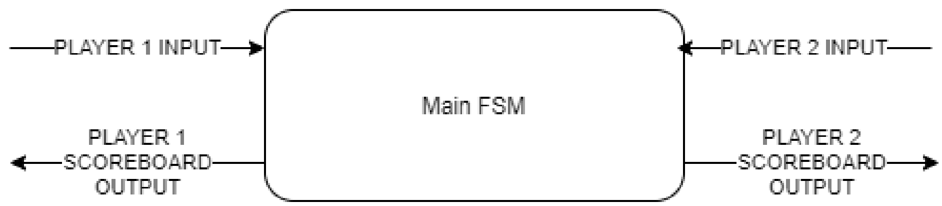
CSE 232 SPRING 2020

PROJECT 1

Berk PEKGÖZ

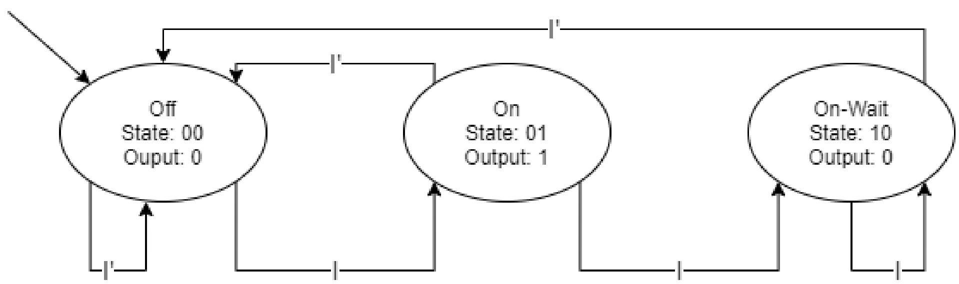
171044041

In this controller we have two inputs and seven outputs(leds) and two extra outputs for scoreboard.



In this game we have a exception. If one of the players keep pushing button. That push will be counted as one time push. It means Player 2 can win the game while Player 1 pushing. So we need "Input Synchronizer " for inputs. With this way we can provide this exception.

State diagram for Input Synchronizer:

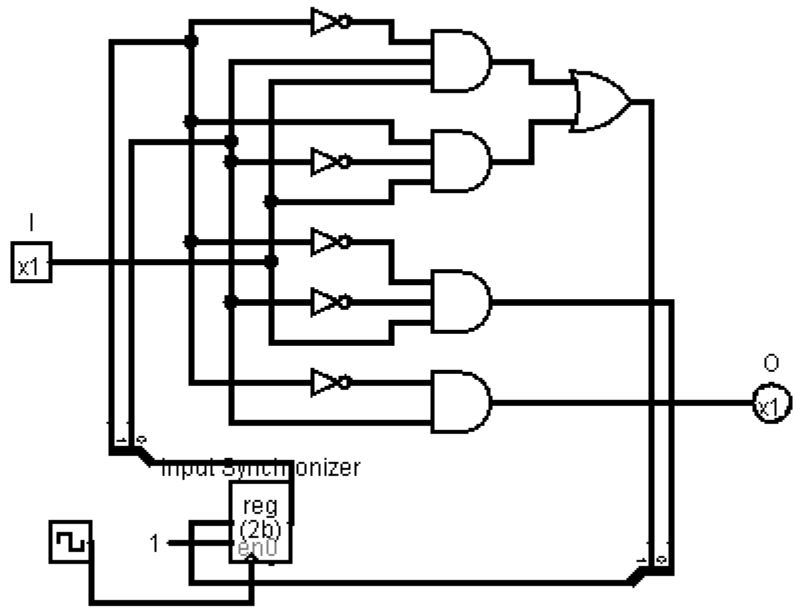


Truth table for Synchronizer:

Inputs			**Outputs**		
S1	**S0**	**I**	**N1**	**N0**	**O**
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

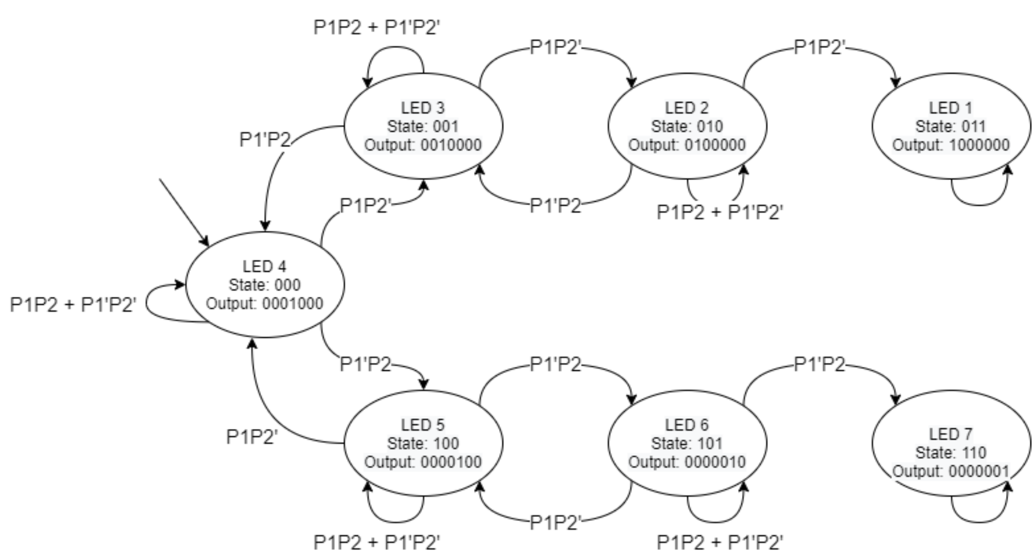
N1 = S1' S0 I + S1 S0' I
N0 = S1' S0' I
O = S1' S0 I' + S1' S0 I = S1' S0

Circuit for Synchronizer:



Now we have inputs synchronizers so we can design out Main FSM without thinking “infinite push” bug. If we would not have synchronizers we have the create 4 state for each LED output(a’b’ / a’b / ab’ / ab). In the end we would have 28 state. 28 state means 5 bit register.
>> 5 bit + 2 input bit = 7 bit truth table(128 row)
This design is very complex and error chance is very high. But we are using synchronizers so we don’t need 4 state for each LED output. So we can create 1 state for each LED output.

State Diagram for Main FSM:



Truth table for Main FSM:

P2 P1 >> Player 1-2
S2 S1 S0 >> Register outputs
L7 L6 L5 L4 L3 L2 L1 >> LEDS
N2 N1 N0 >> Register inputs
<u>Reset Input directly linked to the register’s reset bit, it is not on the truth table.</u>

P2	P1	S2	S1	S0	L7	L6	L5	L4	L3	L2	L1	N2	N1	N0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	1	0	0	1	0
0	0	0	1	1	0	0	0	0	0	0	1	0	1	1
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	1	0	0	0	0	0	1	0	1
0	0	1	1	0	1	0	0	0	0	0	0	1	1	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	1
0	1	0	0	1	0	0	0	0	1	0	0	0	1	0
0	1	0	1	0	0	0	0	0	0	1	0	0	1	1
0	1	0	1	1	0	0	0	0	0	0	1	0	1	1
0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
0	1	1	0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	1	0	1	0	0	0	0	0	0	1	1	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	1	0	1	1
1	0	1	0	0	0	0	1	0	0	0	0	1	0	1
1	0	1	0	1	0	1	0	0	0	0	0	1	1	0
1	0	1	1	0	1	0	0	0	0	0	0	1	1	0
1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	1	0	0	0	0	1
1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
1	1	0	1	1	0	0	0	0	0	0	1	0	1	1
1	1	1	0	0	0	0	1	0	0	0	0	1	0	0
1	1	1	0	1	0	1	0	0	0	0	0	1	0	1
1	1	1	1	0	1	0	0	0	0	0	0	1	1	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Boolean expression for Main FSM:

<u>Leds: (LED outpus only 1 when the S2S1S0 = Led’s state)</u>

L7 = S2 S1 S0’
L6 = S2 S1’ S0
L5 = S2 S1’ S0’
L4 = S2’ S1’ S0’
L3 = S2’ S1’ S0
L2 = S2’ S1 S0’
L1 = S2’ S1 S0

Register inputs:

N2 = P2'P1'S2S1'S0' + P2'P1'S2S1'S0 + P2'P1'S2S1S0' + P2'P1S2S1'S0 + P2'P1S2S1S0' + P2P1'S2'S1'S0' + P2P1'S2S1'S0' + P2P1'S2S1'S0 + P2P1'S2S1S0' + P2P1S2S1'S0' + P2P1S2S1'S0 + P2P1S2S1S0'

Simplifying >>

P2'P1'S2S1S0' + P2'P1S2S1S0' + P2P1'S2S1S0' + P2P1S2S1S0' = **S2S1S0'**
P2'P1'S2S1'S0 + P2'P1S2S1'S0 + P2P1'S2S1'S0 + P2P1S2S1'S0 = **S2S1'S0**
P2P1'S2'S1'S0' + P2P1'S2S1'S0' + = **P2P1'S1'S0'**
P2P1S2S1'S0 + P2P1'S2S1'S0' +P2P1'S2S1'S0 + P2P1S2S1'S0' = **P2S2S1'**
P2'P1'S2S1'S0' + P2'P1'S2S1'S0 + P2P1'S2S1'S0' + P2P1'S2S1'S0 = **P1'S2S1'**

N2 = P1'S2S1' + S2S1'S0 + S2S1S0' + P2P1'S1'S0' + P2S2S1'

N1 = P2'P1'S2'S1S0' + P2'P1'S2'S1S0 + P2'P1'S2S1S0' + P2'P1S2'S1'S0 + P2'P1S2'S1S0' + P2'P1S2'S1S0 + P2'P1S2S1S0' + P2P1'S2'S1S0 + P2P1'S2S1'S0 + P2P1'S2S1S0' + P2P1S2'S1S0' + P2P1S2'S1S0 + P2P1S2S1S0'

Simplifying >>

P2'P1'S2'S1S0' + P2'P1'S2'S1S0 +P2'P1S2'S1S0' + P2'P1S2'S1S0 = **P2'S2'S1**
P2'P1S2'S1S0' + P2'P1S2'S1S0 + P2P1S2'S1S0' + P2P1S2'S1S0 = + **P1S2'S1**
P2'P1'S2S1S0' + P2'P1S2S1S0' + P2P1'S2S1S0' + P2P1S2S1S0' = + **S2S1S0'**
P2'P1S2'S1'S0 + P2'P1S2'S1S0 = + **P2'P1S2'S0**
P2'P1'S2'S1S0 + P2'P1S2'S1S0 + P2P1'S2'S1S0 + P2P1S2'S1S0 = **S2'S1S0**
P2P1'S2S1'S0 = **P2P1'S2S1'S0**

