Design Report

Digital Logic Design

SMART CAR PARKING SYSTEM

BEE 11-B GROUP NUMBER 4

NATIONAL UNIVERSTIY OF SCIENCES AND TECHNOLOGY | H-12 Sector Islamabad

TEAM LEADER:

Name: Ch. Muhammad Shaheer Yasir

Reg. No. : 286021

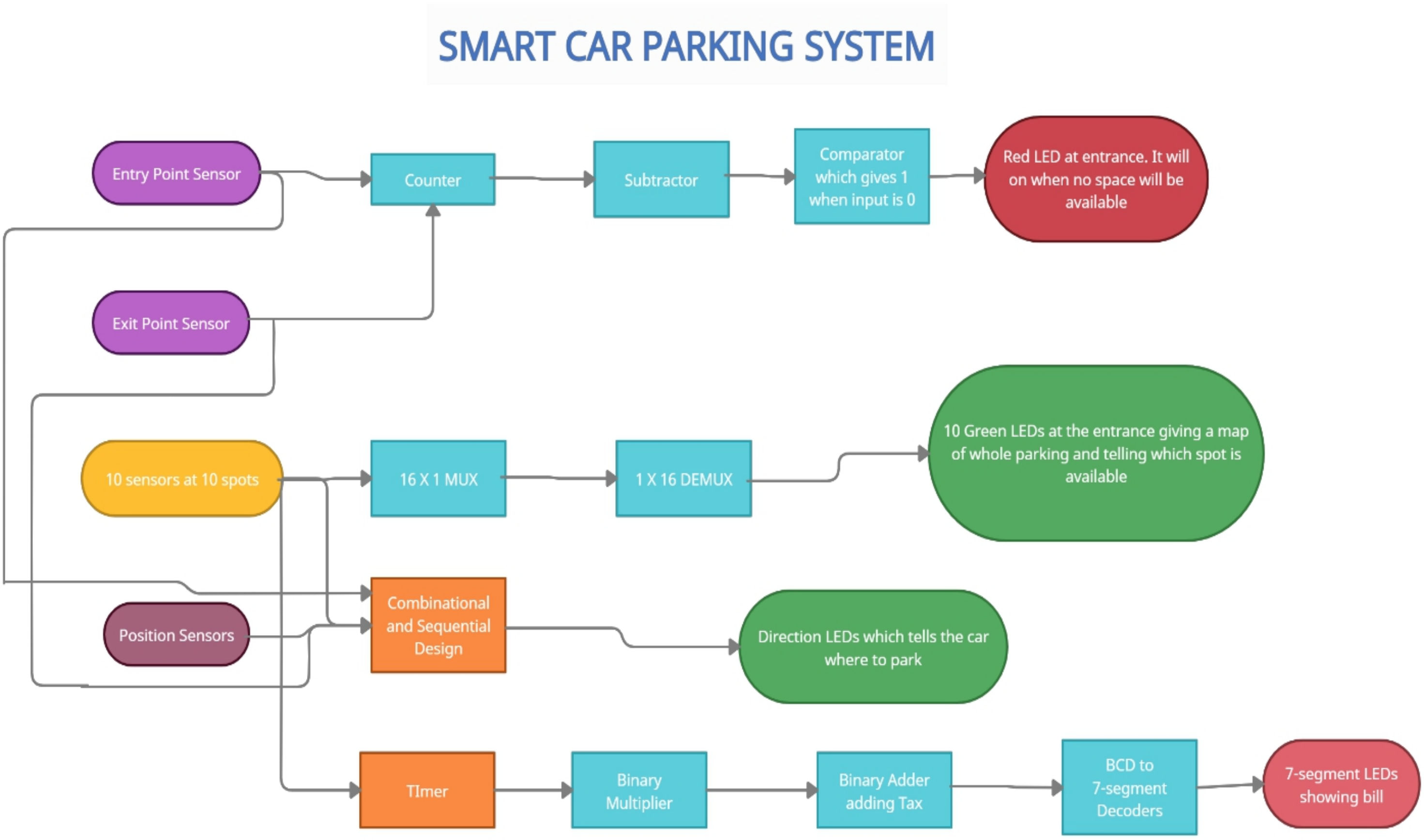
TEAM MEMBERS:

Farheen Gul- 290352

Muhammad Hamza- 290951

Amna Abid- 321913

BLOCK DIAGRAM OF SYSTEM PROPOSED



Amna Abid's Part:

