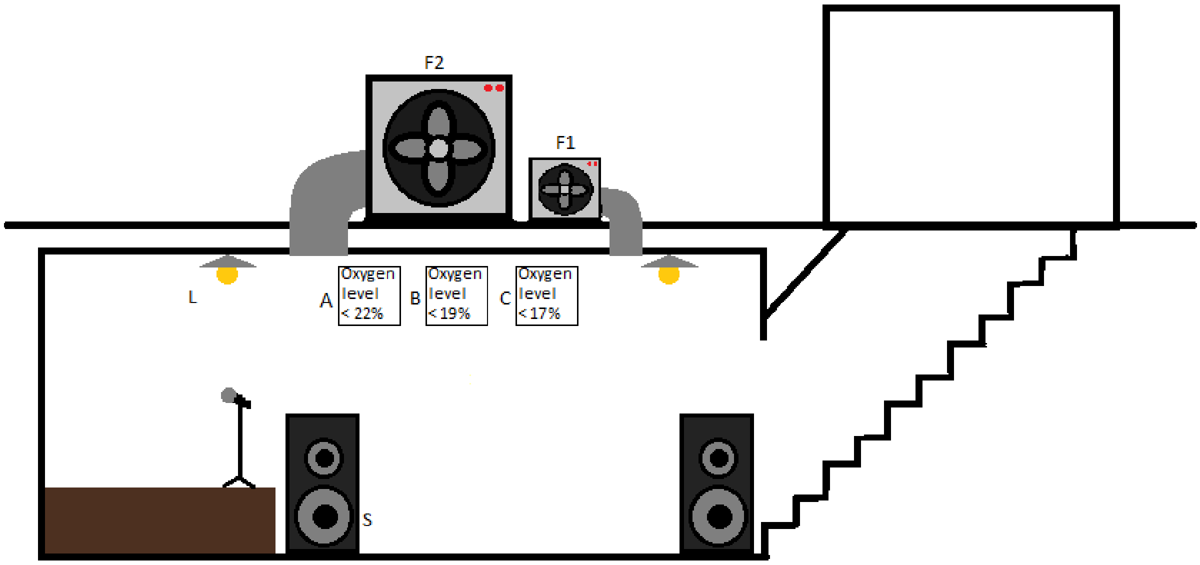
Experiment 2
Combinational Digital Circuit and Implementation of it on XILINX Vivado
Batu Arda Düzgün
03 / 10 / 2019

Abstract/ Objective:

In this experiment we need to come up with a real life problem then find a solution for it using logic and Boolean algebra. Then we will model our solutions as logic gate schemes. Finally we will implement our solutions on our BASYS 3 using XILINX Vivado. We are doing this how real file problems could be solved using logic and Boolean algebra then we are implementing our solutions to see if they hold up and get familiar with the tools (XILINX Vivado, BASYS 3) we will be using for the rest of the semester.

Design Specification Plan:

My real life problem is an underground bars ventilation based on the oxygen percentage and the music playing inside.



- A is a sensor which detects if the oxygen level in the room is below 22 percent.
- B is a sensor which detects if the oxygen level in the room is below 19 percent.
- C is a sensor which detects if the oxygen level in the room is below 17 percent.
- S is a sensor which detects if the speakers are being used.
- F1 is a fan.
- F2 is a bigger fan.
- L are lights.

F1 & F2 = 0 , L = 1 if A = 0 , S = 0
F1 = 1 F2 & L = 0 if A = 0 , S = 1
F1 & L = 1 , F2 = 0 if A = 1 , B = 0 , S = 0
F1 & F2 = 1 , L = 0 if A = 1 , B = 0 , S = 1
F1 & F2 & L = 1 if B = 1 , C = 0 , S = 0
F1 & F2 = 1 , L = 0 if B = 1 , C = 0 , S = 1
F1 & F2 & L = 1 if C = 1

As it can be seen it is a quite complex situation so to solve it I will be drawing a truth table of all the possible situations then from that truth table I will make K maps for each output. Then from those K maps I will make more logic gate scheme then implement it on my BASYS 3. I am using truth table and K map because they are quite robust tools which can be used to simplify even the most complex situations(which is only easy to show with 3, 4 variables).