N1= P2P1'S2S1'S0 + P2'P1S2'S0 + S2'S1S0 + S2S1S0' + P1S2'S1 + P2'S2'S1

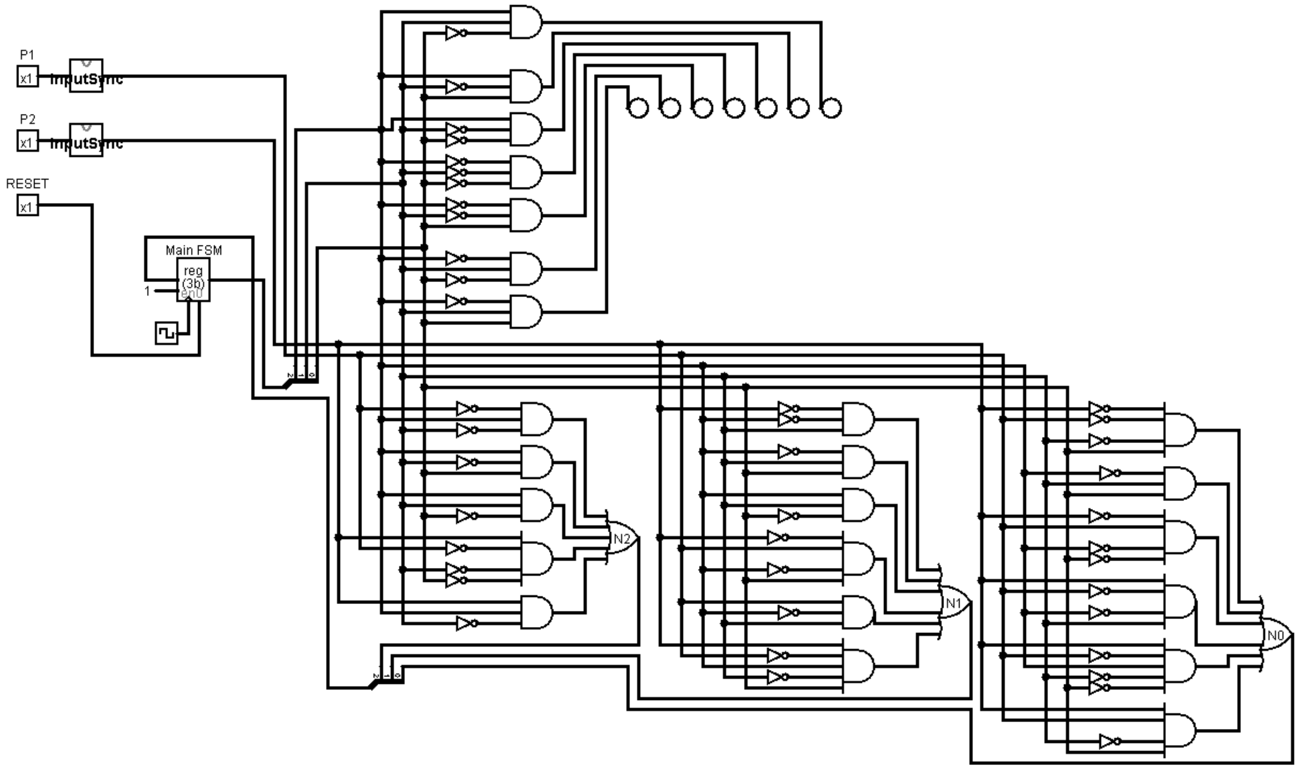
N0= P2'P1'S2'S1'S0 + P2'P1'S2'S1S0 + P2'P1'S2S1'S0 + P2'P1S2'S1'S0' + P2'P1S2'S1S0' + P2'P1S2'S1S0 + P2P1'S2'S1S0' + P2P1'S2'S1S0 + P2P1'S2S1'S0' + P2P1S2'S1'S0 + P2P1S2'S1S0 + P2P1S2S1'S0

Simplifying >>

P2'P1'S2'S1'S0 + P2'P1'S2S1'S0 = **P2'P1'S1'S0**
P2'P1S2'S1'S0' + P2'P1S2'S1S0' = **P2'P1S2'S0'**
P2P1'S2'S1S0' + P2P1'S2'S1S0 = **P2P1'S2'S1**
P2P1S2'S1'S0 + P2P1S2S1'S0 = **P2P1S1'S0**
P2'P1'S2'S1S0 + P2'P1S2'S1S0 + P2P1'S2'S1S0 + P2P1S2'S1S0 = **S2'S1S0**
P2P1'S2S1'S0' = **P2P1'S2S1'S0'**

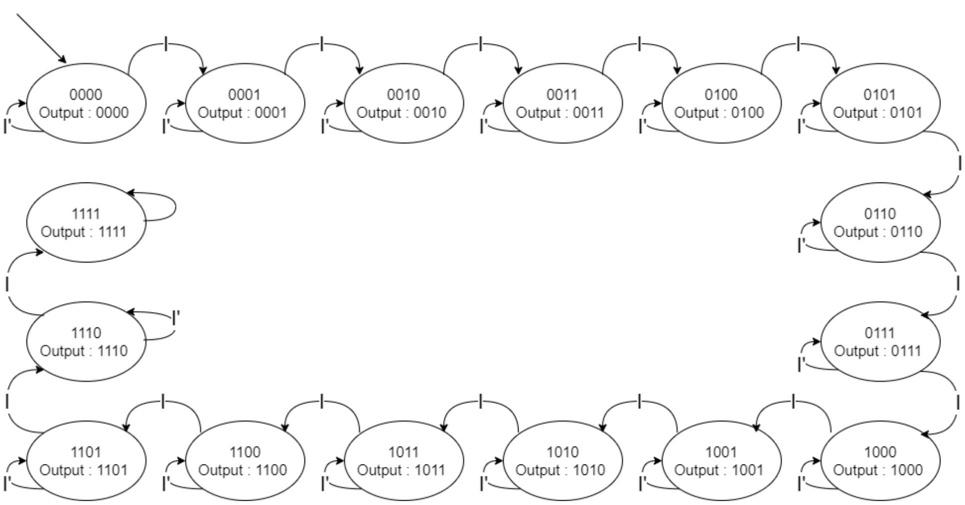
N0 = P2P1'S2S1'S0' + P2'P1'S1'S0 + P2'P1S2'S0' + P2P1'S2'S1 + P2P1S1'S0 + S2'S1S0

Circuit for Main FSM:



Score board counter FSM(EXTRA)

State diagram for score board:



Truth table for score board counter FSM:

I	S3	S2	S1	S0	N3	N2	N1	N0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	1	1
0	0	1	0	0	0	1	0	0
0	0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1	0
0	0	1	1	1	0	1	1	1
0	1	0	0	0	1	0	0	0
0	1	0	0	1	1	0	0	1
0	1	0	1	0	1	0	1	0
0	1	0	1	1	1	0	1	1
0	1	1	0	0	1	1	0	0
0	1	1	0	1	1	1	0	1
0	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	1	1
1	0	0	1	1	0	1	0	0
1	0	1	0	0	0	1	0	1
1	0	1	0	1	0	1	1	0
1	0	1	1	0	0	1	1	1
1	0	1	1	1	1	0	0	0
1	1	0	0	0	1	0	0	1
1	1	0	0	1	1	0	1	0
1	1	0	1	0	1	0	1	1
1	1	0	1	1	1	1	0	0
1	1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1	0
1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1

Boolean expression for Score Board counter FSM:

N3: S3 + IS2S1S0 (Can see in table obviously)

N2:

Can be see in first half of table obviously **= I’S2**

Second half = IS3'S2'S1S0 + S3'S2S1'S0' + S3'S2S1'S0 + S3'S2S1S0' + **IS3S2'S1S0** + S3S2S1'S0' + S3S2S1'S0 + S3S2S1S0' + S3S2S1S0 (We don’t need to consider ‘I’ this time because first half has them except I**S3'S2'S1S0 and IS3S2'S1S0**)

Simplifying >>

IS3'S2'S1S0 + IS3S2'S1S0 = I**S2'S1S0**
IS3'S2S1'S0' + IS3'S2S1'S0 + IS3S2S1'S0' + S3S2S1'S0 = **S2S1'**
S3'S2S1'S0' + S3'S2S1S0' + S3S2S1'S0' + S3S2S1S0' = **S2S0'**
S3S2S1S0 + S3S2S1'S0' + S3S2S1'S0 + S3S2S1S0' = **S3S2**

N2: I’S2 +IS2'S1S0 + S2S1' + S2S0' + S3S2

N1: I'S3'S2'S1S0' + I'S3'S2'S1S0 + I'S3'S2S1S0' + I'S3'S2S1S0 + I'S3S2'S1S0' + I'S3S2'S1S0 + I'S3S2S1S0' + I'S3S2S1S0 + IS3'S2'S1'S0 + IS3'S2'S1S0' + IS3'S2S1'S0 + IS3'S2S1S0' + IS3S2'S1'S0 + IS3S2'S1S0' + IS3S2S1'S0 + IS3S2S1S0' + IS3S2S1S0

Simplifying >>

I'S3'S2'S1S0' + I'S3'S2'S1S0 + I'S3'S2S1S0' + I'S3'S2S1S0 + I'S3S2'S1S0' + I'S3S2'S1S0 + I'S3S2S1S0' + I'S3S2S1S0 = + **I'S1**
I'S3'S2'S1S0' + I'S3'S2S1S0' + I'S3S2'S1S0' + I'S3S2S1S0' + IS3'S2'S1S0' + IS3'S2S1S0' + IS3S2'S1S0' + IS3S2S1S0' = **S1S0'**
IS3'S2'S1'S0 + IS3'S2S1'S0 + IS3S2'S1'S0 ' + IS3S2S1'S0 = **IS1'S0**
I'S3S2S1S0' + I'S3S2S1S0 + IS3S2S1S0' + IS3S2S1S0 = **S3S2S1**