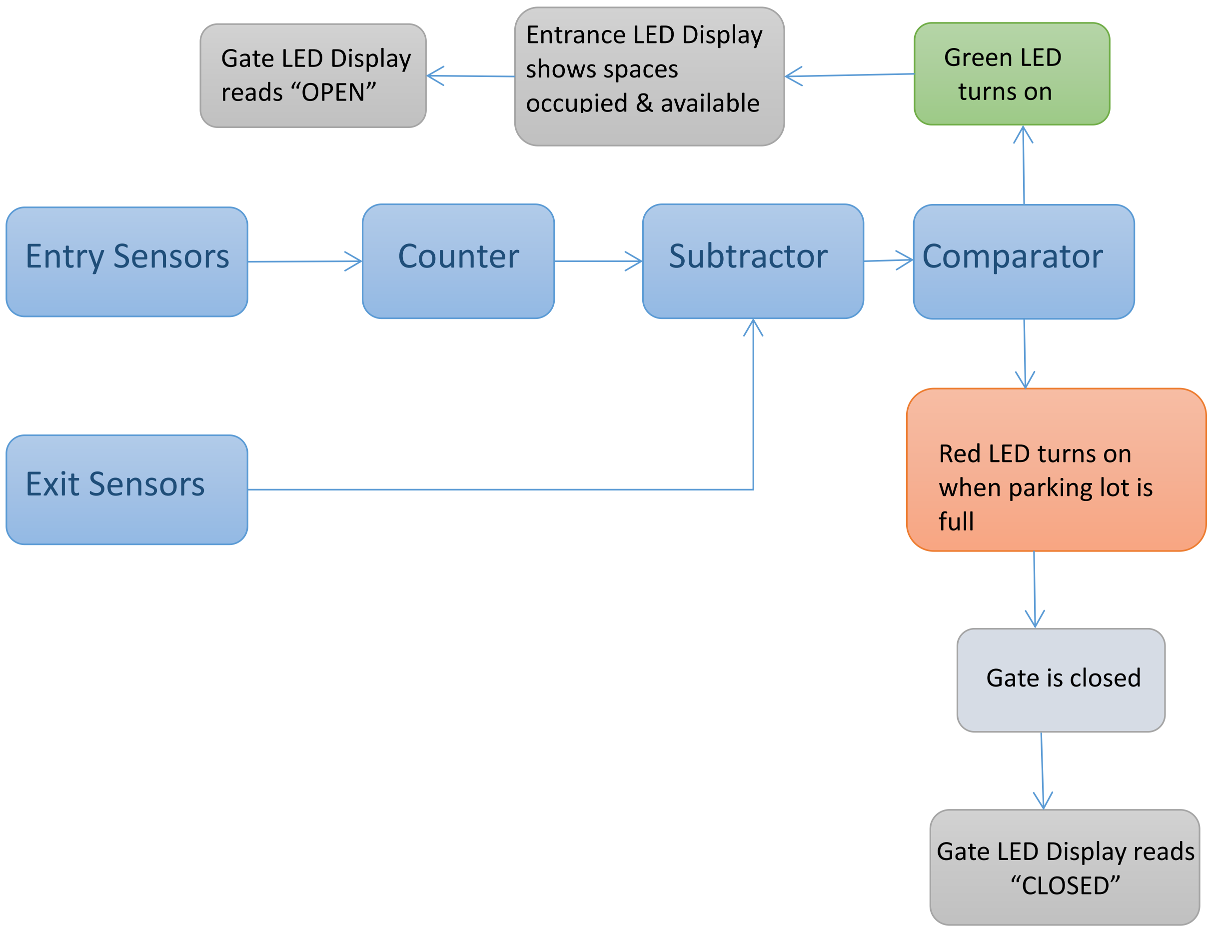
Parking Space Indicator:

Two motion sensors are located at the entrance and the exit of the parking lot. A counter keeps track of the number of cars that have entered and not left the parking lot in real time.

In the Entrance Booth of the parking lot, two 7 segment LED Displays keep the real time record on display showing the current number of parking spaces that are preoccupied by vehicles and the current number of parking spaces that are unoccupied. Each of the two 7 segment LED Displays are preceded by two LED lights; one green, and one red. Once the parking lot reaches it's maximum capacity, the red LED turns on while at all other times the green LED remains on signaling the presence of an empty parking spot within the parking lot.

On the Entrance Gate an LED display shows the current status of the gate. When there is space available for parking within the parking lot, the gate remains open and the LED display on the gate reads "OPEN". Once the parking lot is fully occupied, and the red LED has turned on to indicate such to the driver outside at the entrance, then the gate at the entrance will be closed not allowing entry to any more vehicles outside, unless one or more vehicles leave the parking lot premises, and the LED display on the gate will now read "CLOSED".

Block Diagram:

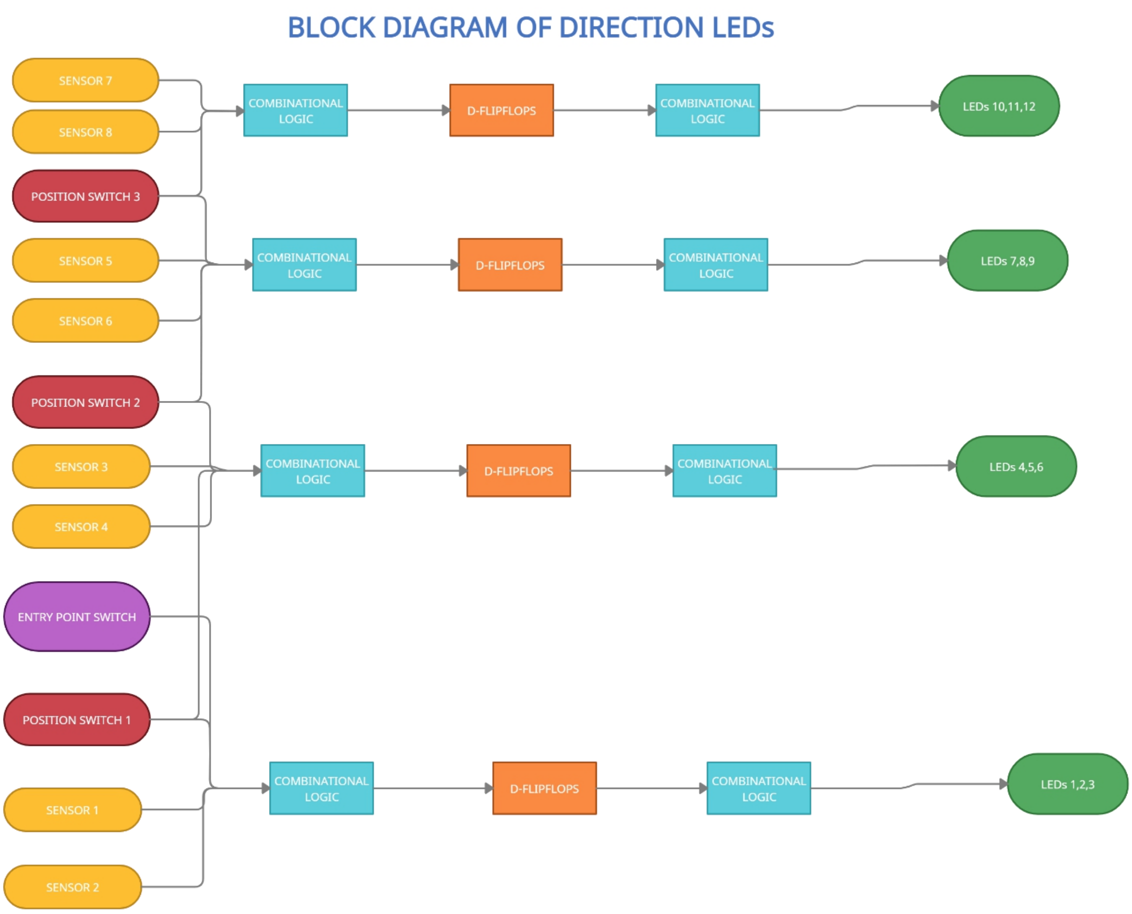


Sequence of design steps:

- Sensors located at the entrance will give the inputs to the counter as each vehicle enters.
- The counter is constructed using the help of flip flops.
- If a vehicle or vehicles exits the parking lot, a subtractor will deduct that specific number from the counter's output.
- The final number of occupied places counted will be displayed on a 7 segment LED display.
- The counted number will be subtracted from the total number of places available (8) using a subtractor and will be displayed as the number of free spaces available on a 7 segment LED display.
- If the counter reaches a total count of 8, then the red LED will be switched on and green LED switched off.
- The Gate will be closed and the gate LED display will stop displaying "OPEN" and "CLOSED" will be displayed instead.

Ch. Muhammad Shaheer Yasir's Part:

Block Diagram:



