ELEN 21 Lab 3: Two Level Circuit Design
Pre-Lab

Tamir Enkhjargal

October 2018

II. Pre-Lab

Problem Statement:
You will design a highway entrance ramp metering controller with the following specifications for controlling the release of traffic from an entrance ramp:

There are three metered entrance lanes: one carpool lane and two others for cars that are not carpool. Each lane has its own light (red = stop and green = go) and its own sensor to detect a waiting car.

The carpool lane is given priority for a green light over the other two lanes. Otherwise, a "round robin" scheme is used in which the green lights alternate between the left and right lanes.

- If there is a car in the car pool lane, CL = 1
- If there are no cars in CL or RL, but there is a car in LL, LL = 1
- If there are no cars in CL or LL, but there is a car in RL, RL = 1
- If there are no cars in CL, LL, or RL, then RL = 1
- If there are no cars in CL, but there are cars in LL and RL and RR = 0, then LL = 1
- If there are no cars in CL, but there are cars in LL and RL and RR = 1, then RL = 1
- If any of CL, LL, or RL is not specified to be 1 in the conditions above, then it has a value of 0.

The Design Process:

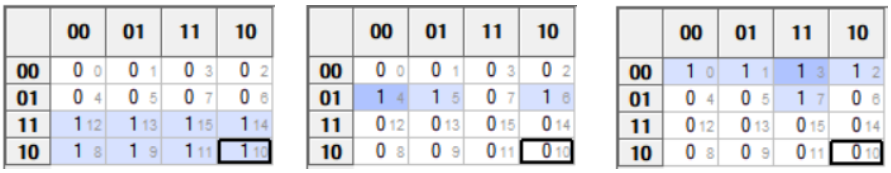
Identify the inputs and outputs for the circuit

- Inputs: CS, LS, RS, RR
- Outputs: CL, LL, RL

From the problem statement, write the truth table and the algebraic expressions for the traffic controller outputs:

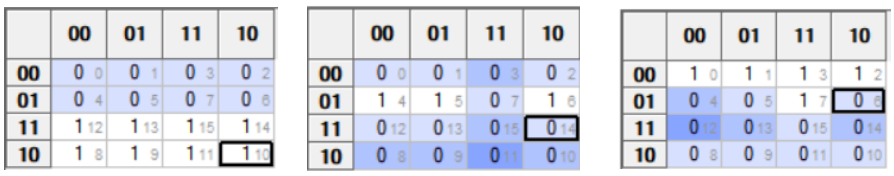
CS	LS	RS	RR	CL	LL	RL
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

Table 1: Truth Table of the Traffic Controller



(a) K-map of CL (b) K-map of LL (c) K-map of RL

Figure 1: Sum of Products K-maps of the three functions



(a) K-map of CL (b) K-map of LL (c) K-map of RL

Figure 2: Product of Sums K-maps of the three functions

The minimized SOP implementation of the three functions are:

$$\sum SOP_{CL} = CS$$

$$\sum SOP_{LL} = \overline{CS} \cdot LS \cdot \overline{RS} + \overline{CS} \cdot LS \cdot \overline{RR}$$

$$\sum SOP_{RL} = \overline{CS} \cdot RS \cdot RR + \overline{CS} \cdot \overline{LS}$$

The minimized POS implementation of the three functions are:

$$\sum POS_{CL} = CS$$

$$\sum POS_{LL} = (\overline{RS} + \overline{RR}) \cdot (LS) \cdot (\overline{CS})$$

$$\sum POS_{RL} = (\overline{LS} + RS) \cdot (\overline{LS} + RR) \cdot (\overline{CS})$$

Compare the cost of the POS form and the SOP form when the cost is measured by the sum of the number of logic gates and the number of inputs to all logic gates:

Just a numerical overview:

SOP needs 0, 3, and 3 gates, for the three functions, respectively (not including INV/NOT gates). SOP also needs 1, 6, and 5 inputs to all gates, for the three functions, respectively.

POS needs 0, 3, and 4 gates, for the three functions, respectively (not including INV/NOT gates). POS also needs 1, 4, and 5 inputs to all gates, for the three functions, respectively.

We can see that POS needs one more gate for the RL function output than the SOP cost (2 ORs and 3 ANDs vs. 3 ANDs and 1 OR).

However, the amount of inputs needed for the LL function output needs 2 less inputs for the POS than the SOP costs (4 total/4 unique vs. 6 total inputs/4 unique).

There isn't really too big of a difference in the costs between the SOP and POS minimal functions, this is because we are analyzing 16 maximums (split into 3 functions) and 16 minimums (split into 3 functions).