Laboratory Report Cover Sheet

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Faculty of Engineering and Technology
Department of Electronics and Communication Engineering

18ECC206J VLSI Design
VI Semester, 2020-2021 (Even Semester)

Name : V Andal Priyadharshini

Register No. : RA1811004010087

Title of the project : Car Parking System

Lab Supervisor : Mrs.K.Ferents Koni Jiavana

Team Members : V Andal Priyadharshini RA1811004010087

Vemu Lakshmi Ananya RA1811004010089

Narayana Vijaya Durga RA1811004010092

Reg. No →		**RA1811004010087**	**RA1811004010089**	**RA1811004010092**
Mark split up ↓	**Maximum Marks**	**Marks obtained**	**Marks obtained**	**Marks obtained**
Novelty in the project work / Abstract	5			
Level of understanding of the design / configuration	10			
Individual Contribution to the project	5			
Report writing	5			
Total	25			

REPORT VERIFICATION

Staff Name :

Signature :

CAR PARKING SYSTEM

OBJECTIVE

To design a secure car parking system using Verilog HDL in Xilinx and simulate the output in ModelSim to simulate the waveform and check our result.

ABSTRACT

In today's days, motor vehicle use is growing day by day, causing noise, traffic congestion, issues with parking spaces, and finding a vacant parking space is becoming increasingly difficult. In this paper we proposed a safe car parking management framework using Verilog HDL. This is about designing an efficient system which takes over the task of identifying free slots in a parking area that keeps parked vehicle records. Parking a vehicle also requires a password. With the rapid increase in the availability and use of cars in recent years, finding a vacant car park is a little complicated. It creates the issue of traffic congestion, emissions (noise and air), as the number of vehicles increases day by day. A FPGA based parking system has been proposed to conquer this problem.

INTRODUCTION

The main goal is systematic parking with protection. Protection requires the use of password when parking, indication of number of available vacancies as well as their locations where only the adjacent vacancies are needed in particular, total number of vacancies available in a specific slot. In order to avoid these problems, design is proposed for secured car parking management system, which will be implemented on FPGA to check vacancies and provide car protection. The FPGA architecture is robust, programmable and can be re-functioned. Our proposed system is designed for FPGA design using Moore State Machine. [3]

SOFTWARE TOOLS REQUIREMENT

EQUIPMENT'S **:** Computer with Xilinx and ModelSim Software Specifications
HP Computer P4 Processor – 2.8 GHz, 2GB RAM, 160 GB Hard Disk

SOFTWARE'S: SYNTHESIS TOOL **:** Xilinx ISE

SIMULATION TOOL **:** ModelSim Simulator

PROBLEM

Car parking system has become one of the biggest problems in city life. Number of vehicles are increasing so that the manual car parking system is not enough. It is difficult to find out a parking place in main cities to park vehicles. Here we proposed a most effective smart car parking system to park a car in a safe and secure way. This was the key impetus in deciding to incorporate the FPGA method. With the support of Xilinx ISE Design Suite, smart car parking system is implemented using Verilog HDL.

CONCEPTS USED

Moore machine, a finite state machine (FSM) in which the output depends only on the present state of the system. The car parking system designed operates under the control of the Finite State Machine (FSM) modelling.

THEORY

This simple project is to implement a car parking system in Verilog. The Verilog code for the car parking system is fully presented. In the entrance of the parking system, there is a sensor 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 would open to let the vehicle get in. Otherwise, the gate is still locked. If the current car is getting in the car park being detected by the exit sensor and another car comes, the door will be locked and requires the coming car to enter passwords. [2]

REALISTIC CONSTRAINSTS:

The intercommunication between the vehicles and the Car Parking System will require a feasible solution and FPGA provides such an interface. As soon as a car enters the lot, a space is reserved for it and the space number flashed on the Display. This would guide the user to the allotted space in the lot. The green light indicates that the password entered is correct and a red light indicates the password entered is wrong and the sensor does not work. It is just an implementation of Finite State Machine of Moore Model.

LOGIC DIAGRAM/STATE DIAGRAM:

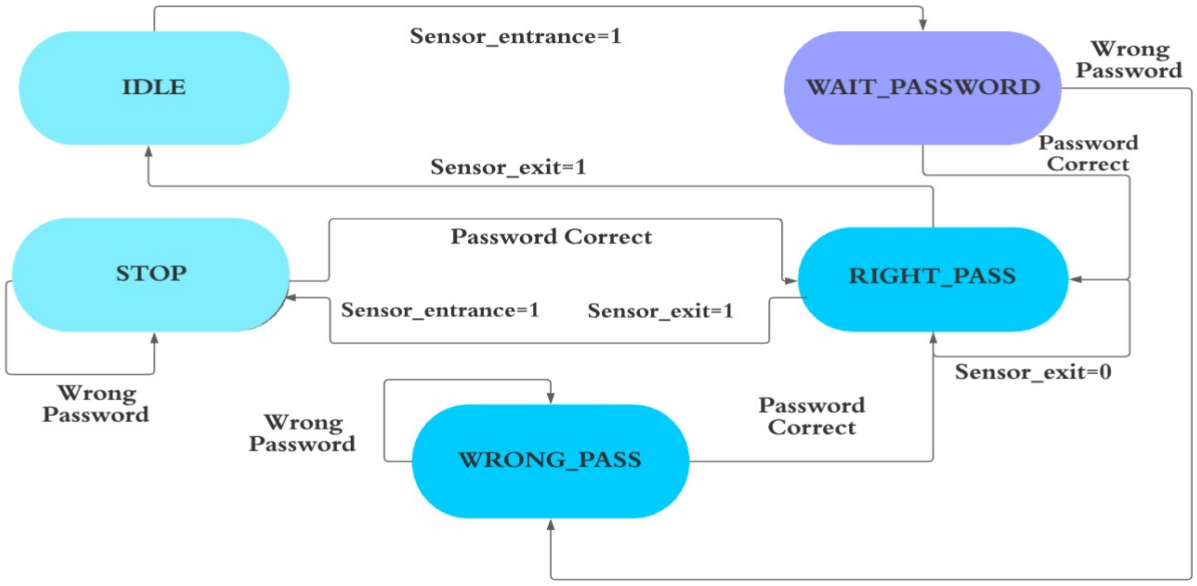


Figure 1: State Diagram

- **IDLE:**

  The system remains in Idle state until the sensor is triggered. When the car approaches the sensor, Sensor_entrance =1 and it waits for the password to be entered. After the right password is entered, Sensor_exit=1 and the system goes back to idle state.

- **WAIT_PASSWORD:**

  When the sensor gets triggered on the approach of a car, the WAIT_PASSWORD state gets activated. If a correct password is entered, it goes into RIGHT_PASS state. If a wrong password is entered, it goes into WRONG_PASS state.

- **RIGHT_PASS:**

  RIGHT_PASS state gets activated when the correct password is entered. It then activates Sensor_exit=1 and moves to the IDLE state.

- **WRONG_PASS:**

  When a wrong password is entered, WRONG_PASS state gets triggered. It will remain in the WRONG_PASS state until a correct password is entered. When the correct password is entered, it moves to the RIGHT_PASS state.

- **STOP:**

  After the activation of RIGHT_PASS state, Sensor_entrance=1 and Sensor_exit=1 will make the program to move to STOP state. It will remain in the stop state as long as a wrong password is entered. When the correct password is entered, it moves to RIGHT_PASS state.

SOURCE CODE:

```
`timescale 1ns / 1ps
module parking_system(
        input clk,reset_n, input sensor_entrance, sensor_exit, input [1:0] password_1, password_2,
        output wire GREEN_LED,RED_LED, output reg [6:0] HEX_1, HEX_2 );
 parameter IDLE = 3'b000, WAIT_PASSWORD = 3'b001, WRONG_PASS = 3'b010,
RIGHT_PASS = 3'b011,STOP = 3'b100;
 // Moore FSM : output just depends on the current state
 reg[2:0] current_state, next_state;
 reg[31:0] counter_wait;
 reg red_tmp,green_tmp;
 // Next state
 always @(posedge clk or negedge reset_n)
 begin
 if(~reset_n)
 current_state = IDLE;
 else
 current_state = next_state;
 end
 // counter_wait
 always @(posedge clk or negedge reset_n)
 begin
 if(~reset_n)
 counter_wait <= 0;
 else if(current_state==WAIT_PASSWORD)
 counter_wait <= counter_wait + 1;
 else
 counter_wait <= 0;
 end
 // change state
 always @(*)
 begin
 case(current_state)
 IDLE: begin
       if(sensor_entrance == 1)
 next_state = WAIT_PASSWORD;
 else
 next_state = IDLE;
 end
 WAIT_PASSWORD: begin
 if(counter_wait <= 3)
 next_state = WAIT_PASSWORD;
 else
 begin
 if((password_1==2'b01)&&(password_2==2'b10))
 next_state = RIGHT_PASS;
 else
 next_state = WRONG_PASS;
 end
 end
 WRONG_PASS: begin
 if((password_1==2'b01)&&(password_2==2'b10))
```