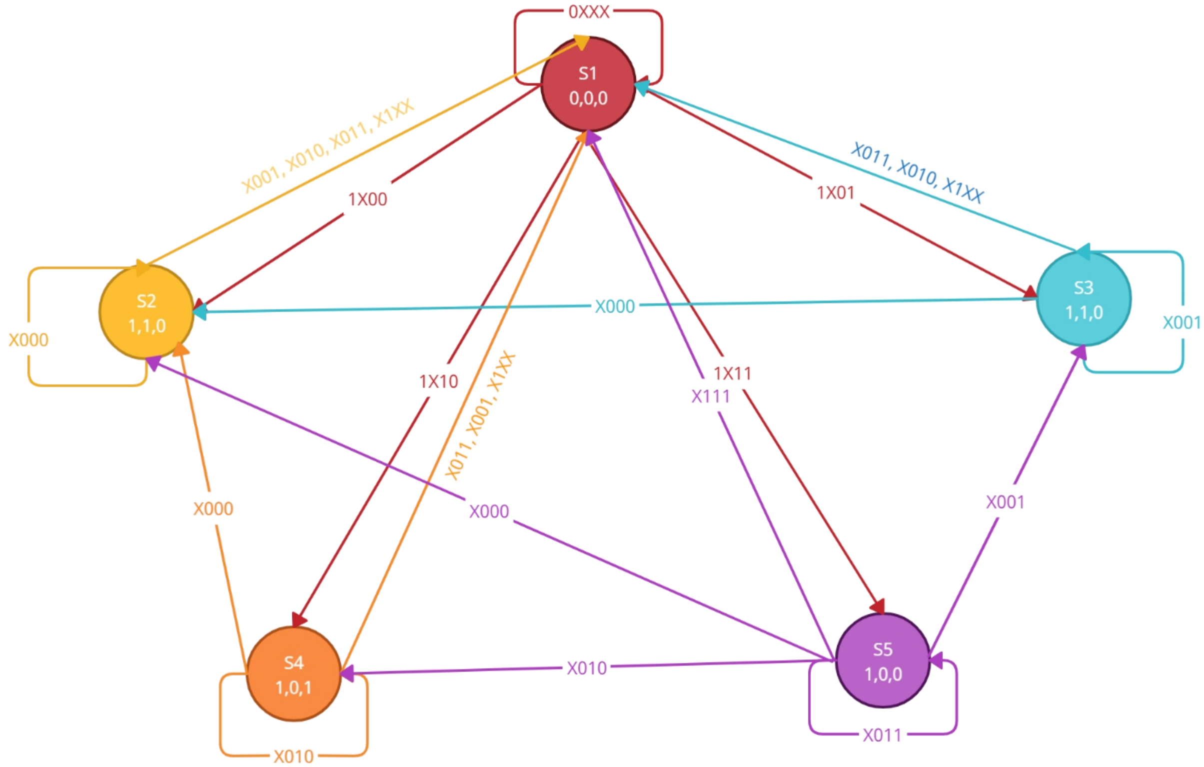
Sequence of Design Steps:

- First, we defined our states i.e. different situations which can exist in this module.
- Then we draw the state diagram and make state table.
- Then we make K-maps and defined next state and LEDs as functions of present state and inputs. So, our design is a Mealy machine design.
- As we have about five states, so we used three D-flip flops for first three LEDs respectively.
- For the next three LEDs, we implemented with exactly same logic but now we take position switch 1 as entry point input and position switch 2 as position input.
- Similarly, for LEDs 7,8 and 9, same logic is applied but entry input is position switch 2 and position input is position switch 3.
- For last three LEDs, as there is not position switch ahead so there is no position input and position switch 3 is taken as entry input.

Explanation of circuit diagram:

For the first three LEDs, we defined the states and then, from these we made state diagram and state table and defined next states which are inputs to our D-flipflops and output functions. Then for next 9 LEDs, similar sequential design procedure is opted to design them. So, the states we defined are as follows:
State 1 (000): Initial State. No Car entered the parking yet. Everything will be off.
State 2 (001): Car enters. First two spots are both available.
State 3 (010): Car enters. The first spot is available but the second is filled.
State 4 (011): Car enters. The second spot is available but the first is filled.
State 5(100): Both spots are filled.
If state 5 occurs then it will be the same state as 001, 010 or 011 for next three LEDs.
To check this state, we have position switch between spot1,2 and spot 3,4. When this switch will be logic 1, it will do the same as entry point switch for next two spots.
The state diagram of the design is as follows:

STATE DIAGRAM OF DIRECTION LEDs



The state table of our design is as follows:

Present State			Input			Next State			Output		
Q_A	Q_B	Q_C	ES	S_1	S_2	Q_A^+	Q_B^+	Q_C^+	L_1	L_2	L_3
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	1	0	1	1	0
0	0	0	1	1	0	0	1	1	1	0	1
0	0	0	1	1	1	1	0	0	1	0	0
0	0	1	0	0	0	0	0	1	1	1	0
0	0	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0
0	0	1	1	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	1	1	0
0	1	0	0	0	1	0	1	0	1	1	0
0	1	0	0	1	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	1	1	1	0
0	1	0	1	0	1	0	1	0	1	1	0
0	1	0	1	1	0	0	0	0	0	0	0

0	1	0	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	1	0
0	1	1	0	0	1	0	0	0	0	0	0
0	1	1	0	1	0	0	1	1	1	0	1
0	1	1	0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	1	0
0	1	1	1	0	1	0	0	0	0	0	0
0	1	1	1	1	0	0	1	1	1	0	1
0	1	1	1	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1	1	0
1	0	0	0	0	1	0	1	0	1	1	0
1	0	0	0	1	0	0	1	1	1	0	1
1	0	0	0	1	1	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1	1	1	0
1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	1	1	0	0	1	1	1	0	1
1	0	0	1	1	1	1	0	0	1	0	0

We haven't used Position Sensor 1 as input, because when it will be 0 this state table will work. When it will be 1, next state will be 000 and L1, L2 and L3 will also be zero. Then this PS1 will act as entry point sensor for the next three LEDs and design procedure will be same then.
Using K-maps, we found next states and outputs as function of present state and inputs. Their function is as follows:

$Q_A^+ = PS_1' (Q_B'Q_C' S_1S_2 (ES + Q_A))$
$Q_B^+ = PS_1' (S_1S_2' (Q_B'Q_C' (ES + Q_A) + Q_A'Q_BQ_C) + Q_C' S_1' S_2 (Q_A' (Q_B + ES) + Q_A Q_B'))$
$Q_C^+ = PS_1' (S_2' (Q_B' Q_C' (ES + Q_A) + Q_A' (Q_C (S_1' + Q_B) + Q_B S_1')))$
$L_1 = PS_1' (Q_B' Q_C' (ES + Q_A) + Q_A' (Q_c S_2' (S_1' + Q_B) + Q_c' S_1' Q_B))$
$L_2 = PS_1' (S_1' (Q_A' Q_C' (Q_B + ES) + Q_A' Q_C S_2' + Q_A Q_C' Q_B')$
$L_3 = PS_1' (S_1S_2' (Q_B' Q_C' (ES + Q_A) + Q_A' Q_B Q_C))$

For the next LEDs the logic is similar except we will treat previous position switch as their entry point input. So, they will only be turn on when the car crosses the position switch 1. For the last three LEDs, similar logic is implemented but there is no position switch ahead so, the position sensor input will be eliminated, and functions will be somehow simplified as:

$Q_A^+ = (Q_B'Q_C' S_1S_2 (ES + Q_A))$
$Q_B^+ = (S_1S_2' (Q_B'Q_C' (ES + Q_A) + Q_A'Q_BQ_C) + Q_C' S_1' S_2 (Q_A' (Q_B + ES) + Q_A Q_B'))$
$Q_C^+ = (S_2' (Q_B' Q_C' (ES + Q_A) + Q_A' (Q_C (S_1' + Q_B) + Q_B S_1')))$
$L_1 = (Q_B' Q_C' (ES + Q_A) + Q_A' (Q_c S_2' (S_1' + Q_B) + Q_c' S_1' Q_B))$
$L_2 = (S_1' (Q_A' Q_C' (Q_B + ES) + Q_A' Q_C S_2' + Q_A Q_C' Q_B')$
$L_3 = (S_1S_2' (Q_B' Q_C' (ES + Q_A) + Q_A' Q_B Q_C))$

Farheen Gul's Part:

Billing system of Parking lot:

Billing system of parking lot is made such that it increases the parking bill after every hour and also tax is included in the bill. The bill is then displayed using seven segment LED system. Each of the 8 parking places have the same arrangement of LEDs and integrated circuit for displaying bill. But in our project we have made the billing system for only one parking place.

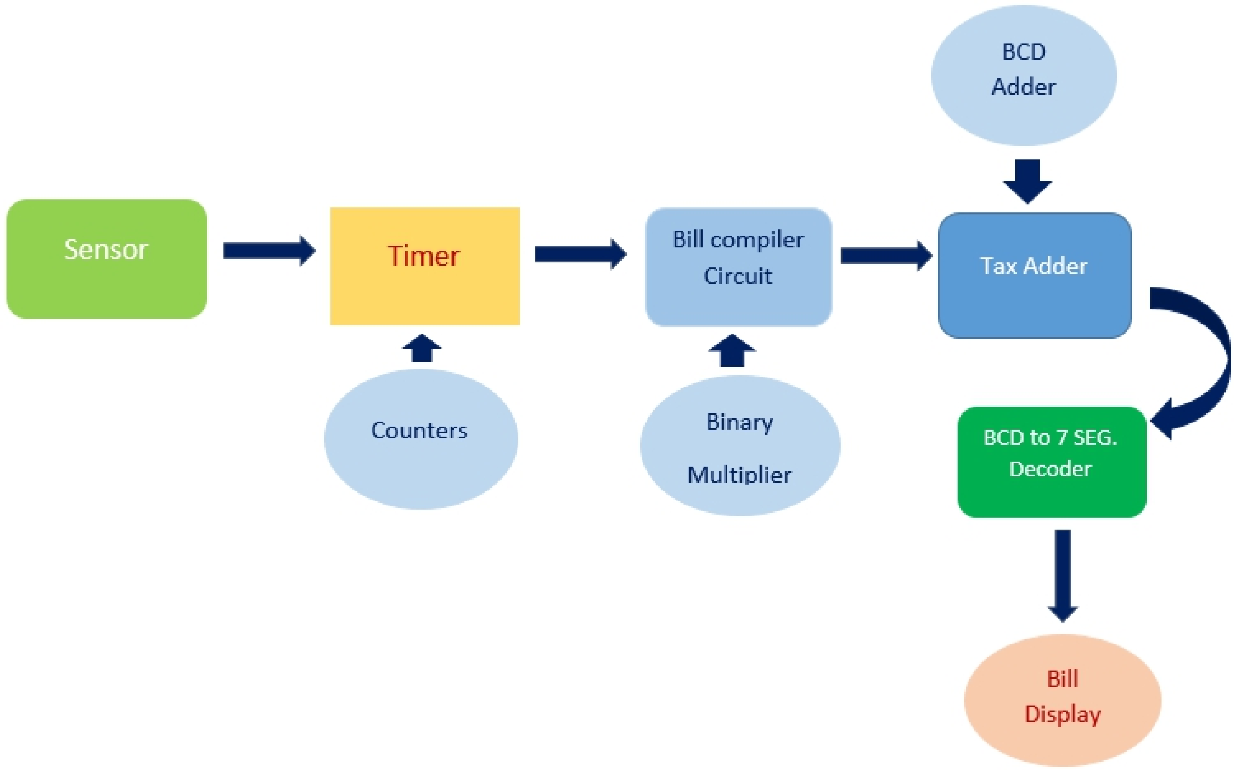
Basically it consists of an arrangement of counters, binary multiplier circuit, and BCD adder for adding tax and an arrangement of seven segment LEDs for displaying bill.

