Convert the circuit to only NAND and NOR gates.

A

B

C

D

E

F

G

H

I

Output 1

Output 2
Please describe basic functionality of this circuit and convert it into Verilog module.
A system with three inputs will output true if at least two inputs are true.

This system also has a "test" switch that causes the system to output true if any one input is true, and the test switch is true.

Provide the truth table, K-map, and circuit design of this system.
Given the following Boolean eq., find all the errors the student made when attempting to simplify.

\[ F = (A + AB + BC + BC) \left( \overline{A + BC + CD} \right) \]

\[ F = (A + B + BC) \left( \overline{A + BC + CD} \right) \]

\[ F = (A + B + BC)(A \land BC \land C + D) \]

\[ F = (A + B + BC)(C \land (A \land B + D)) \]

\[ F = (AC + B + BC + BC) (AB + D) \]

\[ F = (AC + BC + BC) (AB + D) \]

\[ F = (AC + BC) (AB + D) \]

\[ F = ABC + ACD + ABC + BCD \]

\[ F = ABC + ACD + BCD \]
Draw the State Diagram and State Table for a string recognizer whose output will be true if last 5 input values are "11111".
Both flip-flops have been reset so their present states are 0. How must $A$ change ("0" to "1" or "1" to "0"), if at all, to make Output TRUE?
Using Boolean Algebra rules, Minimize the function.

\[ F = (A + \overline{B} + \overline{B}D) (ABC + \overline{A}D + C) \]
Write the Boolean Eqn.
For the following K-Map
Parking Garage Gate
Create the state diagram for a gate at a parking garage that only accepts credit cards. If someone takes a ticket on the “enter side”, the gate should lift to allow the car to enter, and then lower after a period of T-open (TO). Once a valid card is used to pay on the exit side, the gate should lift to allow the car to leave and then lower after a period of TO. If the card is invalid, then a “Card declined” light should come on and the gate should not lift. Assume there will never be a car at both the enter and exit side at the same time.
For the circuit below, show the output “F” wave form. Gate delays are all 5ns
Given a state table, implement this circuit with D flip-flops, and draw the state diagram.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Input</th>
<th>Output</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1, P0</td>
<td></td>
<td></td>
<td>N1, N0</td>
</tr>
<tr>
<td>0, 0</td>
<td>0</td>
<td>1</td>
<td>0, 0</td>
</tr>
<tr>
<td>0, 1</td>
<td>0</td>
<td>0</td>
<td>0, 0</td>
</tr>
<tr>
<td>1, 0</td>
<td>0</td>
<td>0</td>
<td>0, 1</td>
</tr>
<tr>
<td>1, 1</td>
<td>0</td>
<td>X</td>
<td>X, X, X</td>
</tr>
</tbody>
</table>

The output is 1 for states 0, 0 and 0, 1, and 0 for states 1, 0 and 1, 1.
There are 3 indicator lights in Red, Yellow, and Green to show the working condition of three devices. If all of the devices work properly, then the green lights on. If one of them goes wrong, the yellow lights on. If two of them goes wrong, the red lights on. If all of them goes wrong, the red light and yellow lights turn on together.

Write the boolean equations and simplify for three lights and draw the circuit diagram.
Write the Verilog Code for the following state diagram: