

## **FPGA Project**

# **RAILWAY TICKET RESERVATION SYSTEM**

**Name:** G.Sriharsha

**Name:** G.Badrinath Reddy

**Name:** K.Vamsi Pramod

### **Aim:**

**Allotment of limited number of tickets(tatkal) to the users in a similar way of online railway ticket booking**

### **Apparatus**

SEVEN SEGMENT DISPLAY  
LED  
FPGA  
BREAD BOARD  
RESISTORS  
ULN DRIVER

### **About Railway System**

Railway website can be accessed by user who has a login id and password. When he enters in to the website he checks whether the tickets are available or not. If the tickets are available he will further proceed to book that tickets. Then a session will be allocated to the user to make payment for booking ticket. If he

fails to make payment in that session he will be thrown out of the website. If he pays in that session then he will receive booking confirmation message.

### **Specifications**

- \*At a time ,maximum 4 users can access the railway website.
- \*A user can book maximum 3 tickets at a time.
- \*Number of tickets available are 15.
- \*User should make payment with in 15 seconds (session).
- \*Number of tickets available are displayed on 7segment display

### **Description**

A particular user can enter in to the website by pressing the push button. Since the maximum number of users access website at a time is 4 , four push buttons will be there. When the user entered in to the website , a session will be allocated to that user. With in that session he should complete the whole process of ticket booking . This operation is done by the counter. The counter starts when the user entered in to the website. counter will run from 0 to 15.

Number of tickets available are displayed on 7segment display. When the user entered in to the website he checks the available number of tickets on the seven segment display. Two inputs will be given to the user to enter number of tickets he want to buy. Using those inputs(binary) he can buy atmost 3 tickets. He checks whether the input given is more or less than available number of tickets on the red led light. The red led light will be on if the available tickets are more than or equal to the user input. If not the red led will be off.

Then he makes payment by pressing the same push button which he used to enter the website. Then he receives the confirmation message from a green led(which will be on if payment is successful). Now the number of tickets available will be updated i.e difference of number of tickets available before

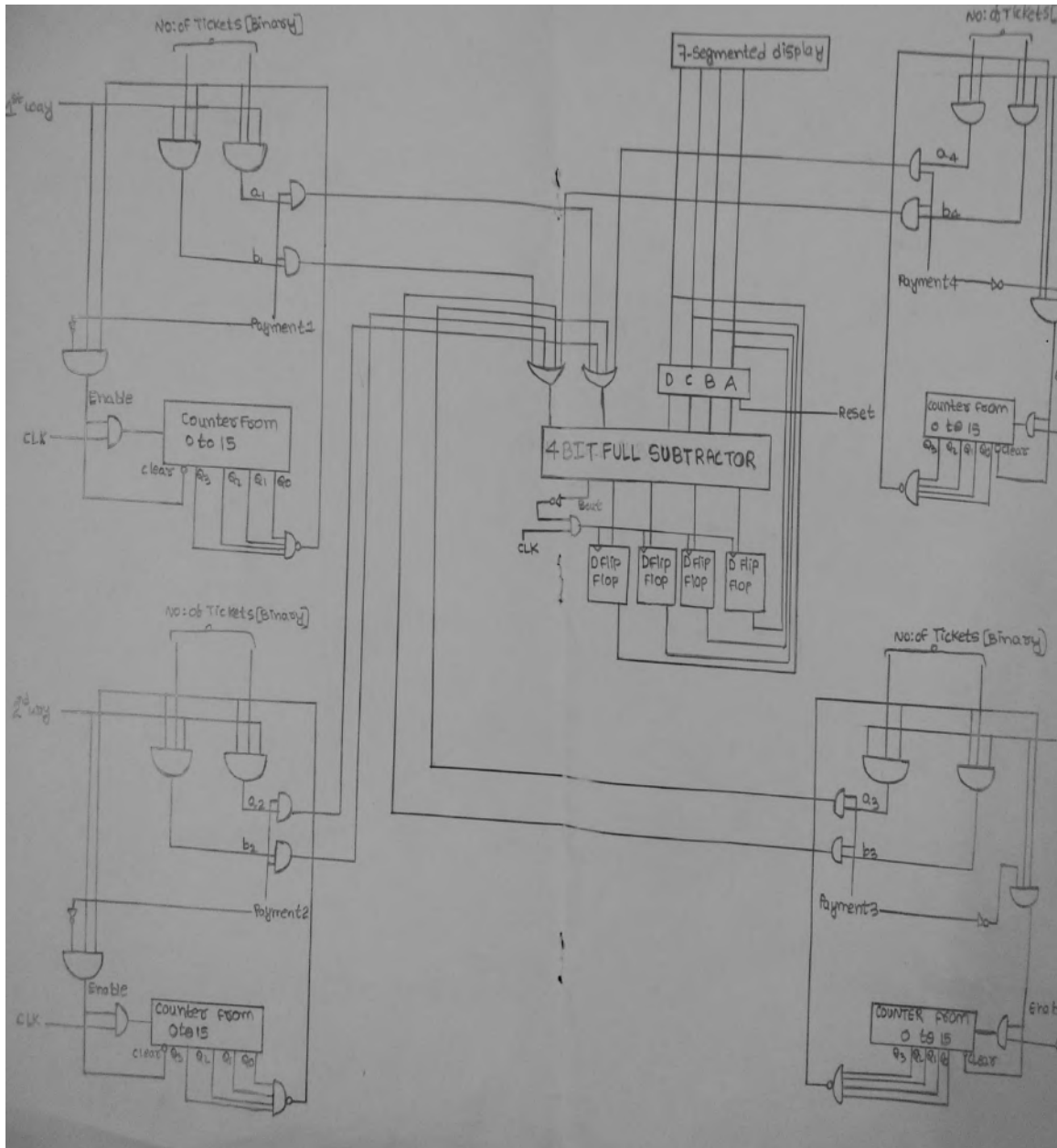
booking and number of tickets user booked . This operation is done by full subtractor.