The parking price is 5 Rupees for 1 hour and it goes on increasing till the vehicle is parked in the parking lot, and tax of 2 rupees is added in the total parking price which is then displayed. This all process is managed with a sensor LED which turns on when the vehicle is parked and turns off when the Vehicle leaves the place, the output of this LED act as an input for the counter which is calculating the time of Parking. Then the time is multiplied by the parking price in a binary multiplier and then tax is added using a BCD adder and then BCD to seven segment decoders are used for display.

Sequence of Design Steps:

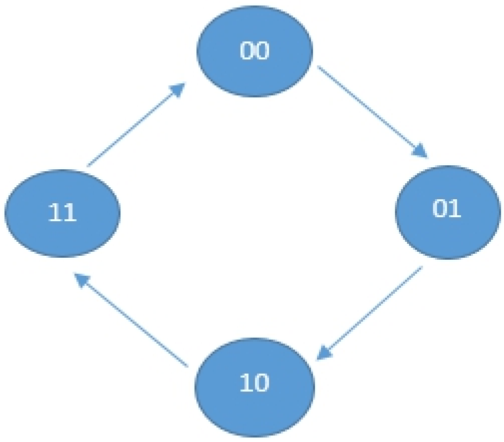
- LED at each of the 8 parking places in Car park turns on as car is parked there and remains on, till the car is there.
- This LED act as the input for the counter which is designed to count the number of hours. Clock frequency in the counter is adjusted that it counts 1 after every 1 minute and this one minute time delay is considered as 1 hour delay in our project.
- The counter we used for counting the number of hours is made of JK flip flops, whose design can be varied according to the number of hours allowed for the parking.
- And the clock which is inserted in the above counter is made using BCD counters, BCD counters are cascaded in special arrangement for this purpose. So that after every 1 minute , which we consider as one hour the JK flip flop counter counts another hour.
- The number of hours of car park are then multiplied by 5 in a binary multiplier, as 1 hour parking bill is fixed as 5 rupees.
- Then a 2 Rupees of Tax is added in the bill in a BCD adder.
- Finally BCD output is displayed on 7 segment LEDs using a BCD to 7-segment decoder.

Block Diagram:



Counter explanation:

State diagram:



Excitation table of JK Flip flop:

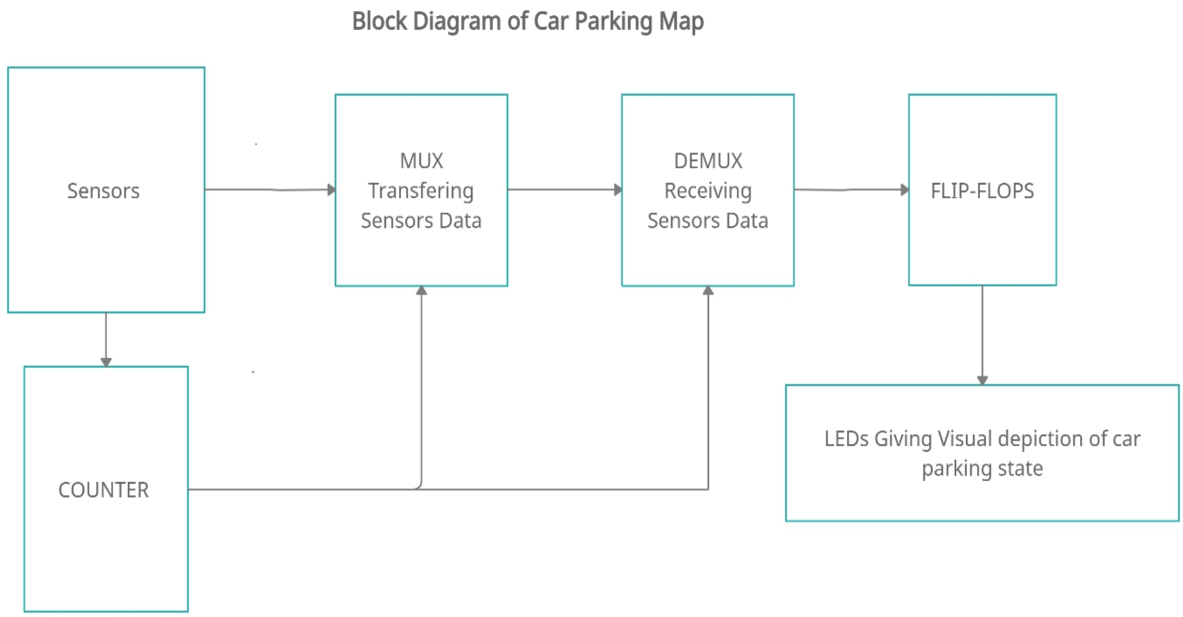
Q n	Q(n+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Circuit Excitation table:

Q1	Q2	Q1*	Q2*	J1	K1	J2	K2
0	0	0	1	0	x	1	X
0	1	1	0	1	X	x	1
1	0	1	1	x	0	1	X
1	1	0	0	x	1	x	1

Muhammad Hamza's Part:

Block Diagram of purposed system:



Design Explanation:

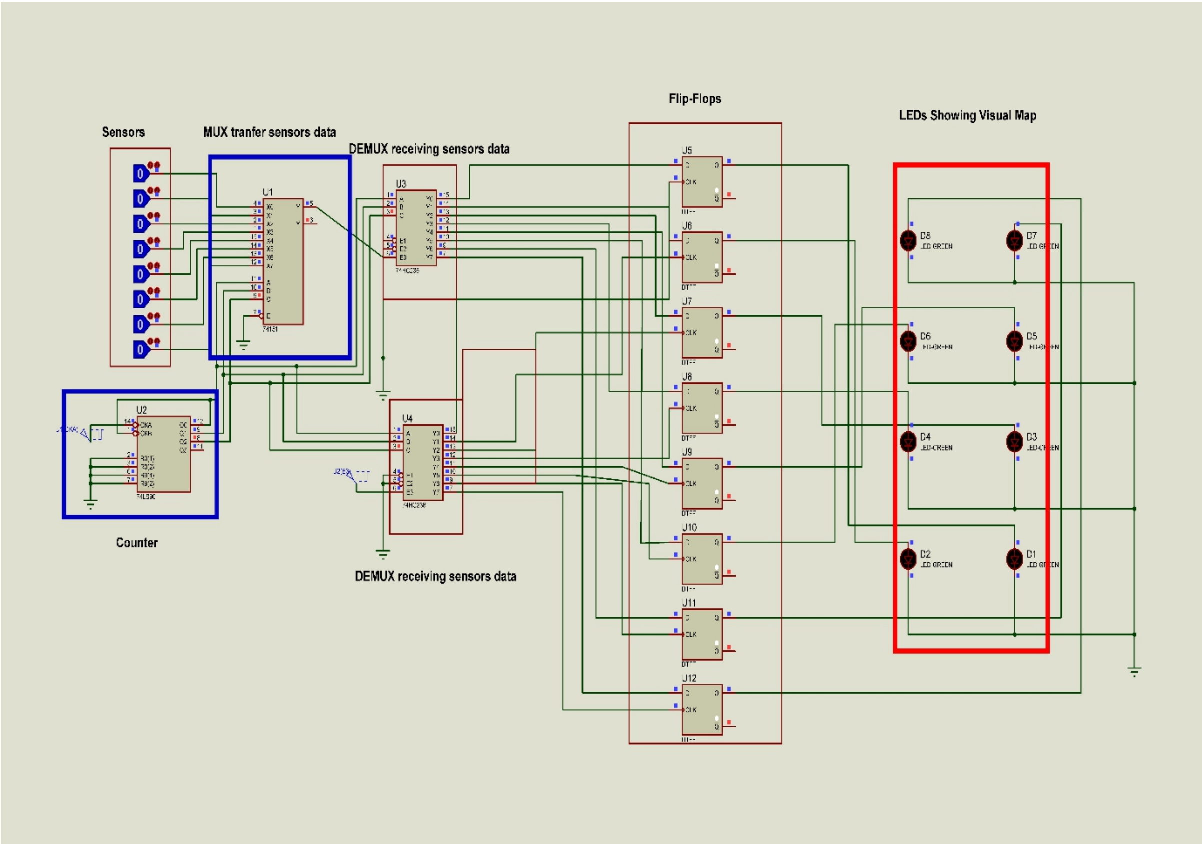
The main purpose was to design the car parking map at the entrance which gives the driver a visual depiction of which spots are available, and which are filled. The system will take input from 8 sensors installed at 8 different parking spots and then a multiplexer will be used to convert these inputs in to one output. That output is again de-multiplexed at the entrance giving 8 outputs which are then connected to 8 LEDs in the parking map. The use of multiplexers was to use minimum no. of wires coming from sensors design to the map at entrance. Flip-Flops were used before LEDs because of the use of D-Clock generator as an input in the counter. These output from counter also serve as select lines for the multiplexers and De-multiplexers.

