z_{17} = z_{100}$$

$$z_{18} = z_{19} = z_{20} = z_{101}$$

$$z_{16} = z_{10}$$

$$z_{21} = z_{22} = z_{24} = z_{110}$$

$$z_{25} = z_{26} = z_{27} = z_{111}$$

$$z_{23} = z_{11}$$

9 outputs

$$Z = z_{000}\, z_{00}\, z_{001}\, z_{010}\, z_{01}\, z_{011}\, z_{100}\, z_{10}\, z_{101}\, z_{110}\, z_{11}\, z_{111}$$

In short, $z_{xy}$ = middle segment of panel $(xy)_2$

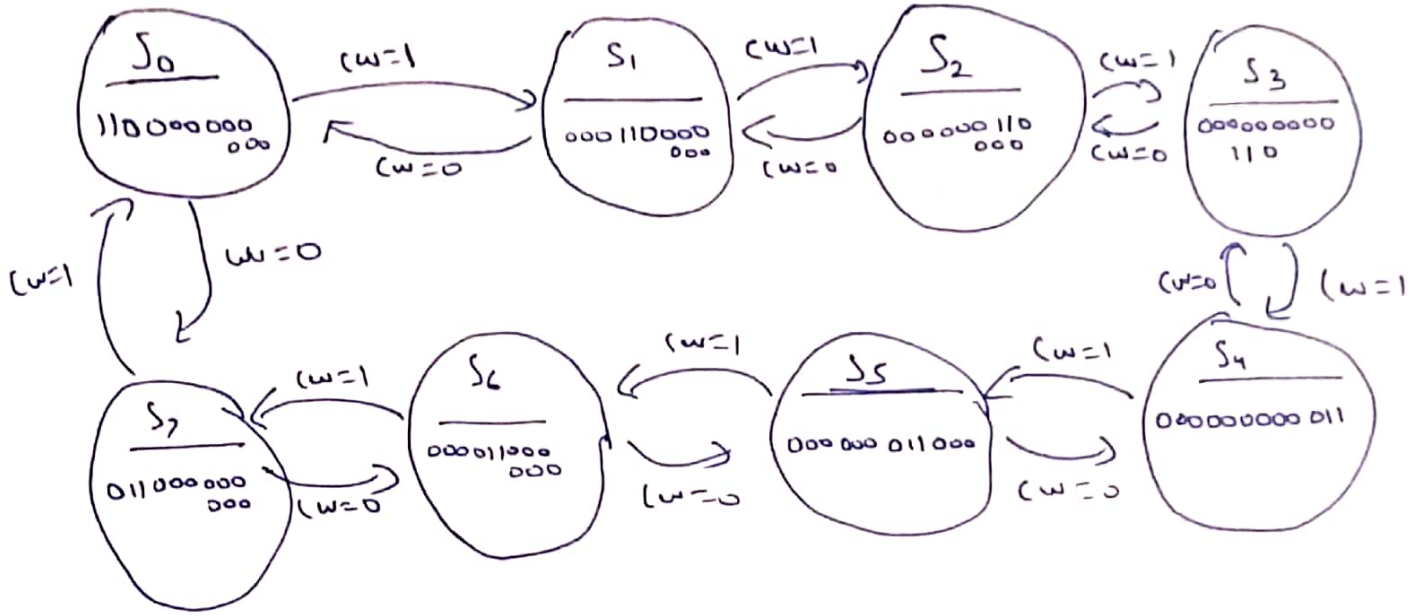
$z_{xy8}$ = Upper 3 segments of panel $(xy)_2$

$z_{xy1}$ = Lower 3 segments of panel $(xy)_2$

This is based on the fact that, to form a square all 3 of any upper or lower segment + the middle segment must be lit.

Let's create the state diagram :-

It is a Moore Machine as the output only depends on the present state.