At a time , 4 different users can access the website through 4 ways . The same process is done for all the users. Once the above process is completed by user , another user can enter the website or if the session of that user completes another user can enter the website. If any user is ongoing through the process , another user can't enter the website through that way. The next user can check this by the led light. The led light will be on if any user is ongoing through the process. So the next user can enter only after that led is off.

If suppose 3 tickets are available , if the first user enters and select the tickets but he didn't complete the payment in the session during that time another user may enter , select and pays the money then the second user gets the 3 tickets though the first user session is going on .

To reset the entire process , give the reset input as high.

### **Circuit Diagram**



### Explanation of Circuit Diagram:

- \*When the user enters the website the **clock Enable** is high and the **clear** is high so the counter starts running from "0".
- \*And the user can choose number of tickets upto 3 in meantime.
- \*The clock Enable remains high until the counter reaches value 15 or the user makes payment.

**\*CASE1:**

If **counter reaches 15** the count Enable becomes low so user becomes inactive[Displayed by LED1].

\*As soon as the Enable becomes low clear is low so the values of flip-flops in counter are cleared to zero.

\*So when user again enters the website counter starts form zero.

**\*CASE2:**

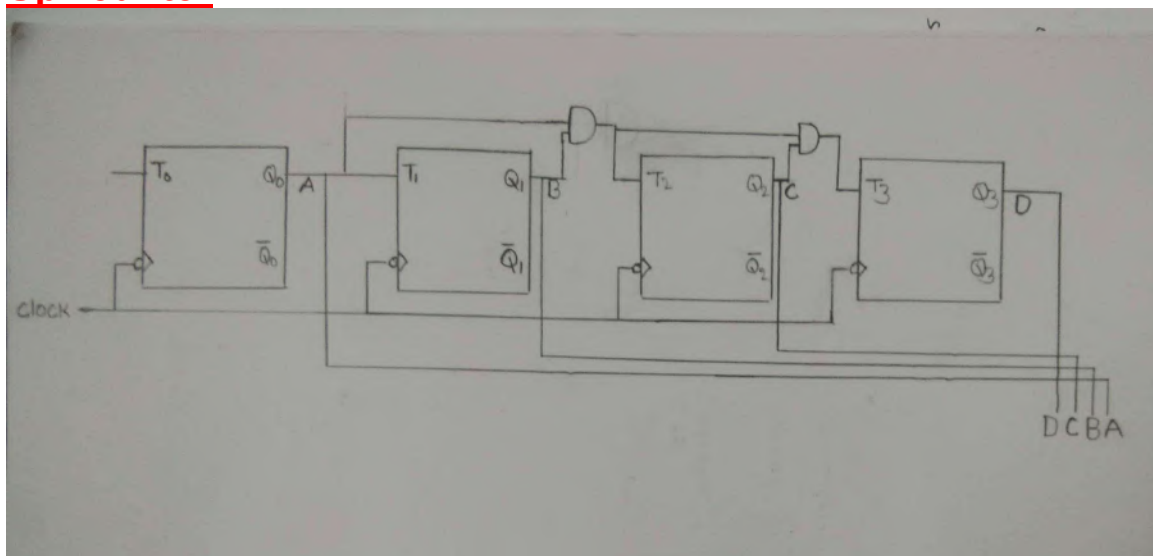
If **user makes payment** the count Enable becomes low and remaining logic is same as in CASE1 for counter

\*If the number of tickets selected by the user are available [Displayed by LED2].

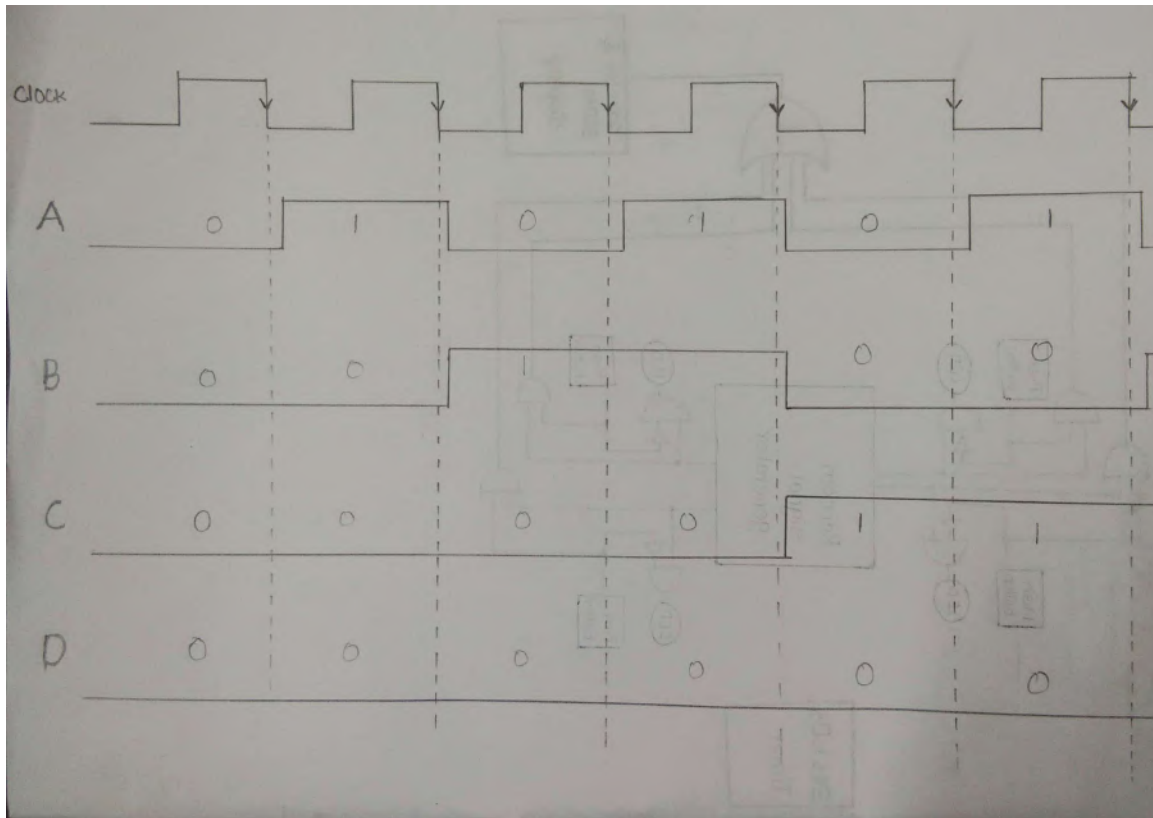
\*Then payment is shown sucessfull [Displayed by LED3]