Proposed Design Methodology:

First I draw the truth table based on the real life situation I took A, B, C, S censors as inputs and F1, F2 and L as outputs. A truth table is a mathematical table which sets out the functional values of logical expressions. While drawing it I put "d" for doesn't matter on situations which cannot be true such as censor B being active while censor A is in active

A	B	C	S	---	F1	F2	L
0	0	0	0	---	0	0	1
0	0	0	1	---	1	0	0
0	0	1	0	---	d	d	d
0	0	1	1	---	d	d	d
0	1	0	0	---	d	d	d
0	1	0	1	---	d	d	d
0	1	1	0	---	d	d	d
0	1	1	1	---	d	d	d
1	0	0	0	---	1	0	1
1	0	0	1	---	1	1	0
1	0	1	0	---	d	d	d
1	0	1	1	---	d	d	d
1	1	0	0	---	1	1	1
1	1	0	1	---	1	1	0
1	1	1	0	---	1	1	1
1	1	1	1	---	1	1	1

Then I draw 3 K maps from the truth table. K map is like a special truth table for only one output. On K maps the outputs are arranged in an easy to simplify way. I then using the K maps I found POS of each output. We can group neighboring cells of K maps because in logic we can say

$$\bar{A} * B * C * S + A * B * C * S = B * C * S$$

Then we can group these groups of two because

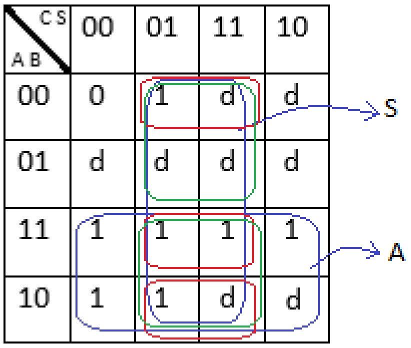
$$\bar{B} * C * S + B * C * S = C * S$$

Then we can group these groups of 4 because

$$\bar{C} * S + C * S = S$$

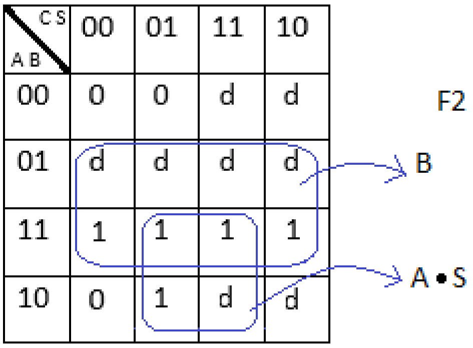
Then we can add these groups (+ is the "or" symbol)

(I showed the grouping up proses fully on the first one then just showed the last step to make it easier to see)



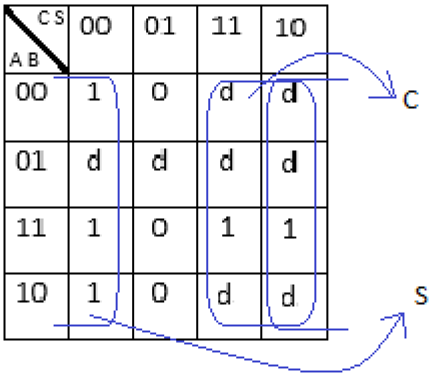
CS AB	00	01	11	10
00	0	1	d	d
01	d	d	d	d
11	1	1	1	1
10	1	1	d	d

F1 = S • A



CS AB	00	01	11	10
00	0	0	d	d
01	d	d	d	d
11	1	1	1	1
10	0	1	d	d

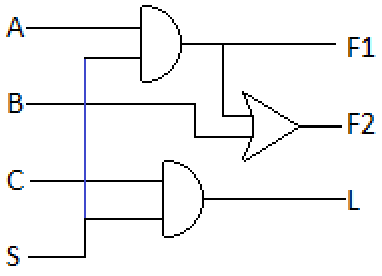
F2 = A • S + B



CS AB	00	01	11	10
00	1	0	d	d
01	d	d	d	d
11	1	0	1	1
10	1	0	d	d

L = C • A

Then using these POS's from the K maps I will construct my logic gate scheme.



The blue cable is not touching the B and C cables

Then I will implement my design on my BAYSY 3 with XILINX Vivado and test its functionality.