Moore State Diagram

State table (Let's replace state with numbers $Q = Q_2 Q_1 Q_0$ to represent the state)

| Present State | | | Input | Next state | | | Output | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_2$ | $Q_1$ | $Q_0$ | w | $Q_2^+$ | $Q_1^+$ | $Q_0^+$ | $Z_{000}$ | $Z_{00}$ | $Z_{001}$ | $Z_{010}$ | $Z_{01}$ | $Z_{011}$ | $Z_{100}$ | $Z_{10}$ | $Z_{101}$ | $Z_{110}$ | $Z_{11}$ | $Z_{111}$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Since it is a Moore State Machine, Z only depends on Q

Also, we know that the middle segment is turned on only when either segment on the leaves segment is Turned on in order to form a square

$$\therefore \quad Z_{00} = Z_{000} \mid Z_{001} \qquad \text{where } \mid \text{ represents OR}$$
$$Z_{01} = Z_{010} \mid Z_{011}$$
$$Z_{10} = Z_{100} \mid Z_{101}$$
$$Z_{11} = Z_{110} \mid Z_{111}$$

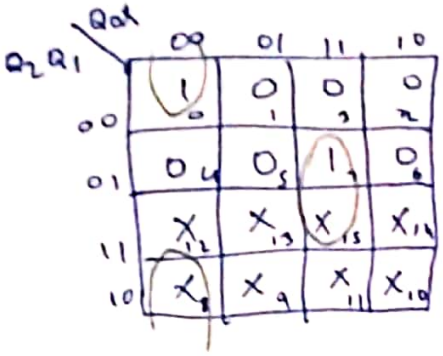
we can verify the same from the truth table.

Let us create the table for JK flip flop :-

$$Q_x^+ = J Q_x' + K' Q_x$$

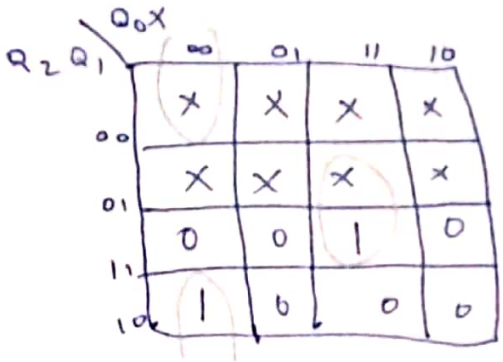
| Present State | | | Input | Next State | | | JK Values | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_2$ | $Q_1$ | $Q_0$ | (in | $Q_2^+$ | $Q_1^+$ | $Q_0^+$ | $J_2$ | $K_2$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | X | 1 | X | 1 | X |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | X | 0 | X | 1 | X |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | X | 0 | X | X | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | X | 1 | X | X | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | X | X | 1 | 1 | X |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | X | X | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | X | X | 0 | X | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | X | X | 1 | X | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | X | 1 | 1 | X | 1 | X |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | X | 0 | 0 | X | 1 | X |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | X | 0 | 0 | X | X | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | X | 0 | 1 | X | X | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | X | 0 | X | 1 | 1 | X |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | X | 0 | X | 0 | 1 | X |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | X | 0 | X | 0 | X | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | X | 1 | X | 1 | X | 1 |

$J_2$:



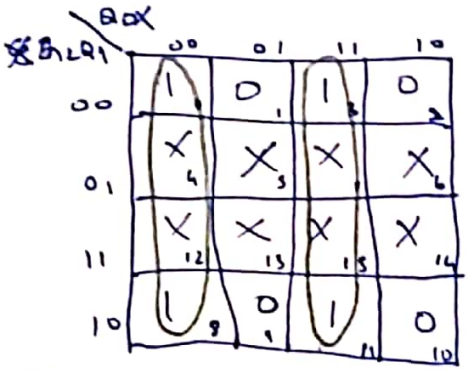

$J_2 = \bar{Q}_1 \bar{Q}_0 \bar{X} + Q_1 Q_0 X$

$K_2$:




$K_2 = \bar{Q}_1 \bar{Q}_0 \bar{X} + Q_1 Q_0 X$

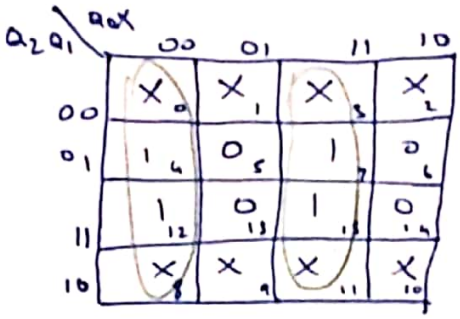
$J_1$:




$J_1 = \bar{Q}_0 \bar{X} + Q_0 X$

$= Q_0 (XNOR) X$

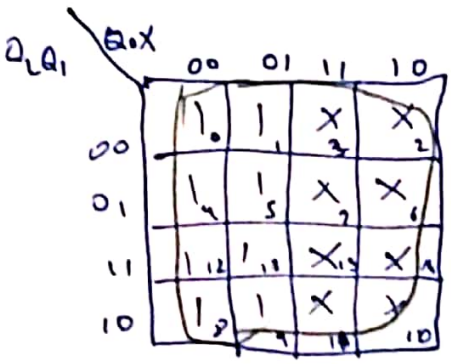
$K_1$:




$K_1 = \bar{Q}_0 \bar{X} + Q_0 X$

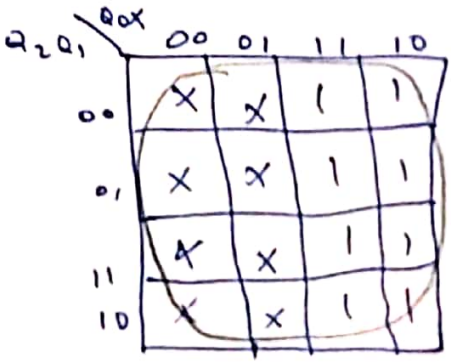
$= Q_0 \ XNOR \ X$

$J_0$:




$J_0 = 1$

$K_0$:




$K_0 = 1$

Each of the 2 other than the middle segments have only 1 minterm

$$\therefore \quad Z_{000} = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 = S_0$$
$$Z_{001} = Q_2 Q_1 Q_0 = S_7$$
$$Z_{010} = \bar{Q}_2 \bar{Q}_1 Q_0 = S_1$$
$$Z_{011} = Q_2 \bar{Q}_1 \bar{Q}_0 = S_6$$
$$Z_{100} = \bar{Q}_2 Q_1 \bar{Q}_0 = S_2$$
$$Z_{101} = Q_2 \bar{Q}_1 Q_0 = S_5$$
$$Z_{110} = \bar{Q}_2 Q_1 Q_0 = S_3$$
$$Z_{111} = Q_2 \bar{Q}_1 \bar{Q}_0 = S_4$$

} States that enable the respective Z

And as seen before,

$$Z_{00} = Z_{000} \mid Z_{001} = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_2 Q_1 Q_0 = S_0 + S_7$$