\*The logic is same for all the users and the clock provided to all the counters is the same clock.

## UpCounter



## TIMING DIAGRAM



This is the timing diagram for an up counter from 0 to 15 and again comes back to 0. We have used 4 up counters and the timing diagram is same for all the counters.

### **INFERENCES:**

The verilog code written in vivado simulator is shown below

```

1  module code(
2      input CLK,
3      input way1,way2,way3,way4,
4      input ticket11,ticket21,ticket31,ticket41,
5      input reset,
6      input ticket12,ticket22,ticket32,ticket42,
7      output a,b,c,d,e,f,g,
8      output reg active1,reg active2,reg active3,reg active4,
9      output reg success1,reg success2,reg success3,reg success4,
10     output reg avail2,reg avail3,reg avail4,
11     output reg avail1
12
13 );
14 reg k1,k2,k3,k4,avail1,temp;
15 reg [4:0]selected1; //no of tickets selecccted by user1
16 reg [4:0]selected2; //no of tickets selecccted by user2
17 reg [4:0]selected3; //no of tickets selecccted by user3
18 reg [4:0]selected4; //no of tickets selecccted by user4
19 wire A,B,C,D;
20 reg [3:0] totaltickets; //Total Number of available tickets
21 reg [32:0] counter;
22 reg [4:0] timer1; //Timer for user1(starts if he enters website)
23 reg [4:0] timer2; //Timer for user2
24 reg [4:0] timer3; //Timer for user3
25 reg [4:0] timer4; //Timer for user4
26 initial counter=0;
27 initial totaltickets=4d15; //Total Number of tickets is 15
28 initial timer1=0; //Initializing all timer variables to 0
29 initial timer2=0;
30 initial timer3=0;
31 initial timer4=0;
32
33 initial k1=0;
34 initial k2=0;
35 initial k3=0;
36 initial k4=0;
37
38 initial success1=0; //Confirms the payment of user if success is 1
39 initial avail1=0; //Avail1 will be high if selected number of tickets are available
40 initial active1=0; //Active1 will be high if any user is in the website
41
42 initial success2=0; //Confirms the payment of user if success is 1
43 initial avail2=0; //Avail2 will be high if selected number of tickets are available
44 initial active2=0; //Active2 will be high if any user is in the website
45
46 initial success3=0; //Confirms the payment of user if success is 1
47 initial avail3=0; //Avail3 will be high if selected number of tickets are available
48 initial active3=0; //Active3 will be high if any user is in the website
49
50 initial success4=0; //Confirms the payment of user if success is 1
51 initial avail4=0; //Avail4 will be high if selected number of tickets are available
52 initial active4=0; //Active4 will be high if any user is in the website
53
54 always @ (posedge CLK)
55 begin
56     if(counter==32d125000000)
57     begin
58         counter=0;
59         if(reset==1) //If reset is high all variables are initialised to 0 again
60         begin
61             k1=0;
62             active1=0;
63             success1=0;
64             avail1=0;
65             k2=0;
66             active2=0;
67             success2=0;
68             avail2=0;
69             k3=0;
70             active3=0;
71             success3=0;
72             avail3=0;
73             k4=0;
74             active4=0;
75             success4=0;
76             avail4=0;
77         end
78     end
79 end

```



```

63      success1=0;
64      avail1=0;
65
66      k2=0;
67      active2=0;
68      success2=0;
69      avail2=0;
70
71      k3=0;
72      active3=0;
73      success3=0;
74      avail3=0;
75
76      k4=0;
77      active4=0;
78      success4=0;
79      avail4=0;
80
81      totaltickets=4d15;
82      end
83
84      if(way1==1)          // FIRST USER CODE BEGINS
85          begin
86              k1=~k1;
87          end
88          active1=k1;
89
90      if(active1==1)
91      begin
92          if (timer1==5d16 )    //when the timer1 goes to '16' user1,session will be
93              begin            //completed for user1 and all variables are set to 0 for new user
94                  k1=0;
95                  active1=0;
96                  timer1=0;
97                  avail1=0;
98                  success1=0;
99              end
100          else
101          begin
102              if(ticket11==0 && ticket12==0)
103                  begin
104                      selected1=4d0;
105                  end
106              if(ticket11==0 && ticket12==1)
107                  begin
108                      selected1=4d1;
109                  end
110              if(ticket11==1 && ticket12==0)
111                  begin
112                      selected1=4d2;
113                  end
114              if(ticket11==1 && ticket12==1)
115                  begin
116                      selected1=4d3;
117                  end
118              if(totaltickets>=selected1 && selected1!=4d0)
119                  begin
120                      avail1=1;    //Avail1 will be high if selected number of tickets are available
121                  end
122              else
123              begin
124                  avail1=0;
125              end
126          end
127      end

