

0.1 Experiment 1 (Analysis and Implementation of a Problem Definition)

0.1.1 Aim

In this experiment, you will implement a circuit design of a solution to a problem.

0.1.2 Problems

A toy car with remote control has an interesting warning light (y) on its remote. We know that the state of this light (1: on, 0: off) is controlled by three factors:

x_0 : Battery-low signal of the toy car (1: Low, 0: High).

x_1 : Distance of the car to the remote (1: Far, 0: Near).

x_2 : Autopilot (1: Activated, 0: Deactivated).

You will create and implement the circuit design of the warning light y in terms of its inputs x_0 , x_1 , and x_2 . Following observations were recorded to guide you in this process:

- If the autopilot is deactivated, the toy car is far from the remote, and the battery is high, then the warning light is on.
- Else if the autopilot is activated and the battery is high, then the warning light is off.
- Else if the car is far from the remote, then the warning light is on.
- Else if the battery is low, then the warning light is on.
- Otherwise, the warning light is off.

You have only 2xAND, 2xOR, and 4xNOT gates at your disposal. The AND and OR gates are binary, i.e. they take exactly two inputs. You are expected to design the circuit which uses the fewest number of gates as possible. Designs with the fewest number of gates will get the highest grades.

During Verilog implementation, you have to use the template source codes we provide.

0.1.3 Preliminary Work

Before the experiment, you should prepare the following materials:

1. Fill the truth table given in the template according to the design requirements.
2. Write the sum of products (SOP) expression for the truth table.

3. Find the minimized SOP expression by using Boolean theorems and axioms (show your work clearly, minimize the equation you previously found in step 2 using the laws used in class, using exactly one law per line). The minimized expression must also be in SOP form.
4. Write the product of sums (POS) expression for the truth table.
5. Find the minimized POS expression by using Boolean theorems and axioms (show your work clearly, minimize the equation you previously found in step 4 using the laws used in class, using exactly one law per line). The minimized expression must also be in POS form.
6. Draw the circuit for the minimized expression using only binary (which means it has two inputs) AND, OR, and NOT gates.
7. Fill the **gate** level verilog code for the circuit chosen at the previous step. You can use the built-in AND, OR, and NOT gates.
8. Write the code for the testbench in order to test all possible input combinations.
9. Verify that the outputs of your implementation is the same as the truth table.

Then, submit the your code, and your report under the name

<StudentID1>_<StudentID2>_PRE1.zip through Moodle. One submission per group is required.