N1: IS1'S0 + I'S1 + S1S0' + S3S2S1

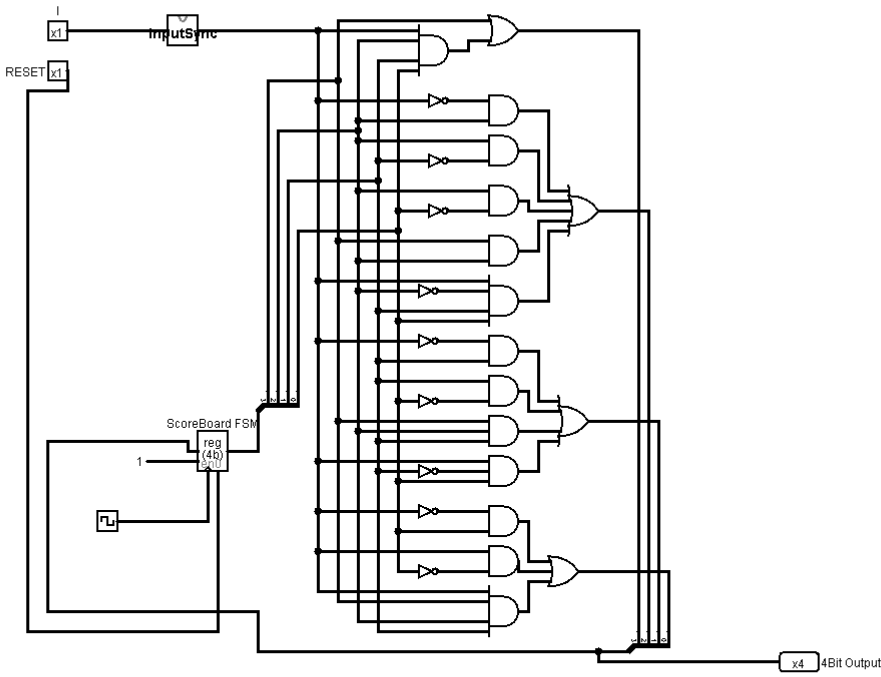
N0: I'S3'S2'S1'S0 + I'S3'S2'S1S0 + I'S3'S2S1'S0 + I'S3'S2S1S0 + I'S3S2'S1'S0 + I'S3S2'S1S0 + I'S3S2S1'S0 + I'S3S2S1S0 + IS3'S2'S1'S0' + IS3'S2'S1S0' + IS3'S2S1'S0' + IS3'S2S1S0' + IS3S2'S1'S0' + IS3S2'S1S0' + IS3S2S1'S0' + IS3S2S1S0' + IS3S2S1S0

