**System Description**

**For**

**[ CAR GARAGE ]**

**Date: [30/4/2023]**

**Contents**

[**1.** **Abstract** 3](#_Toc133526059)

[**2.** **Overview** 3](#_Toc133526060)

[**3.** **Structure** 4](#_Toc133526061)

[**4.** **Results** 4](#_Toc133526062)

# **Abstract**

Car Garage Project is responsible for counting the number of cars that enter the garage or out the garage, with maximum capacity of 50 cars.

This System displays the number of cars and gives you a warning if the garage is full by displaying “ garage is full ” and turn “garage\_full” to 1.

The system also displays the count of cars by two 7-segment and represents Current state and next state .

# **Overview**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Input/output | Description | Possible output |
| Clk | Input | Pulses or contraction of clock | - |
| Reset | Input | Make the system reset | - |
| Car\_enter | Input | Detect when car enter | - |
| Car\_out | Input | Detect when car exit | - |
| Garage\_full | Output | Show if the garage is full | 1 or 0 |
| Leds1 | Output | display the first digit | 7’bits  First digit |
| Leds2 | Output | display the second digit | 7’bits second digit |
| Car\_count | output | Represents number of cars | 1 to 50 |

System Icon

Car Garage

clk car\_count

reset leds1

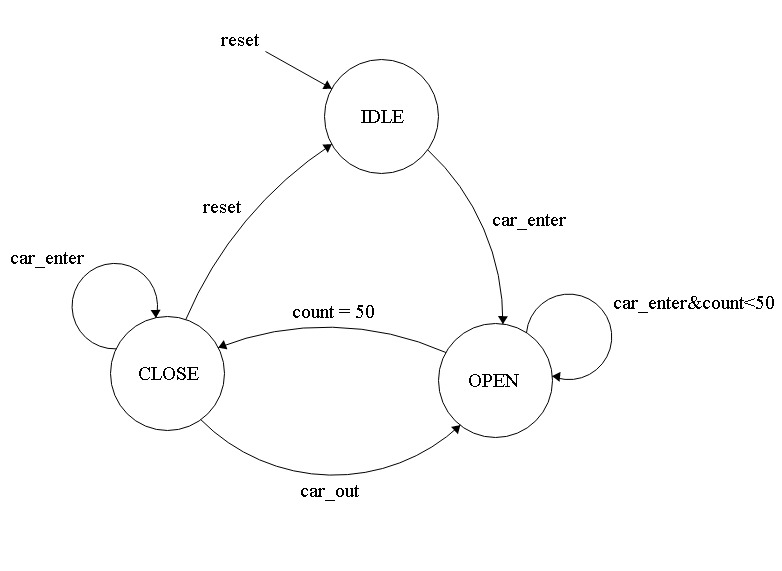
leds2

car\_enter

car\_out garge\_full

# **Structure**

* 1. FSM Diagram



Description of the states:

1. IDEL:

The first and initial state represents by 00.

In this state everything is restart to 0.

When a car enters the system goes to second state and return to this state when reset or is empty.

1. Open:

This state represents the garage is working, and have two cases,

When car is entered this increment the counter, and decreases the counter when car is out.

The system is return to IDEL when reset and goes two the next state when the garage full and return to this state if car out and is working, when car is enter and the garage is not full it remains the state is not change.

1. Close:

This state represents when the system is close and count equal 50.

The state goes the first state when system is reset and goes to OPEN state when car is out.

* 1. Description of Each Module:

1. Counter module:

This module is detecting when a car enters and exit.

When a car is entered increment the counter by 1, with a condition the garage is not full, and decrement the counter when a car exit and is not empty.

In this state when a garage is full, and sensor detects a car.

This warming the car by displaying "GARAGE IS FULL”.

2-decoder module:

This module takes 4 input and produces 7 outputs.

This module is responsible for evaluating the equation of the 7-segment (decoding the inputs), each led of the 7 leds is representing by an equation.

The input is 4 digits of the number we want to decode.

3-counter to decoder module:

This module is taking the count of cars and splits it into two 4’bit digits, and responsible for connection to the decoder module and the output of the decoder module is displays there.

4-car garage module:

this is the main module that connects everything and displays the outputs.

In this module we represent each state and detect when a car is entered , out and counter module is ON there by evaluating the car count.

The counter to decoder module is responsible for displaying the number of cars .

5-DUT module:

In this module we connect every other module to test our system to know every thing is doing well , by set values to our input to test the system.

# **Results**

* 1. Test Strategy

Our test Strategy is reaching the maximum number of cars to check that everything is working well, then decreases it to IDEL state.

|  |  |  |
| --- | --- | --- |
| Input / output | Input value | Output |
| clk | Initial is 0 ,and it changes every (50 ps ) | - |
| reset | 0 then 1 | - |
| Car\_enter | Initial is 0 , then 1 until the car\_count is 50  Then 0. | - |
| Car\_out | First (0) when car\_count is 50.  we change it to (1). | - |
| Car\_count | - | From 1 to 50 |
| Garage\_full | - | 1 or 0 |
| Leds1 | - | 0 to 9 |
| Leds2 | - | 0 to 5 |
| Cur\_state | - | 00 , 01 or 10 |
| Next\_state | - | 00 , 01 or 10 |

* 1. Simulation Results
     1. Car\_count = 50 the first

# 

# 

# **2 – we make car\_out =1 ,and car\_in = 0 ,**

# **until car\_count =0**

# 

****