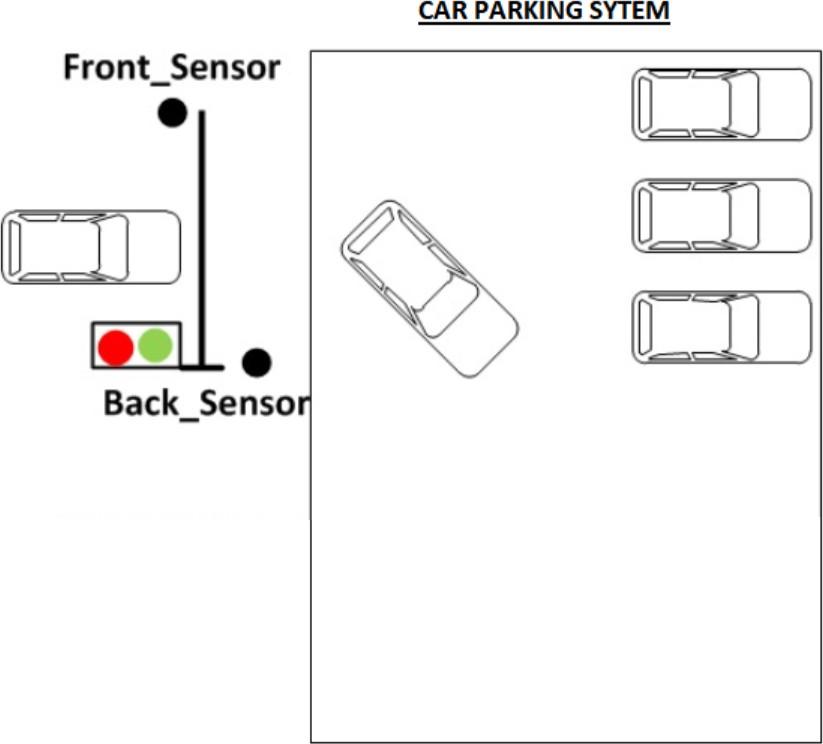
**GateWiz**

Due date: 19 Jan, 2024

Due time: 11:59 PM

Car Parking System (For group 1)



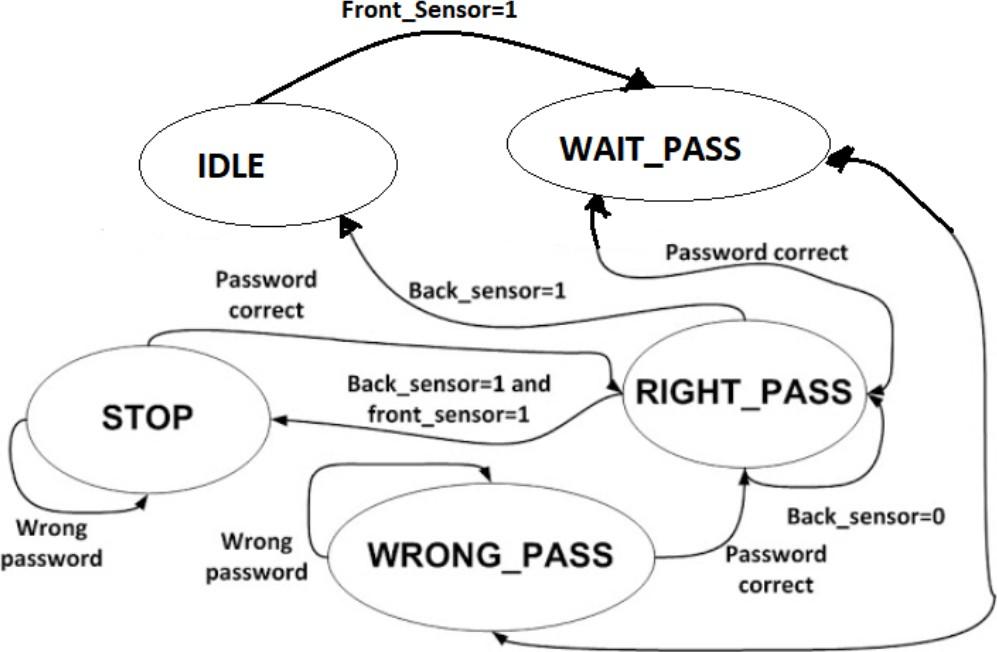


Imagine designing a futuristic Verilog code for a smart car parking system! At the entrance of this advanced parking lot, we have a high-tech sensor eagerly waiting to detect the arrival of a vehicle. Once the sensor spots a car, it triggers a magical sequence – the system prompts the driver for a secret password to unveil the gateway to parking paradise!

Now, here's where the Verilog wizardry begins. Picture a digital guardian that checks if the entered password aligns with the mystical correct password. If it's a match, voila! The gates swing open majestically, inviting the car to gracefully glide into its designated spot.

But, ah, the plot thickens! If, by any chance, the system senses that there's already a car enjoying the VIP parking experience inside, it doesn't panic. Instead, it calmly locks the gates and shifts into a mode where it challenges the next approaching car to present the correct password for entry.

Here is the flow chart of how the Car Parking system works!





* Car Parking System

module CarParkingSystem(

input front\_sensor,

input back\_sensor,

input clock,

input reset,

input [1:0] password,

output wire green\_LED,

output wire red\_LED,

output reg[3:0] display\_screen

);

Inputs:

1. clock:

Role: Clock Signal.

Function: This input is typically connected to the clock signal of the system. It serves as a reference for the timing of state transitions and other operations within the module. The posedge clock\_in condition in the always blocks indicates that certain operations should occur on the rising edge of the clock.

2. reset:

Role: Reset Signal.

Function: This input is used to reset the module. When the reset signal is active (low or asynchronous), the module initializes or resets its internal state and counters.

3. front\_sensor:

Role: Front Sensor Input.

Function: This input represents the state of the front sensor, indicating whether there is a car in front of the parking space. It is used to trigger state transitions when a car enters the parking system.

4. back\_sensor:

Role: Back Sensor Input.

Function: Similar to Front\_Sensor, this input represents the state of the back sensor, indicating whether there is a car behind the parking space. It is used to trigger state transitions when a car is parked or leaves the parking system.

5. password:

Role: Password Inputs.

Function: These inputs represent the binary components of a password. They are used in the WAIT\_PASSWORD state to check if the entered password matches the expected value.

Outputs:

1. green\_LED:

Role: Green LED Output.

Function: This output represents the state of the green LED. Its value is controlled based on the current state of the system. For example, it blinks or remains steady in certain states, such as when the gate opens.

2. red\_LED:

Role: Red LED Output.

Function: Similar to G\_LED, this output represents the state of the red LED. Its value is controlled based on the current state of the system. For example, it blinks or remains steady in certain states, such as when the gate is closed.

3. display\_screen:

Role: 4-Segment Display Outputs.

Function: These outputs represent the values to be displayed on two 4-segment displays. The specific values are determined based on the current state of the system. For example, it displays different characters or numbers in different states.