Fundamentals of Computer Engineering Laboratory
ECE284

Combinational Logic Design
Report for Week # 2

By: Full Name
Partner(s): Full Name

Instructor's Name:

Section: Friday 9 am to 11:50 am
Date:

Honor Pledge:

"I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community, it is my responsibility to turn in all suspected violators of the Honor System. I will report to an Honor Council hearing if summoned."

Signature: ______________________

Introduction:

In this lab, a combinational logic circuit to implement a car alarm has been designed, implemented, and tested. The requirements for the car alarm have been provided in the lab description and are summarized below. This lab involves going through the different processes of logic design to transform a word problem into a digital circuit. In the following report, the process of logic modeling and transformation of a logic expression into a digital logic circuit is presented.

Preliminary Work:

From the lab problem statement, the car alarm must sound under three specific conditions. First, the car alarm should sound if the key is in the ignition and the door is open. Second, the alarm should sound if the key is in the ignition and the seat belts are not on. Third, the alarm should sound if the key is not in the ignition and the lights are on. From these conditions, we note that the alarm system has four inputs for detecting the key, door status, lights, and seat belts. Clearly, the alarm also has one output: the signal that turns on the alarm signal. For this lab, we make the following signal definitions used in the design:

Alarm Signal	≡ **Alarm**	**Alarm** =1	=> Alarm sounds
Key	≡ **K**	**K** =1	=> Key in ignition
Door	≡ **D**	**D** =1	=> door is open
Lights	≡ **L**	**L** =1	=> lights are on
Seat Belts	≡ **S**	**S** =1	=> seat belts are fastened

From review of the problem statement, the **Alarm** must sound if

- the key is in the ignition (**K** =1) AND the door is open (**D** =1)

OR

- the key is in the ignition (**K** =1) AND the seat belts are not on (**S** =0)

OR

- the key is not in the ignition (**K** =0) AND the lights are on (**L** =1)

We now have enough information to construct the truth table for the **Alarm** function. The following is the 4 input, 1 output truth table for the **Alarm** function:

K	**D**	**L**	**S**	**Alarm**
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Each one in the truth table corresponds to one minterm. The canonical SOP form for the **Alarm** function is the sum of the ten minterms in the function. The minterm expression for the **Alarm** function is

$$\mathbf{Alarm} = \overline{\mathbf{K}}\,\overline{\mathbf{D}}\mathbf{L}\overline{\mathbf{S}} + \overline{\mathbf{K}}\,\overline{\mathbf{D}}\mathbf{LS} + \overline{\mathbf{K}}\mathbf{DL}\overline{\mathbf{S}} + \overline{\mathbf{K}}\mathbf{DLS} + \mathbf{K}\overline{\mathbf{D}}\,\overline{\mathbf{L}}\,\overline{\mathbf{S}} + \mathbf{K}\overline{\mathbf{D}}\mathbf{L}\overline{\mathbf{S}} + \mathbf{KD}\overline{\mathbf{L}}\,\overline{\mathbf{S}} + \mathbf{KD}\overline{\mathbf{L}}\mathbf{S} + \mathbf{KDL}\overline{\mathbf{S}} + \mathbf{KDLS}$$

Again, from the problem, the **Alarm** must sound if

- the key is in the ignition AND the door is open => $\mathbf{K} \cdot \mathbf{D}$

OR

- the key is in the ignition AND the seat belts are not on => $\mathbf{K \cdot \overline{S}}$

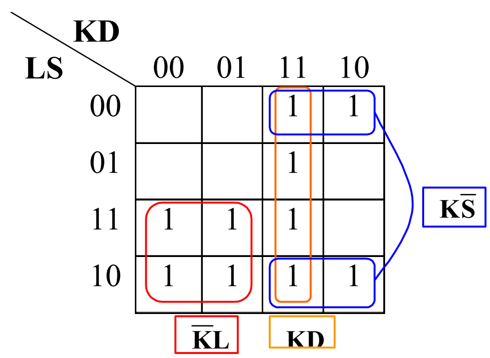
OR

- the key is not in the ignition AND the lights are on => $\mathbf{\overline{K} \cdot L}$

