G.T.U

CSE232 – Logic Circuit and Design

Project 1

Student Name : Nevzat Seferoglu

Student Number : 171044024

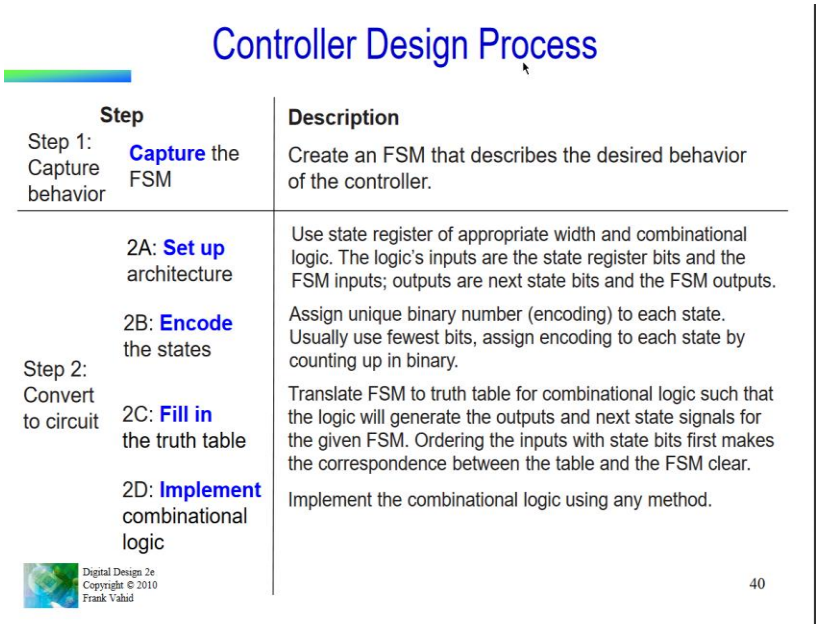
*Project Definition :

It will be a game implementation. In this game, you use 7 LEDs and two buttons for two players. The game starts with the led in the middle turned on. One user tries to shift the ON led to left while the other tries to push it to right. The one pushing the button faster wins the game. Winning occurs when the turned on led is at the rightmost or at the leftmost position. Do not forget to consider the cases when both players push the buttons at the same time, or they do not push at all.

Your FSM controller has two button inputs as B1 and B2. It has 7 outputs one for each led. Also, there will be a reset input to start a new game.

*Problem Approach :

Designing finite state machine was a first step to begin the project. After designing finite state machine boolean table can be constructing. For implementing this game , there were some requirement. These requirements are , controller , buttons , gates etc. Exploiting these components has actualized the project.

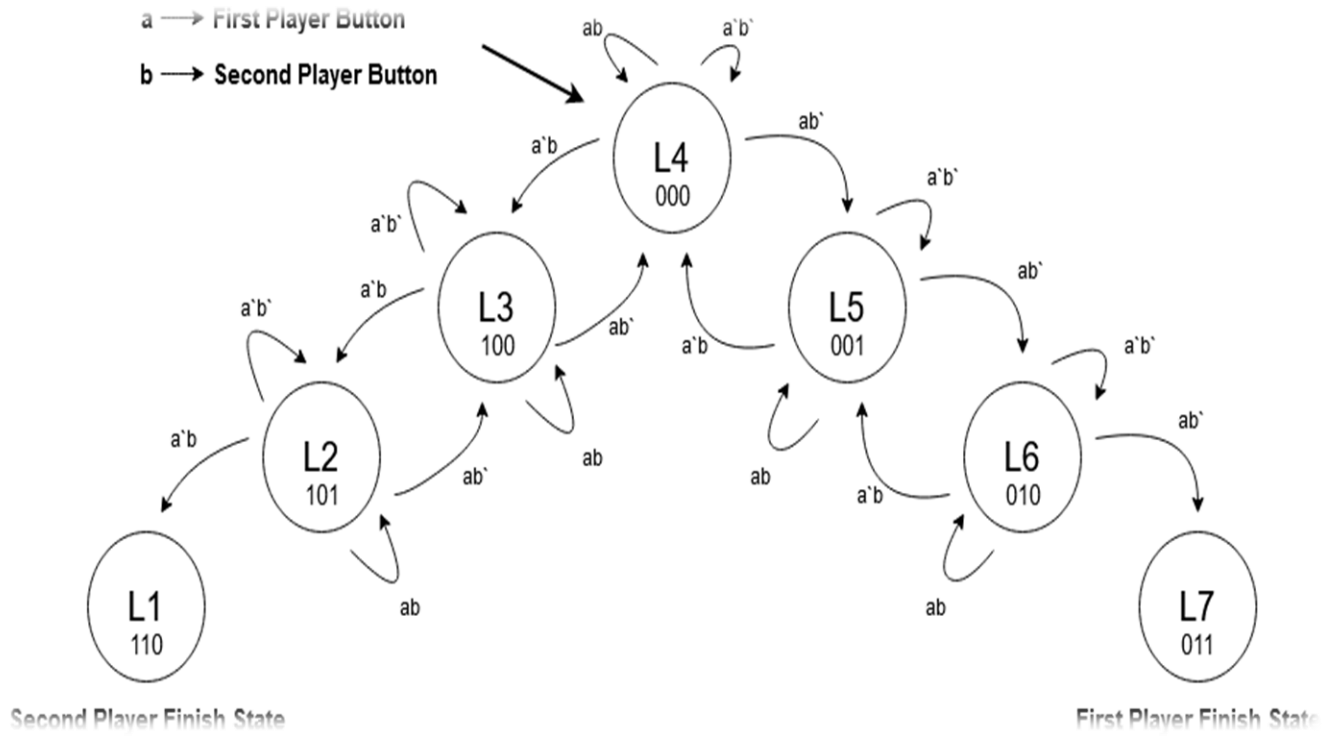


Using the technique of controller design process , there were two main stages of this project. The first one is combinational logic stage ; second one is sequential logic stage. They must exist at the same time ; because next stage calculated according to previous stage and given input. Conventional circuit is not enough to implement.

Project = Combinational circuit + Sequential circuit + Buttons

*Indication : Every single part must have been work properly.

Main Finite State Machine Design :



Boolean Table :

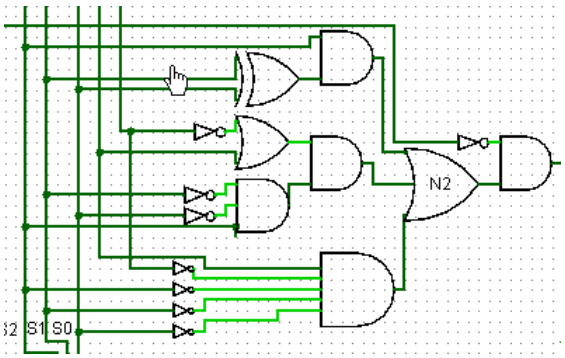
S2	S1	S0	Button1	Button2	N2	N1	N0	L1	L2	L3	L4	L5	L6	L7
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
0	0	1	1	0	0	1	0	0	0	0	0	1	0	0
0	0	1	0	1	0	0	0	0	0	0	0	1	0	0
0	0	1	1	1	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	1	1	0	1	0	0	1	0	0	0	0
1	0	0	1	1	1	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	1	1	0	0	0	0	0	1	0
0	1	0	0	1	0	0	1	0	0	0	0	0	1	0
0	1	0	1	1	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	1	0	1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	0	0	1	0	0	0	0	0

1	0	1	0	1	1	1	0	0	1	0	0	0	0	0
1	0	1	1	1	1	0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	1	1	0	0	0	0	0	0	1
0	1	1	1	0	0	1	1	0	0	0	0	0	0	1
0	1	1	0	1	0	1	1	0	0	0	0	0	0	1
0	1	1	1	1	0	1	1	0	0	0	0	0	0	1
1	1	0	0	0	1	1	0	1	0	0	0	0	0	0
1	1	0	1	0	1	1	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	1	0	0	0	0	0	0
1	1	0	1	1	1	1	0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	1	0	0	0
1	1	1	1	0	0	0	0	0	0	0	1	0	0	0
1	1	1	0	1	0	0	0	0	0	0	1	0	0	0
1	1	1	1	1	0	0	0	0	0	0	1	0	0	0

***Note :** Each “and” gate at the end of the expressions belongs to reset. So reset configuration is not shown in the expression. Reset button has been made at the end of the project and does not affect main purpose of each node majorly.

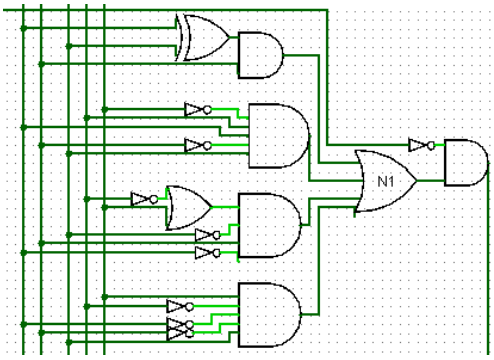
N2 = S2` S1` S0` B1` B2 + S2 S1` S0` B1` B2` + S2 S1` S0` B1` B2 + S2 S1` S0` B1 B2 + S2 S1` S0 + S2 S1 S0`

With some simplifying :



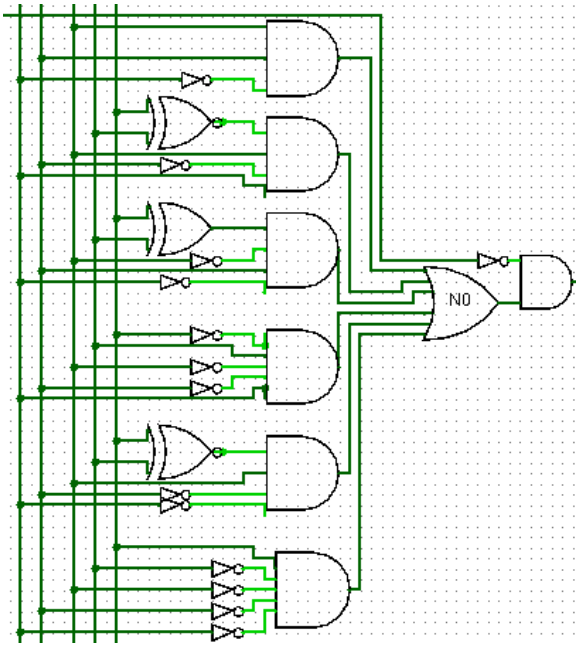
N1 = S2` S1` S0 B1 B2` + S2` S1 S0` B1` B2` + S2` S1 S0` B1 B2` + S2` S1 S0` B1 B2 + S2 S1` S0 B1` B2 + S2` S1 S0 + S2 S1 S0`

With some simplifying :



N0 = **S2` S1` S0` B1 B2` + S2` S1` S0 B1` B2` + S2` S1` S0 B1 B2 + S2 S1` S0` B1` B2 + S2` S1 S0` B1 B2` + S2` S1 S0` B1` B2 + S2 S1` S0 B1` B2` + S2 S1` S0 B1 B2 + S2` S1 S0**

With some simplifying :



***Note2 :** There are some XNOR and NOR as simplified gates. Each gate has been obtained by simplification ; I do not want to show complicated simplification process, but main expression can be taking out from the gates itself.

Led 1 = **S2 S1 S0`**

Led 2 = **S2 S1` S0**

Led 3 = **S2 S1` S0`**

Led 4 = **S2` S1` S0` + S2 S1 S0**

Led 5 = **S2` S1` S0**

Led 6 = **S2` S1 S0`**

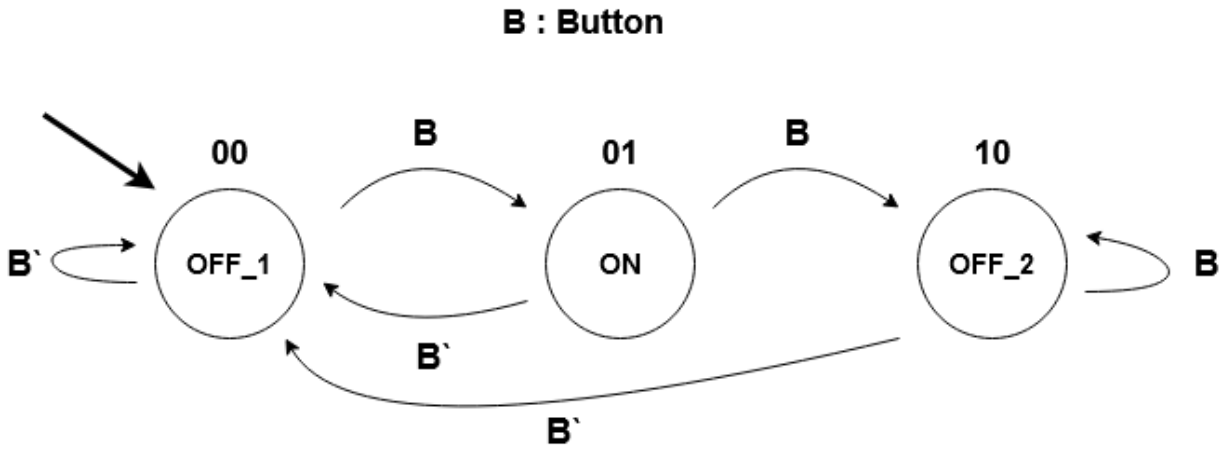
Led 7 = **S2` S1 S0**

***Note3 =** Led does not depend on button inputs. It depends only S2 , S1 , S0.

Result of main state :

As a result, N2 , N1 , N0 represent the next state of FSM , S2 , S1 , S0 represent current state of FSM. They can be converted to boolean table and this table can be converted into logic gates combination.

Buttons Finite State Machine Design :



Boolean Table :

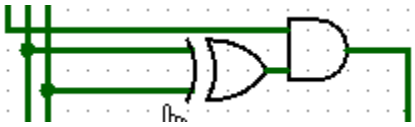
S1	S0	Button	N1	N0	Current State
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	0

N1 = **S1` S0 B + S1 S0` B**

With some simplifying :

N0 = **S1` S0` B**

With some simplifying :



Current State : **S1` S0**