## 6.2 Experiment 2 (Analysis and Implementation of a Problem)

### 6.2.1 Aim

In this experiment, you will design and implement a circuit by analysing the given problem. Please refer to Chapter 4 of the book for more details.

### 6.2.2 Problem

A robot which plays soccer has two different behavior modes($b$): defensive or offensive. In the defensive mode, the robot positions itself in its own halfcourt and supports other team members. In the offensive mode, the robot acts in a way to gain ball control and score a goal. We know that the behavior mode of the robot($b$) (1:defensive, 0:offensive) is determined by three factors:

$i_0$: Distance of the robot to the ball (1: far, 0: near).

$i_1$: If the golie of the opponent team is penalized (1: not penalized, 0: penalized).

$i_2$: If the total number of offensive robots in the team is larger than 2 (1: (total number>2) , 0: (total number<=2)).

You will create and implement the circuit design of the behavior mode $b$ in terms of its inputs $i_0$, $i_1$, and $i_2$. The situation of the game described by the above three factors determines the behavior mode of the robot as follows;

- If the total number of offensive robots is larger than 2, and the goalie of the opponent team is penalized, and the ball is far, then the behavior is defensive.
- Else if the ball is far, and the goalie of the opponent team is penalized, then the behavior is offensive.
- Else if the ball is far, then the behavior is offensive.
- Else if the total number of offensive robots is larger than 2, or the goalie of the opponent team is penalized, then the behavior is defensive.
- Otherwise, the behavior is defensive.

You have only 2 AND, 1 OR, and 3 NOT 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.

### 6.2.3 Preliminary Work

You should prepare the following materials for the preliminary work:

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. Write 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.

Submit your code and report considering the submission rules described in Section 3.1. One submission per group is required.

### 6.2.4 Lab Work

1. Repeat all the procedures done in the preliminary work for the problem given in the lab section. Fill the report, write the Verilog code, and demonstrate your implementation to the assistants.