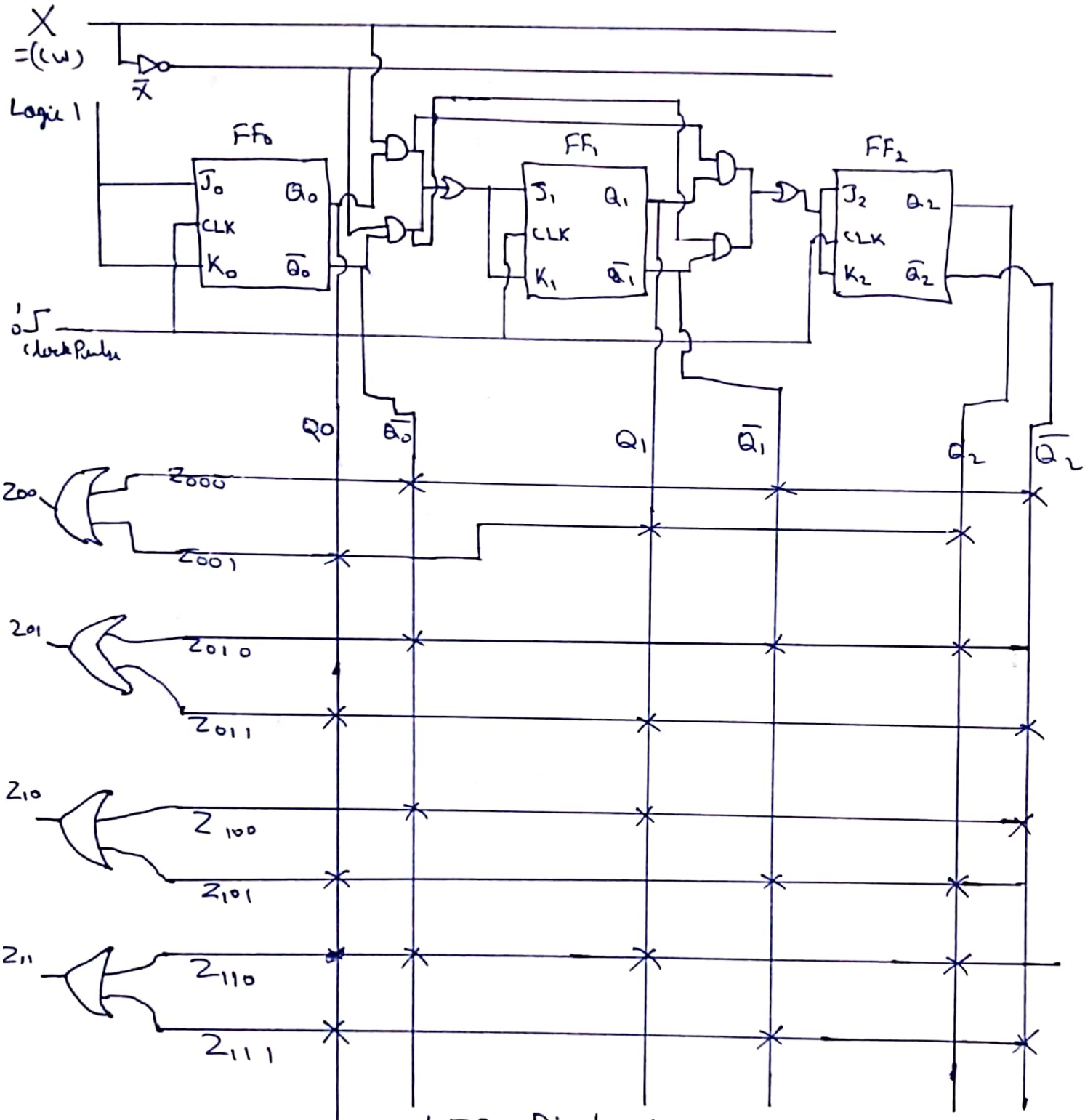
Similarly, $Z_{01} = \bar{Q}_2 \bar{Q}_1 Q_0 + Q_2 Q_1 \bar{Q}_0 = S_1 + S_6$

$$Z_{10} = \bar{Q}_2 Q_1 \bar{Q}_0 + Q_2 \bar{Q}_1 Q_0 = S_2 + S_5$$

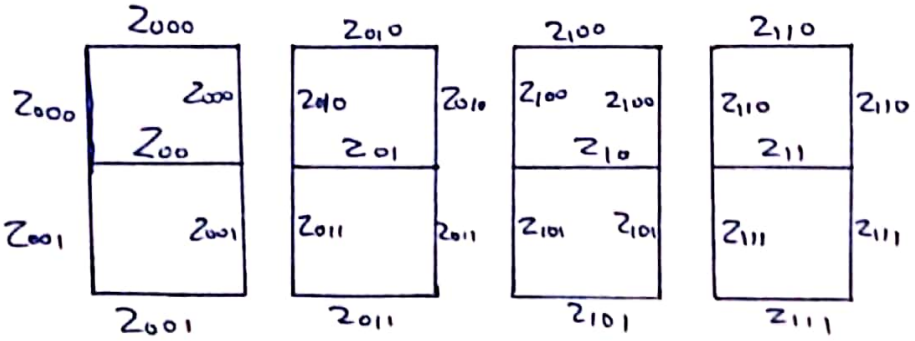
$$Z_{11} = \bar{Q}_2 Q_1 Q_0 + Q_2 \bar{Q}_1 \bar{Q}_0 = S_3 + S_4$$

Each minterm is a complementary portion of the other

Circuit Diagram :-




LED Display :-



```
/*
    The following program is executed on the terminal with input from an
    input file and output printed on the terminal. Hence, input and output
    do not have their respective keywords in the declaration.
*/

module main;
    integer i, file, display;
    reg [8:0] input_c; // input as a character read from the file
    reg [3:0]out; // output whose value used to display the led structure

    function [3:0]disp(input [3:0]cur_state);
        integer j, strbit;
        reg[88:0] string; // The LED denoted as a string of 11 characters each line which are 8 bits each
        begin
            $display("State = %0d", cur_state); // print the state
            for(j = 0; j < 3; j++) begin
                assign string = ".. .. .. ..";
                /*
                    Find the position of the LED to light up using the logic obtained from the Truth table.
                    A simplified logic is used in the code below.
                    Notation:
                        Lightend up -   ##
                        Otherwise -     ..
                    Initially all are set to ..
                    Hence, we need to add (("##" - "..") * power(2, starting position))
                    which is equal to (("##" - "..") << starting position)
                */
                if (cur_state >= 4) begin
                    if (j >= 1) begin
                        assign strbit = ((cur_state - 4) * 3) * 8;
                        assign string = string + (("##" - "..") << strbit);
                    end
                end
                if (cur_state < 4) begin
                    if (j <= 1) begin
                        assign strbit = ((3 - cur_state) * 3) * 8;
                        assign string = string + (("##" - "..") << strbit);
                    end
                end
                $display("%s", string); // display the current LED row
            end
            $display(""); // new line
        end
    endfunction

    initial
        begin
            file = $fopen("input.txt", "r");
            assign out = 0;
            $display("Initially");

            // display the inital LED state
            display = disp(out);

            input_c = $fgetc(file) - 48; // 48 is the ASCII value of '0'
            while (input_c != ('h1ff - 48)) begin
```

```
58                 $display("Input = %d", input_c);
59                 if (input_c == 1)
60                     assign out = (out + 1) % 8; // state increases by 1 and
   goes anticlockwise
61                 else
62                     assign out = (out + 7) % 8; // state reduces by 1 and
   goes clockwise
63 
64                 // display the LED
65                 display = disp(out);
66 
67                 // Read new input
68                 input_c = $fgetc(file) - 48;
69             end
70             $finish;
71         end
72 endmodule
```

```
1111111110001101
```

```
Initially
State = 0
 ## .. .. ..
 ## .. .. ..
 .. .. .. ..

Input =   1
State = 1
 .. ## .. ..
 .. ## .. ..
 .. .. .. ..

Input =   1
State = 2
 .. .. ## ..
 .. .. ## ..
 .. .. .. ..

Input =   1
State = 3
 .. .. .. ##
 .. .. .. ##
 .. .. .. ..

Input =   1
State = 4
 .. .. .. ..
 .. .. .. ##
 .. .. .. ##

Input =   1
State = 5
 .. .. .. ..
 .. .. ## ..
 .. .. ## ..

Input =   1
State = 6
 .. .. .. ..
 .. ## .. ..
 .. ## .. ..

Input =   1
State = 7
 .. .. .. ..
 ## .. .. ..
 ## .. .. ..

Input =   1
State = 0
 ## .. .. ..
 ## .. .. ..
 .. .. .. ..

Input =   1
State = 1
 .. ## .. ..
 .. ## .. ..
 .. .. .. ..

Input =   1
State = 2
 .. .. ## ..
 .. .. ## ..
 .. .. .. ..

Input =   0
State = 1
 .. ## .. ..
```

```
   .. ## .. ..
   .. .. .. ..

 Input =   0
 State = 0
  ## .. .. ..
  ## .. .. ..
   .. .. .. ..

 Input =   0
 State = 7
  .. .. .. ..
  ## .. .. ..
  ## .. .. ..

 Input =   1
 State = 0
  ## .. .. ..
  ## .. .. ..
   .. .. .. ..

 Input =   1
 State = 1
  .. ## .. ..
  .. ## .. ..
  .. .. .. ..

 Input =   0
 State = 0
  ## .. .. ..
  ## .. .. ..
   .. .. .. ..

 Input =   1
 State = 1
  .. ## .. ..
  .. ## .. ..
  .. .. .. ..
```