# Small-Project [Midterm]

Most of the time, car garage is full, and the absence of evidence of that makes things sometimes worse. Therefore, the aim of the project is to help eliminate congestion .The detailed Specification of CAR SYSTEM is as follow:

- there's a button to open and close garage
- there's a counter that increment when car enter garage and decrement when car leave garage
- the maximum number of cars can enter garage is 50
- if the existing cars greater than 50 then garage will not open and display GARAGE IS FULL otherwise car enter garage.
- when garage not full display the count of cars otherwise display GARAGE IS FULL

**The design team has met few times to decide on the following for the CAR GARAGE architecture:**

• Although the system could be modeled as one big FSM, it was decided to decompose the system into smaller FSMs. Design reuse is also recommended. That is, one small FSM model can be repeated multiple times in the overall system model.

• An n-bit up counter is to be used to generate number of cars in garage. The maximum car number value will be (50), with a default value of 0.

**• There will be an internal system clock.**

**In this project, you are going to model the operation of CAR GARAGE and verify it via simulation. Here is the list of deliverables:**

a) In a table, identify CAR GARAGE inputs and outputs and briefly describe their meaning and possible values.

b) Draw an icon for the CAR GARAGE, clearly showing its input and output signals.

c) Draw a block diagram showing the CAR GARAGE structure.

d) Draw the necessary FSM diagram(s).

e) Model the up counter as a separate component using a Verilog behavioral description.

f) Model the required FSM(s) in Verilog.

g) Model CAR GARAGE using a mixed architecture Verilog code. Use concurrent descriptions for the binary flags. The model should issue an alarm if the CAR GARAGE is FULL and display the count of cars existing if not full.

h) In a table, propose a test strategy to verify the operation of the CAR GARAGE model. Carefully select an appropriate set of test cases that test various design aspects.

I) Design a testbench to verify the operation of the CAR GARAGE model using the test strategy proposed in the previous point. Provide simulation result snapshots.

## Here are some hints:

- To apply **hierarchy and modularity concepts**, Try to decompose the CAR GARAGE architecture into a number of small procedures. Each procedure should be responsible for updating some of the output signals. Avoid having multiple procedures updating one single signal.
- Adjust the **Clock** of the system to be 10 milliseconds.

## Here are some Tips:

- In industry, typically the team is divided between designing and testing sub-teams; they put the design document and agree on the interfaces. Then, they start to work simultaneously to speed up the delivery. So, it is recommended to 1– Create a design document before coding. 2– Two team members work on the design development, while the other member works on the testbench development. 3– Once both sub-teams finish, they integrate the design and testbench and verify the system's functionality.

## Here are delivery and evaluation rules:

| | |
|---|---|
| The project runs on ModelSim + Report. | Full Mark |
| Errors in ModelSim Simulator + Report | 0.8 x Full Mark |
| Cheating | Zeros to all members of the Group. |

Good Luck

Computer Architecture Team