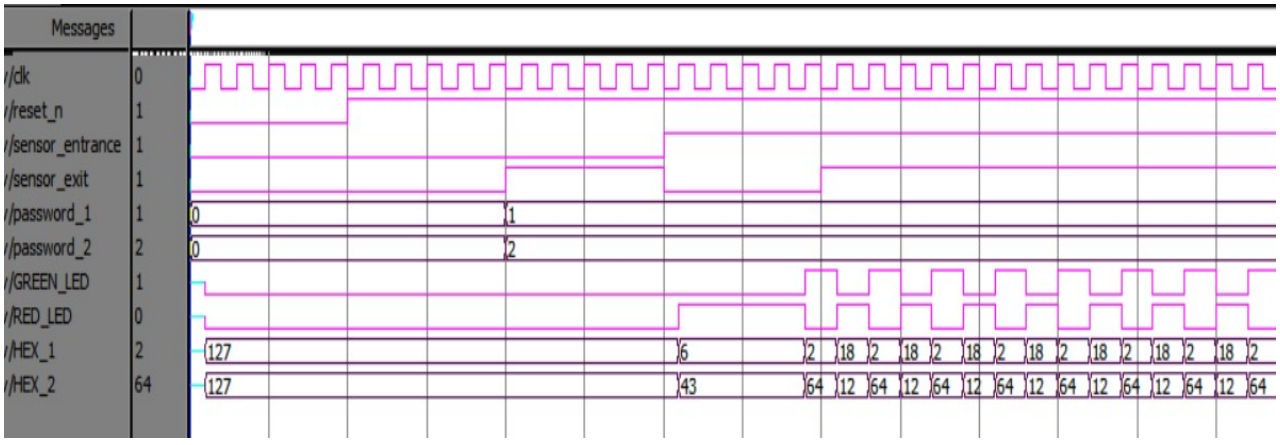
```
next_state = RIGHT_PASS;
else
next_state = WRONG_PASS;
end
RIGHT_PASS: begin
if(sensor_entrance==1 && sensor_exit == 1)
next_state = STOP;
else if(sensor_exit == 1)
next_state = IDLE;
else
next_state = RIGHT_PASS;
end
STOP: begin
if((password_1==2'b01)&&(password_2==2'b10))
next_state = RIGHT_PASS;
else
next_state = STOP;
end
default: next_state = IDLE;
endcase
end
// LEDs and output, change the period of blinking LEDs here
always @(posedge clk) begin
case(current_state)
IDLE: begin
green_tmp = 1'b0;
red_tmp = 1'b0;
HEX_1 = 7'b1111111; // off
HEX_2 = 7'b1111111; // off
end
WAIT_PASSWORD: begin
green_tmp = 1'b0;
red_tmp = 1'b1;
HEX_1 = 7'b000_0110; // E
HEX_2 = 7'b010_1011; // n
end
WRONG_PASS: begin
green_tmp = 1'b0;
red_tmp = ~red_tmp;
HEX_1 = 7'b000_0110; // E
HEX_2 = 7'b000_0110; // E
end
RIGHT_PASS: begin
green_tmp = ~green_tmp;
red_tmp = 1'b0;
HEX_1 = 7'b000_0010; // 6
HEX_2 = 7'b100_0000; // 0
end
STOP: begin
green_tmp = 1'b0;
red_tmp = ~red_tmp;
HEX_1 = 7'b001_0010; // 5
```

```
 HEX_2 = 7'b000_1100; // P
 end
 endcase
 end
 assign RED_LED = red_tmp  ;
 assign GREEN_LED = green_tmp;
endmodule
```

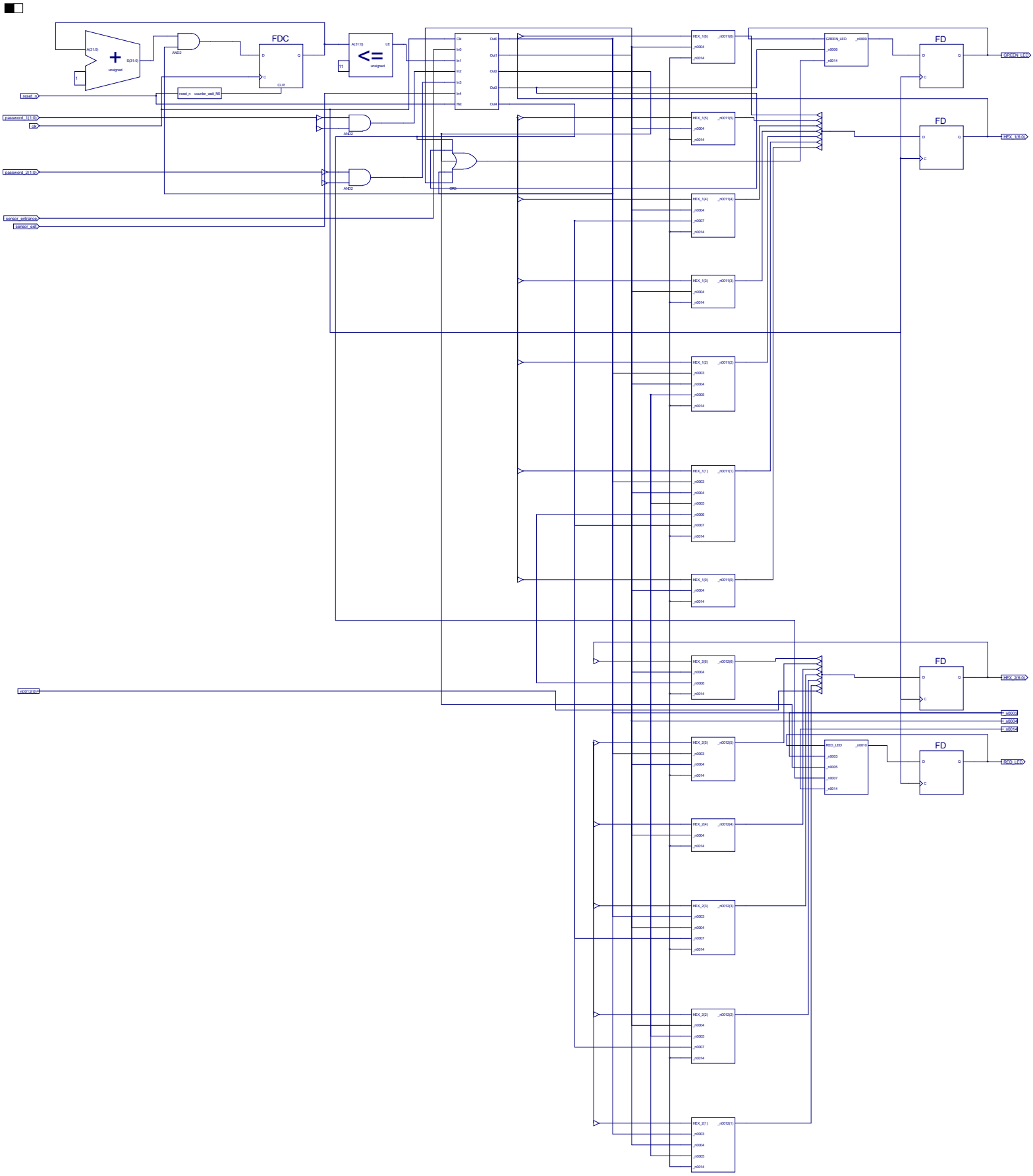
TEST BENCH:

```
`timescale 1ns / 1ps
module tb_parking_system_v;
        //Inputs
        reg clk;          reg reset_n;     reg sensor_entrance;   reg sensor_exit;
        reg [1:0] password_1; reg [1:0] password_2;
        // Outputs
        wire GREEN_LED; wire RED_LED; wire [6:0] HEX_1;     wire [6:0] HEX_2;
        parking_system uut (  .clk(clk),       .reset_n(reset_n),       .sensor_entrance(sensor_entrance),
.sensor_exit(sensor_exit),              .password_1(password_1),              .password_2(password_2),
.GREEN_LED(GREEN_LED), .RED_LED(RED_LED),    .HEX_1(HEX_1),  .HEX_2(HEX_2)  );
        initial begin
                clk = 0;
                forever #10 clk = ~clk;
                end
        initial begin
                reset_n = 0;
                sensor_entrance = 0;   sensor_exit = 0;
                password_1 = 0;  password_2 = 0;     #100;
                reset_n = 1; #100;
                password_1 = 2'b01; password_2 = 2'b10;
                sensor_entrance = 0;   sensor_exit =1;  #100;
                sensor_entrance = 1;  sensor_exit =0; #100;
                sensor_entrance = 1;  sensor_exit =1; #100;
        end
endmodule
```

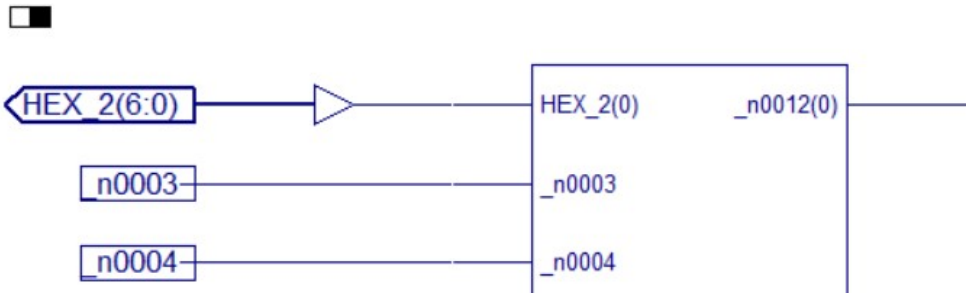
SIMULATION WAVEFORM:



RTL SCHEMATIC (PAGE 1)



RTL SCHEMATIC: (PAGE 2)



RESULT & OBSERVATION:

After thoroughly analysing the algorithms and design features, the programming code was written in Verilog and implemented on Xilinx and ModelSim. An automated Car Parking Management System Model was designed and implemented. In the entrance of the parking system, there is a sensor 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 would open to let the vehicle get in. Otherwise, the gate is still locked. If the current car is getting in the car park being detected by the exit sensor and another car comes, the door will be locked and requires the coming car to enter passwords. [2] Simulation for car Entry and Exit is shown in the output. This program is written in Xilinx ISE and simulated in ModelSim. In real-time implementation using FPGA, if the password entered is correct the green LED glows. If the password entered are incorrect red LED glows.

REFERENCES:

[1] Hodge, A., Humnabadkar, H. and Bidwai, A., 2021. *AUTOMATIC CAR PARKING SYSTEM CIRCUIT USING VERILOG HDL*. [online] Irjet.net. Available at: < https://www.irjet.net/archives/V5/i7/IRJET-V5I7257.pdf > [Accessed 3 May 2021].

[2] System, V., 2021. *Verilog code for Car Parking System*. [online] Fpga4student.com. Available at: < https://www.fpga4student.com/2016/11/verilog-code-for-parking-system-using.html > [Accessed 3 May 2021].

[3] Devi, S., J J R, B., M, D. and A I, K., 2013. *Car Parking System Using FPGA*. [online] www.researchgate.net. Available at: < https://www.researchgate.net/publication/277143568_Intelligent_Car_Parking_Management_System_on_FPGA/citations > [Accessed 3 May 2021].

[4] Kaur, R. and Singh, B., 2013. *DESIGN AND IMPLEMENTATION OF CAR PARKING SYSTEM ON FPGA*. [online] Aircconline.com. Available at: < https://aircconline.com/vlsics/V4N3/4313vlsics07.pdf > [Accessed 3 May 2021].