

Problem Statement:

CAR PARKING SYSTEM:

Design a car parking system such that there is a sensor at the entrance gate, which is activated to detect a vehicle coming. Once the sensor is triggered, a password is requested to open the gate. If the entered password is correct, the gate will open to let the vehicle get in. Otherwise, the gate is still locked. If the current car is getting into the car park being detected by the exit sensor and another car comes, the door will be locked and require the coming car to enter passwords.

CAR PARKING SYSTEM

-DIGIWARE

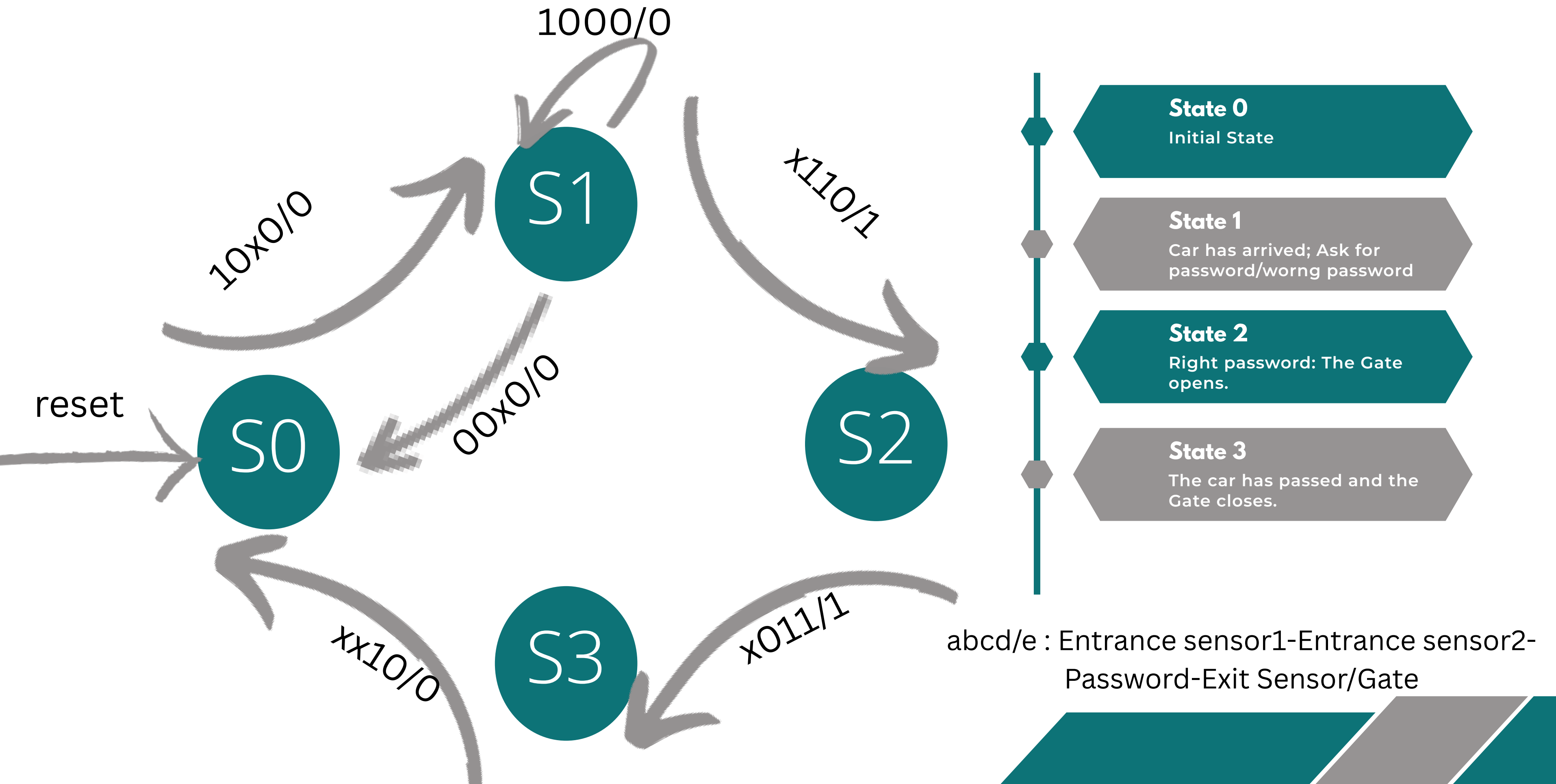
END TERM

PROJECT

PRESENTATION



State Transition Diagram



State Transition Table

STATE 0 = 00

STATE 1 = 01

STATE 2 = 10

STATE 3 = 11

INPUT :
Entrance_Sensor1-
Entrance_Sensor2-Password-
Exit_sensor
OUTPUT:
Gate

RESET	PRESENT STATE	INPUT	NEXT STATE	OUTPUT
1	XX	XXXX	00	0
0	00	10X0	01	0
0	01	00X0	00	0
0	01	X110	10	1
0	01	1000	01	0
0	10	X011	11	1
0	11	X010	00	0