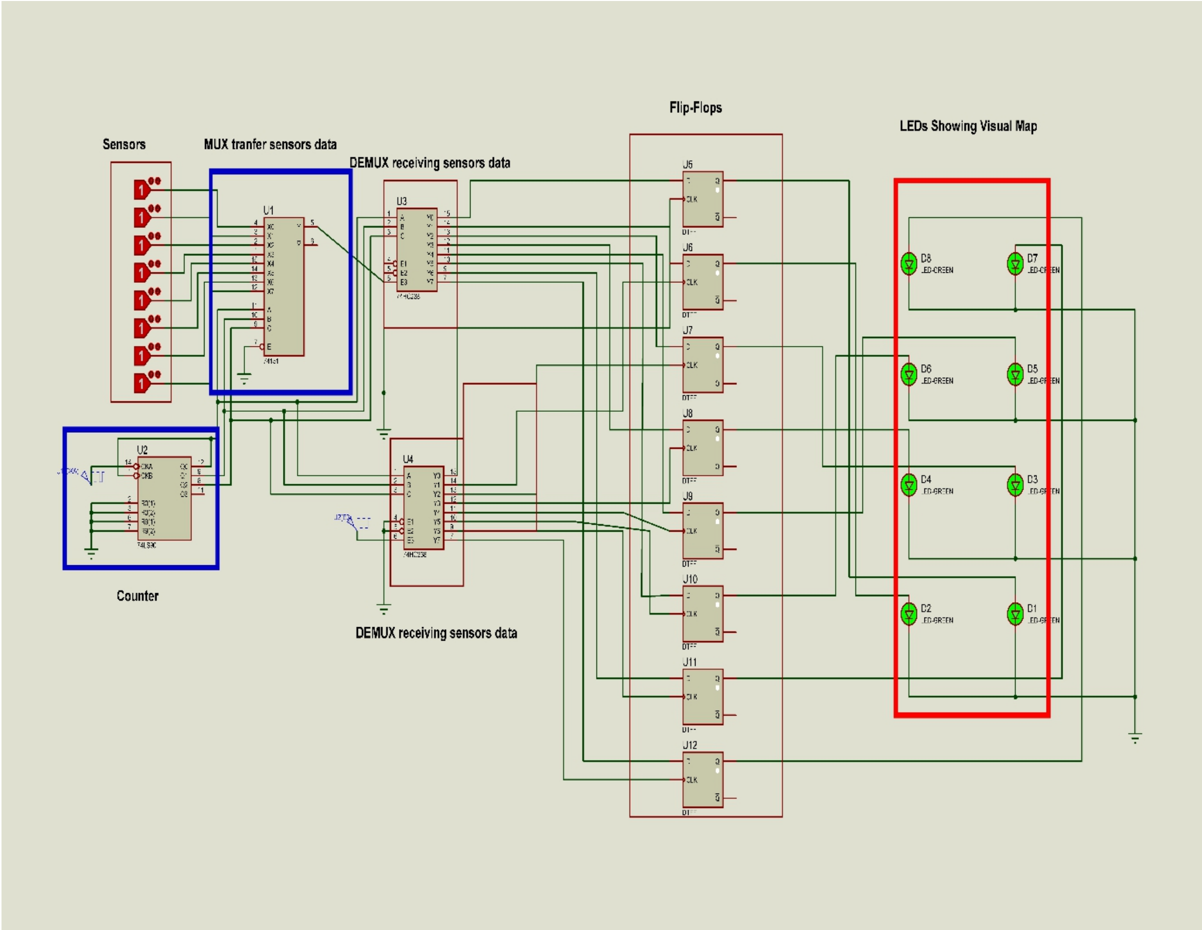
Sequence of Design steps:

In order to design a map of car parking which will give an idea to the user of empty and filled slots in the parking we proceed as:

1. Eight sensors at 8 parking spots which will give 8 inputs whether the slot is empty or full.
2. Since there are 8 inputs so we use 8x1 multiplexer. Here we have used 74ls51, which will convert 8 inputs to one output.
3. Counter(74LS90) will be used. A D-Clock generator is used to produce a series of alternating inputs.
4. The output is again De-multiplexed using two De-mux (74HC238). Here an another D-Clock generator is used.
5. The output from De-mux will now serve as input to Now 8 flip-flops(DTFF).
6. In the ends 8 Green LEDs are connected to these Flip-Flops output which will give the final illustration of car parking states

Schematics and Simulation Result




Flip-Flops
LEDs Showing Visual Map
Sensors
MUX tranfer sensors data
DEMUX receiving sensors data
U1
U3
U2
U4
J5
J6
J7
J8
J9
J10
J11
J12
D8
D7
D6
D5
D4
D3
D2
D1
Counter
DEMUX receiving sensors data