```



```

126     if(way1==1 && k1==1)           // user entered website , timer starts
127     begin
128         timer1=0;
129         timer1<=timer1+1;
130         success1=0;
131     end
132     if(way1==0 && k1==1)
133     begin
134         timer1<=timer1+1;
135     end
136 end
137
138 if(way1==1 && k1==0)               //payment of user1
139 begin
140
141     if(availl==1)
142     begin
143         success1=1;
144         availl=0;
145         totaltickets=totaltickets-selected1;
146         temp=totaltickets;
147         if(temp<selected2)
148         begin
149             avail2=0;
150         end
151         if(temp<selected3)
152         begin
153             avail3=0;
154         end
155         if(temp<selected4)
156         begin
157             avail4=0;
158         end
159     end
160     else
161     begin
162         availl=0;
163     end
164 end
165 // FIRST USER CODE ENDS
166
167 // SECOND USER CODE BEGINS
168 if(way2==1)
169     begin
170         k2=-k2;
171     end
172     active2=k2;
173
174     if(active2==1)
175     begin
176         if(timer2==5d16)           //when the timer2 goes to '16' user2,session will be
177         begin                       //completed for user2 and all variables are set to 0 for new user
178             k2=0;
179             active2=0;
180             timer2=0;
181             avail2=0;
182             success2=0;
183         end
184     else
185     begin
186         if(ticket21==0 && ticket22==0)
187         begin
188             selected2=4d0;

```

```

188         selected2=4d0;
189     end
190     if(ticket21==0 && ticket22==1)
191     begin
192         selected2=4d1;
193     end
194     if(ticket21==1 && ticket22==0)
195     begin
196         selected2=4d2;
197     end
198     if(ticket21==1 && ticket22==1)
199     begin
200         selected2=4d3;
201     end
202     if(totaltickets>=selected2 && selected2!=4d0)
203     begin
204         avail2=1;        //Avail2 will be high if selected number of tickets are available
205     end
206     else
207     begin
208         avail2=0;
209     end
210     if(way2==1 && k2==1) // user entered website , timer starts
211     begin
212         timer2=0;
213         timer2<=timer2+1;
214         success2=0;
215     end
216     if(way2==0 && k2==1)
217     begin
218         timer2<=timer2+1;
219     end
220     end
221 end
222 if(way2==1 && k2==0) //payment of user2
223 begin
224     if(avail2==1)
225     begin
226         success2=1;
227         avail2=0;
228         totaltickets=totaltickets-selected2;
229         temp=totaltickets;
230         if(temp<selected3)
231         begin
232             avail3=0;
233         end
234         if(temp<selected4)
235         begin
236             avail4=0;
237         end
238     end
239     else
240     begin
241         avail2=0;
242     end
243 end
244 end
245 //2 USER CODE ENDS
246
247 //3 USER CODE BEGINS
248 if(way3==1)

```



```

        avail3=0;
        totaltickets=totaltickets-selected3;
        temp=totaltickets;
        if(temp>selected4)
        begin
            avail4=0;
        end
    end
else
begin
    avail3=0;
end
end
//3 USER CODE ENDS

//4 USER CODE BEGINS
if(way4==1)
begin
    k4=k4-k4;
end
active4=k4;

if(active4==1)
begin
    if(timer4==5d16) //when the timer4 goes to '16' user4,session will be
    begin //completed for user4 and all variables are set to 0 for new user
        k4=0;
        active4=0;
        timer4=0;
        avail4=0;
        success4=0;
    end
else
begin
    if(ticket41==0 && ticket42==0)
    begin
        selected4=4d0;
    end
    if(ticket41==0 && ticket42==1)
    begin
        selected4=4d1;
    end
    if(ticket41==1 && ticket42==0)
    begin
        selected4=4d2;
    end
    if(ticket41==1 && ticket42==1)
    begin
        selected4=4d3;
    end
    if(totaltickets>selected4 && selected4!=4d0)
    begin
        avail4=1; //Avail4 will be high if selected number of tickets are available
    end
else
begin
    avail4=0;
end
end
if(way4==1 && k4==1) // user entered website , timer starts
begin
    timer4=0;