the code

code.v

```
1  module CarParkingSystem (  
2      input entrance1,  
3      input entrance2,  
4      input exit,  
5      input clock,  
6      input reset,  
7      input [3:0] password_input,  
8      output reg gate  
9  );  
10  
11  parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;  
12  
13  reg [1:0] current_state, next_state;  
14  
15  parameter [3:0] correct_password = 4'b1010;  
16  
17  always @(posedge clock or posedge reset) begin  
18      if (reset) begin  
19          current_state <= s0;  
20          gate <= 0;  
21      end  
22  
23      else begin  
24          current_state <= next_state;  
25      end  
26  end  
27
```

```
28  always@(*) begin  
29      case (current_state)  
30  
31          s0: begin  
32              if (entrance1==1 && entrance2==0 && exit==0)  
33                  next_state = s1;  
34              else  
35                  next_state = s0;  
36          end  
37  
38          s1: begin  
39              if (entrance1==1 && password_input==correct_password && entrance2==0 && exit==0)  
40                  next_state = s1;  
41              else if (entrance1==0 && entrance2==0 && exit==0)  
42                  next_state = s0;  
43              else if (password_input==correct_password && entrance2==1 && exit==0)  
44                  next_state = s2;  
45          end  
46  
47          s2: begin  
48              if (password_input==correct_password && entrance2==0 && exit==1)  
49                  next_state = s3;  
50          end  
51  
52          s3: begin  
53              if (password_input==correct_password && entrance2==0 && exit==0)  
54                  next_state = s0;  
55          end  
56  
57          default:  
58              next_state = s0;  
59      endcase  
60  end  
61  
62  always@(next_state) begin  
63      case(next_state)  
64          s2: gate = 1;  
65          s3: gate = 1;  
66          default: gate = 0;  
67      endcase  
68  end  
69  endmodule
```


the testbench

testbench.v

```
1  module CarParkingSystem_TB;
2
3  // Inputs
4  reg entrance1;
5  reg entrance2;
6  reg exit;
7  reg clock;
8  reg reset;
9  reg [3:0] password_input;
10
11 // Outputs
12 wire gate;
13
14 CarParkingSystem uut (
15     .entrance1(entrance1),
16     .entrance2(entrance2),
17     .exit(exit),
18     .clock(clock),
19     .reset(reset),
20     .password_input(password_input),
21     .gate(gate)
22 );
23
24 always begin
25     #5 clock = ~clock;
26 end
27
```

```
28 initial begin
29     $dumpfile("CarParkingSystem.vcd");
30     $dumpvars(0, CarParkingSystem_TB);
31     entrance1 = 0;
32     entrance2 = 0;
33     exit = 0;
34     clock = 0;
35     reset = 0;
36     password_input = 4'b0000;
37
38     //s0
39     #10 entrance1 = 1;//s1
40     #10 entrance1 = 0;//s0
41     #10 entrance1 = 1;//s1
42
43     #10 password_input = 4'b1110; // incorrect password s1
44     #10 password_input = 4'b1010; // correct password s1
45     #10 entrance2 = 1; //s2, gate=1
46     #10 entrance2 = 0; exit = 1; //s3
47     #10 exit = 0; //s0, gate=0
48     //second car waiting s1
49     #10 password_input = 4'b1000; //incorrect password s1
50     #10 password_input = 4'b1010; entrance2=1; //correct password s2, gate=1
51     #10 reset=1;//s0, gate=0
52
53     $monitor("time=%0d, reset=%b, entrance1=%b, entrance2=%b, exit=%b, password_input=%b, gate=");
54
55     #200 $finish;
56
57 end
58
59 endmodule
```

Simulation



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Our Team

**Thogiti
Amar
Sathwik**

**Anany
Dev
Choudhary**

**Gude
Dayana**

**Rachit
Jain**

**Bandaru
Venkata
Sritan**



**THANK
YOU**