Simplifying >>

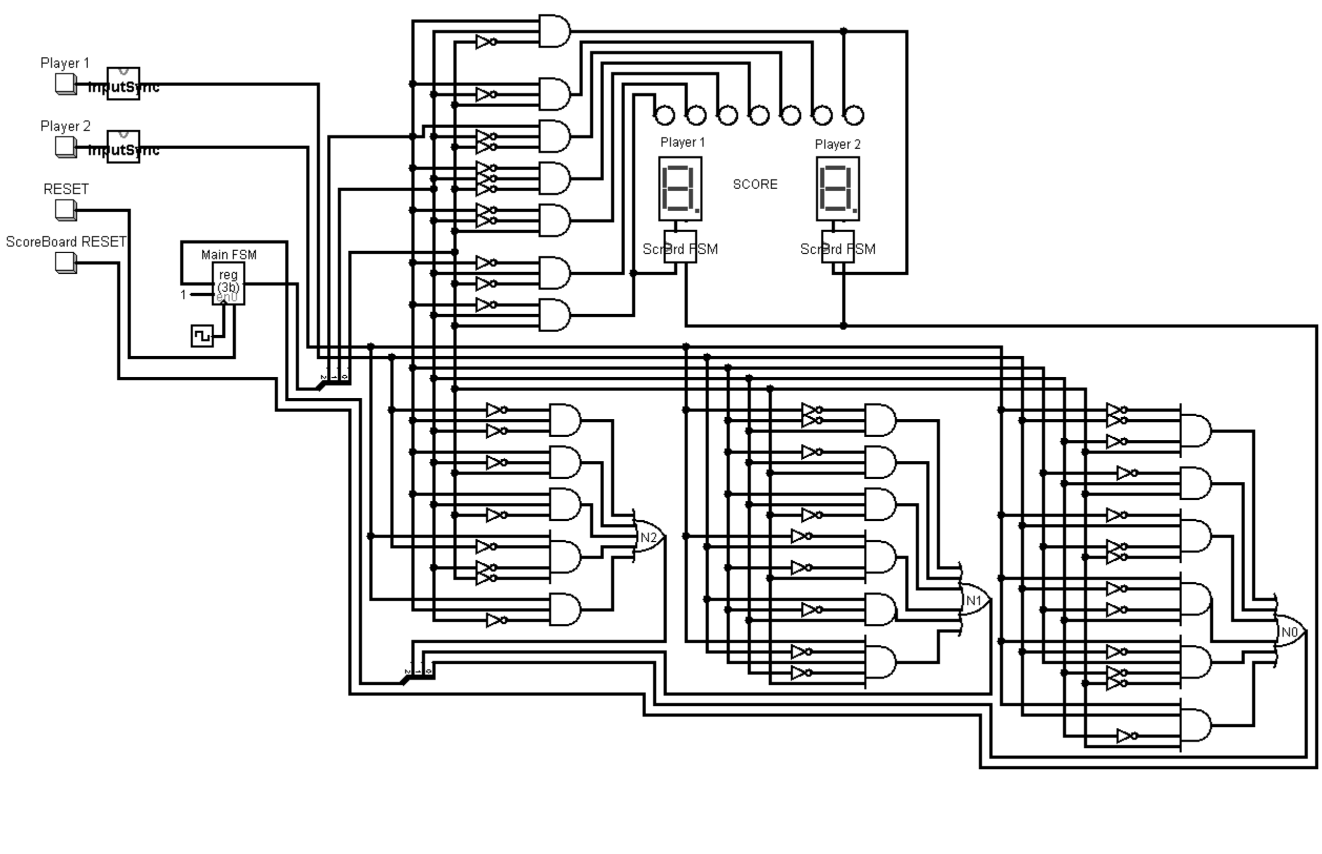
IS3S2S1S0' + IS3S2S1S0 = **IS3S2S1**
I'S3'S2'S1'S0 + I'S3'S2'S1S0 + I'S3'S2S1'S0 + I'S3'S2S1S0 + I'S3S2'S1'S0 + I'S3S2'S1S0 + I'S3S2S1'S0 + I'S3S2S1S0 = **I'S0**
IS3'S2'S1'S0' + IS3'S2'S1S0' + IS3'S2S1'S0' + IS3'S2S1S0' + IS3S2'S1'S0' + IS3S2'S1S0' + IS3S2S1'S0' + IS3S2S1S0' = **IS0'**

N0: I'S0 + IS0' + IS3S2S1

Circuit for Score board counter FSM:

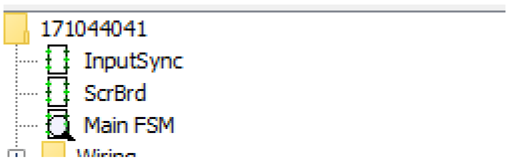


FINAL VERSION OF CIRCUIT:



.circc file of Project:

Scoreboard counter FSM and Button Synchronizer implemented as circuit.



Details of game:

- 7 LEDs, two buttons for two players, two buttons for Reset and Score Board reset.
- 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 and Score of that player increase on scoreboard.
- If both players push the buttons at the same time. LED does not change.
- If one the players keep pushing the button without backing. LED moves only one time.
- If one of the players play correctly(push and back) while other player keep pushing without backing. That player wins the game.
- If one of the players keeps pushing the button. Other player can still play the game(Other player's input is still valid). Because that player's push counted only one time.
- Reset button can be used anytime. And resets the game. Scoreboard will not be reset.
- Score Board reset button can be used any time and Score's of both players will be 0.