```

```

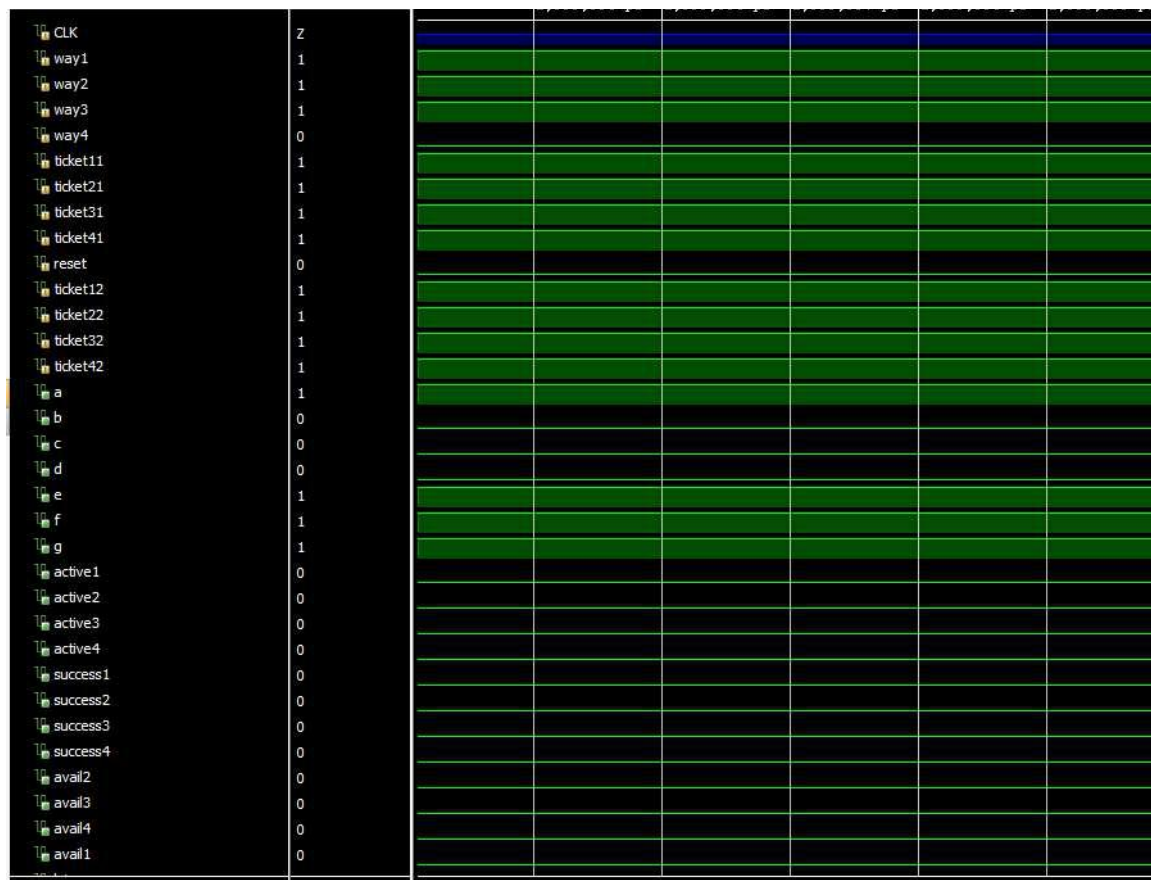
369         timer4<=timer4+1;
370         success4=0;
371     end
372     if(way4==0 && k4==1)
373     begin
374         timer4<=timer4+1;
375     end
376 end
377
378 end
379 if(way4==1 && k4==0) //payment of user4
380 begin
381     if(avail4==1)
382     begin
383         success4=1;
384         avail4=0;
385         totaltickets=totaltickets-selected4
386     end
387     else
388     begin
389         avail4=0;
390     end
391 end
392
393 end
394 //4 USER CODE ENDS
395
396 end
397 else
398 begin
399     counter<=counter+1;
400 end
401 end
402
403 end
404 assign A=totaltickets[3]; //assigning total number of tickets to 4 variables
405 assign B=totaltickets[2]; //to assign it to seven segment display
406 assign C=totaltickets[1];
407 assign D=totaltickets[0];
408
409 assign a = (~B&~D) | (~A&C) | (B&C) | (A&~D) | (~A &B&D) | (A&~B&~C) ; //code for seven segment display
410 assign b = (~B&~D) | (~B&~C) | (~A&~C&~D) | (A&~C&D) | (~A&C&D);
411 assign c = (A&~B) | (~C&D) | (~B&~C) | (~A&D) | (~A&B);
412 assign d = (~A&~B&~D) | (~B&C&D) | (B&~C&D) | (A&~C) | (B&C&~D);
413 assign e = (~B&~D) | (A&B) | (C&~D) | (A&C);
414 assign f = (~C&~D) | (A&~B) | (B&~D) | (A&C) | (~A&B);
415 assign g = (A&~B) | (C&~D) | (A&D) | (~A&B&~C) | (~B&C);
416 endmodule
417

```

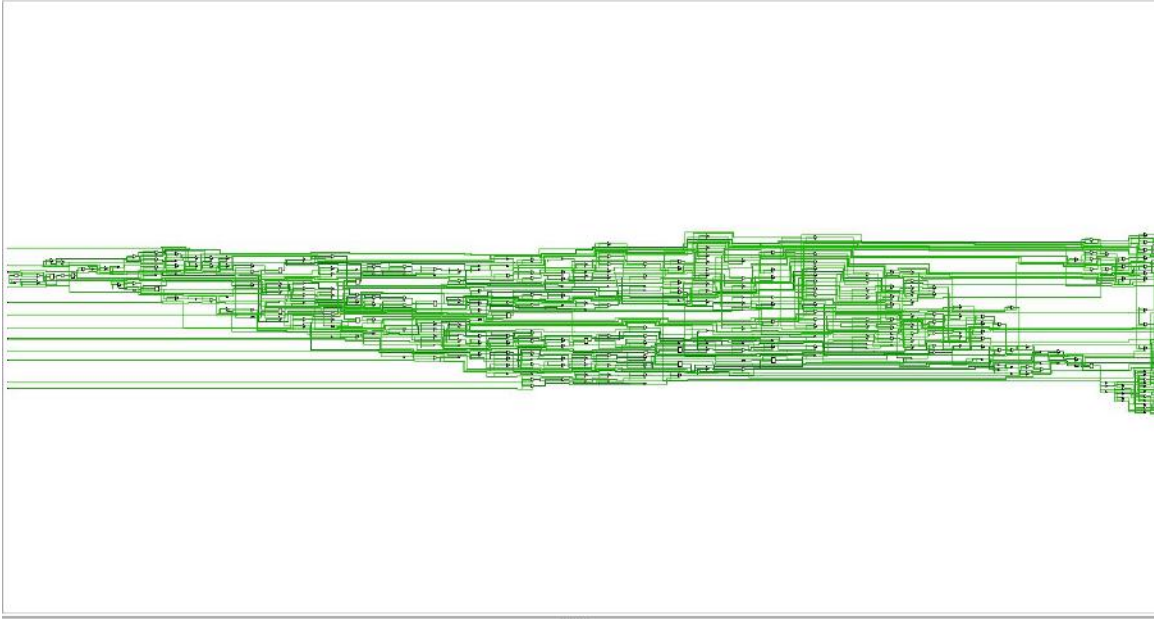
\*The input/output corresponding to each user is as follows.

SL.N	INPUTS		ACTIVITY	OUTPUTS	
	ENTER/ SUCCESSFUL	SELECT TICKETS		AVAILABILITY [LED2]	[LED3]
user1	way1	ticket11 ticket12	active1	avail1	success1
user2	way2	ticket21 ticket22	active2	avail2	success2
user3	way3	ticket31 ticket32	active3	avail3	success3
user4	way4	ticket41 ticket42	active4	avail4	success4

## Simulation



## RTL Schematic



**Constraints:**

\*The **CONSTRAINT** file to implement code on zybo is shown below



```

1 set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports {CLK}];
2
3 set_property -dict {PACKAGE_PIN Y16 IOSTANDARD LVCMOS33 } [get_ports {way1}];
4 set_property -dict {PACKAGE_PIN V16 IOSTANDARD LVCMOS33 } [get_ports {way2}];
5 set_property -dict {PACKAGE_PIN P16 IOSTANDARD LVCMOS33 } [get_ports {way3}];
6 set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports {way4}];
7
8
9 set_property -dict {PACKAGE_PIN N15 IOSTANDARD LVCMOS33 } [get_ports {ticket12}];
10 set_property -dict {PACKAGE_PIN L14 IOSTANDARD LVCMOS33 } [get_ports {ticket11}];
11
12 set_property -dict {PACKAGE_PIN K16 IOSTANDARD LVCMOS33 } [get_ports {ticket22}];
13 set_property -dict {PACKAGE_PIN K14 IOSTANDARD LVCMOS33 } [get_ports {ticket21}];
14
15 set_property -dict {PACKAGE_PIN N16 IOSTANDARD LVCMOS33 } [get_ports {ticket32}];
16 set_property -dict {PACKAGE_PIN L15 IOSTANDARD LVCMOS33 } [get_ports {ticket31}];
17
18 set_property -dict {PACKAGE_PIN J16 IOSTANDARD LVCMOS33 } [get_ports {ticket42}];
19 set_property -dict {PACKAGE_PIN J14 IOSTANDARD LVCMOS33 } [get_ports {ticket41}];
20
21 set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33 } [get_ports {a}]; //JD port
22 set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVCMOS33 } [get_ports {b}];
23 set_property -dict {PACKAGE_PIN P14 IOSTANDARD LVCMOS33 } [get_ports {c}];
24 set_property -dict {PACKAGE_PIN R14 IOSTANDARD LVCMOS33 } [get_ports {d}];
25 set_property -dict {PACKAGE_PIN U14 IOSTANDARD LVCMOS33 } [get_ports {e}];
26 set_property -dict {PACKAGE_PIN U15 IOSTANDARD LVCMOS33 } [get_ports {f}];
27 set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports {g}];
28
29 set_property -dict {PACKAGE_PIN T20 IOSTANDARD LVCMOS33 } [get_ports {active1}]; // JB port 1st row
30 set_property -dict {PACKAGE_PIN U20 IOSTANDARD LVCMOS33 } [get_ports {avail1}];
31 set_property -dict {PACKAGE_PIN V20 IOSTANDARD LVCMOS33 } [get_ports {success1}];
32
33 set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVCMOS33 } [get_ports {active2}]; // JB port 2st row
34 set_property -dict {PACKAGE_PIN Y19 IOSTANDARD LVCMOS33 } [get_ports {avail2}];
35 set_property -dict {PACKAGE_PIN W18 IOSTANDARD LVCMOS33 } [get_ports {success2}];
36
37 set_property -dict {PACKAGE_PIN V15 IOSTANDARD LVCMOS33 } [get_ports {active3}]; // JC port 1st row
38 set_property -dict {PACKAGE_PIN W15 IOSTANDARD LVCMOS33 } [get_ports {avail3}];
39 set_property -dict {PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports {success3}];
40
41 set_property -dict {PACKAGE_PIN W14 IOSTANDARD LVCMOS33 } [get_ports {active4}]; // JC port 4st row
42 set_property -dict {PACKAGE_PIN Y14 IOSTANDARD LVCMOS33 } [get_ports {avail4}];
43 set_property -dict {PACKAGE_PIN T12 IOSTANDARD LVCMOS33 } [get_ports {success4}];
44
45
46 set_property -dict {PACKAGE_PIN G15 IOSTANDARD LVCMOS33 } [get_ports {reset}];

```

## RESULTS:

For each user 3 led lights are provided

LED1 - Shows whether user is active or not

LED2 - Shows whether the selected number of tickets by the user are available or not

LED3 - Confirms the payment of the user

Led Glowing Conditions for each user are as follows.