Thus, the SOP expression can be written

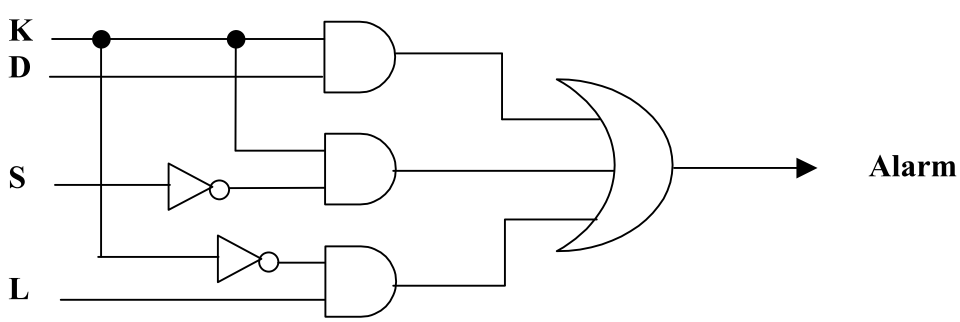
$$\mathbf{Alarm = K \cdot D + K \cdot \overline{S} + \overline{K} \cdot L}$$

Using a Karnaugh map (or K-map), we can verify that this is the minimal SOP expression.

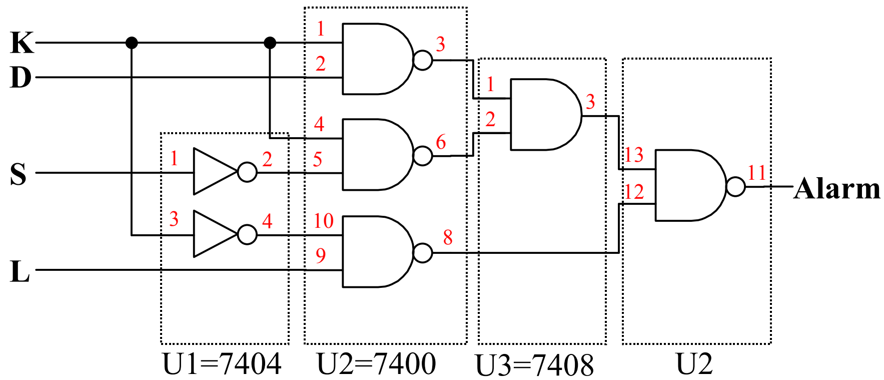


Circuit Diagram and Wire List

The following circuit can be derived from the minimized SOP expression for the alarm function.



Since there are no 3-input OR gates in our kit, we need to modify the circuit in order to implement. Instead, we will use the NAND-NAND implementation for SOP circuits. Because the kits do not have three input NAND gates, we create a 3-input NAND gate using a 2-input AND gate and a 2-input NAND gate from our kits. The following circuit diagram includes these modifications. In addition, the circuit diagram also includes chip designations (U1, U2, and U3) as well as the specific pin numbers to connect these chips. While not explicitly shown in logic circuits, power and ground must also be connected to all chips.



Wire List:

K → SW1 → U1 - 3, U2 - 1, U2 - 4

D → SW2 → U2 - 2

L → SW3 → U2 - 9

S → SW4 → U1 - 1

U1 - 2 → U2 - 5

U1 - 4 → U2 - 10

U2 - 3 → U3 - 1

U2 - 6 → U3 - 2

U2 - 8 → U2 - 12

U3 - 3 → U2 - 13

U2 - 11 → LED

Vcc → U1 - 14, U2 - 14, U3 - 14

Gnd → U1 - 7, U2 - 7, U3 - 7

Lab Results:

The circuit was built and tested and the following truth table was obtained in the lab.

K	D	L	S	Alarm (observed)	Alarm (corrected)
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	0	0
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Observations and Conclusions:

The logical behavior of the circuit was verified as shown in the output table in the lab results section. The **Alarm** signal was true (high) whenever **K** =**D** = 1 or **K** =1 and **S**=0 or when **K**=0 and **L**=1.

Circuit Modifications:

The output of U1 (pin 2) was found to always be low. Suspecting the associated inverter to be faulty, the connections to U1 - 3 and U1 - 4, were changed to U1 - 9 and U1 - 8, respectively, utilizing a spare inverter on the 7404 chip.