SL.NO	ACTIVE[LED1]	AVAIL[LED2]	SUCCESS[LED3]
-------	--------------	-------------	---------------

----

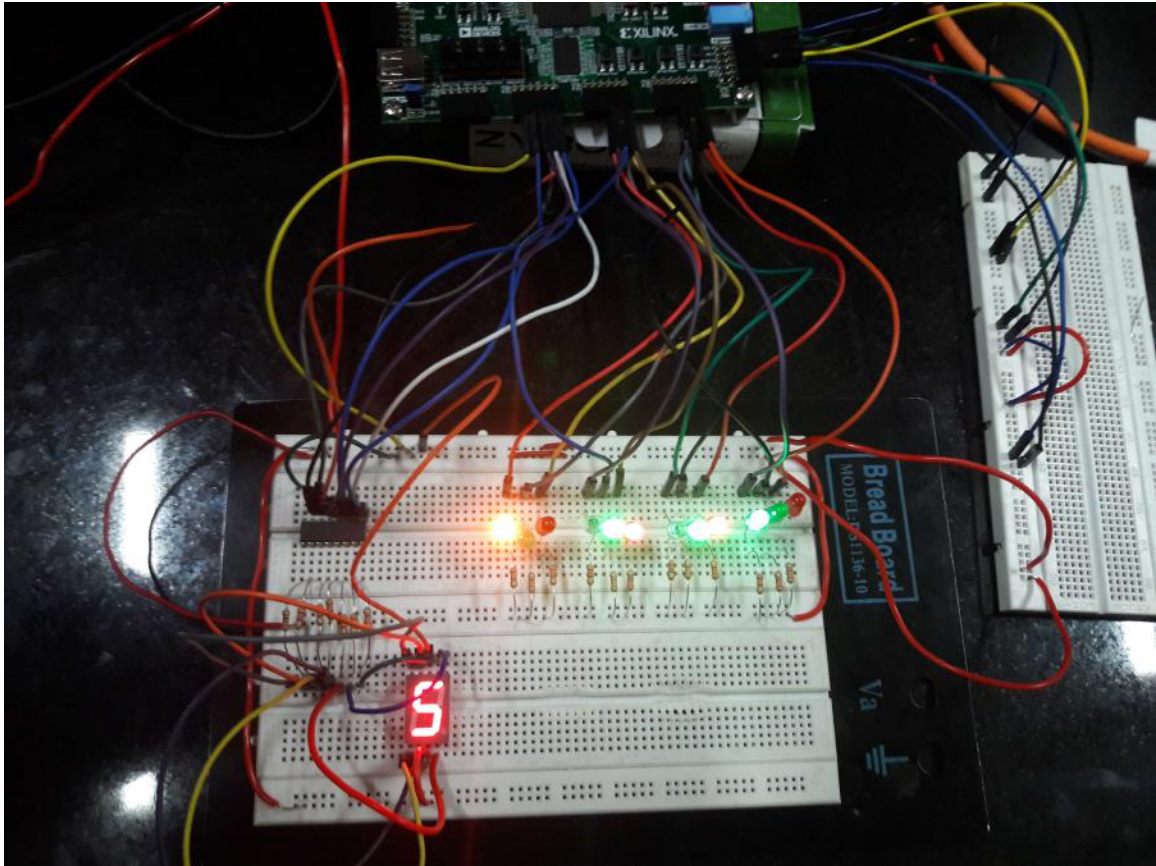
USER1	USER1 IS ACTIVE	INPUT TICKETS OF USER1 ARE AVAILABLE	PAYMENT OF USER1 SUCCESSFULL
USER2	USER2 IS ACTIVE	INPUT TICKETS OF USER2 ARE AVAILABLE	PAYMENT OF USER2 SUCCESSFULL
USER3	USER3 IS ACTIVE	INPUT TICKETS OF USER3 ARE AVAILABLE	PAYMENT OF USER3 SUCCESSFULL
USER4	USER4 IS ACTIVE	INPUT TICKETS OF USER4 ARE AVAILABLE	PAYMENT OF USER4 SUCCESSFULL

-----

\*In this way if one user entered the website and made the payment then another user can enter the website through the same way like this 4 ways are provided in which users come and go irrespective of other 3 ways.

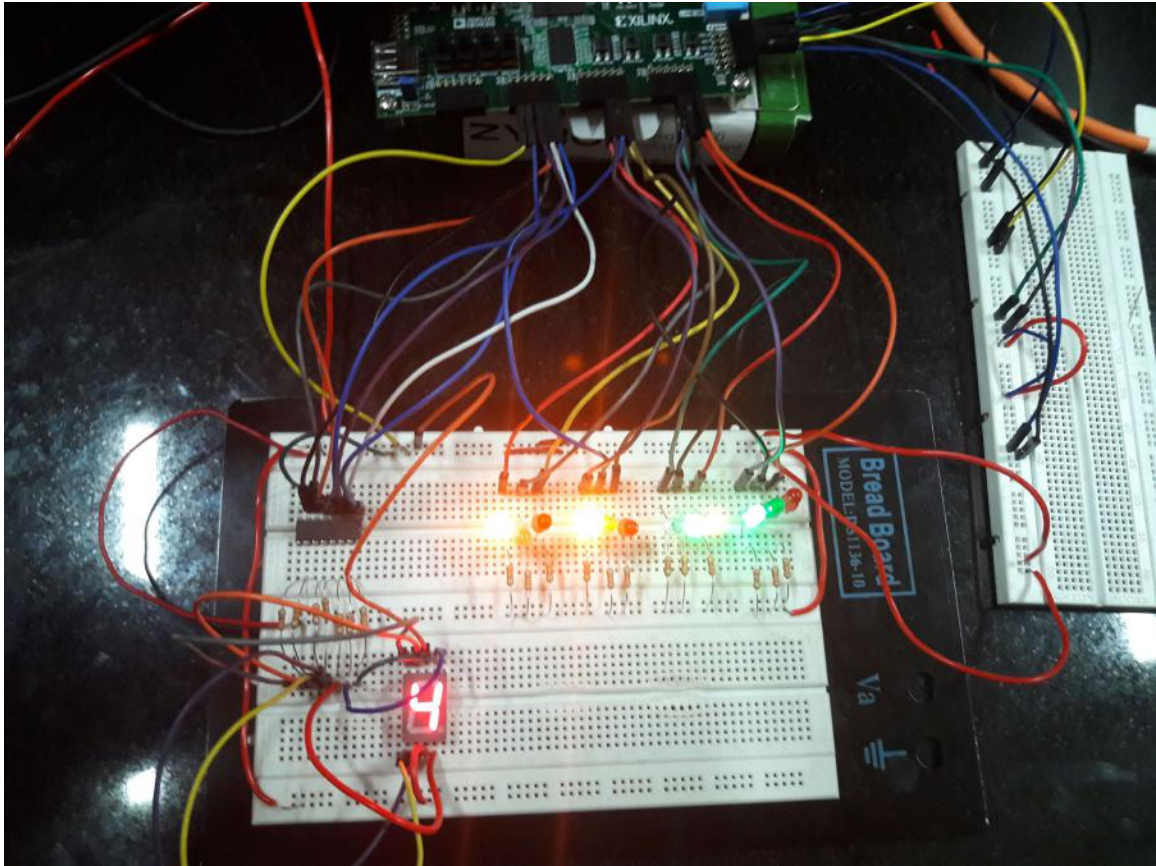
### **CONCLUSIONS:**

Snapshots of Hardware Implementation



\*Here user1,user4 made sucessfull payment so LED3 corresponding to user1,user4 are glowing

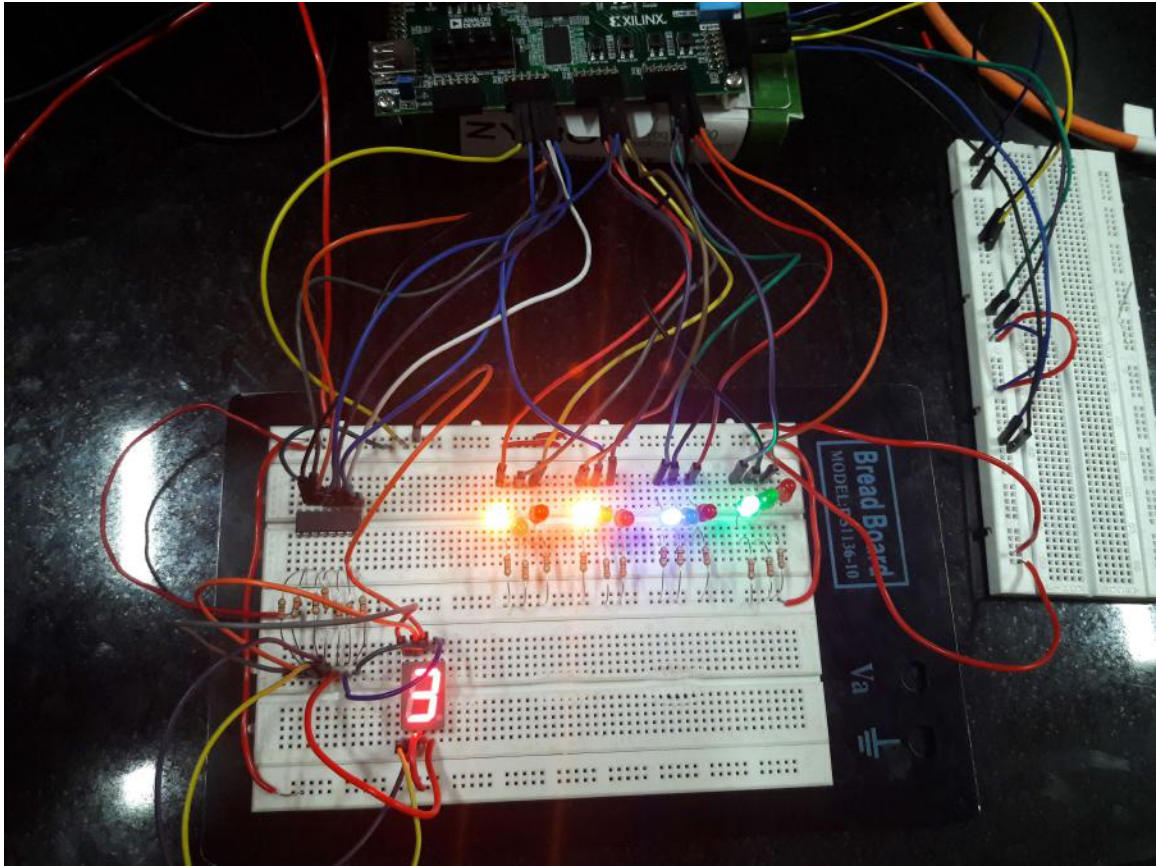
\*user2,user3 are in active state and selected tickets by both the users are so LED1,LED2 of both the users are glowing.



\*Here user1,3,4 made successful payment so LED3 is glowing for user's1,3,4.

\*user2 is in active state and selected number of tickets are available so LED1,2 f user2 are glowing.





\*Here all the user's made sucesful payments so LED3 of all the user's are glowing.

By this way we have implemented ticket reservation system on zybo.

