LOGIC CIRCUITS

AND

DIGITAL DESING

SPRING 2020

PROJECT #1

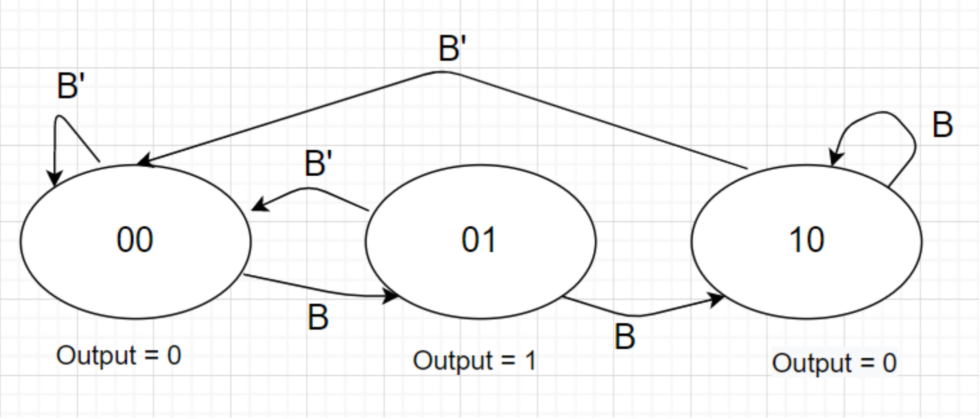
Yusuf Akgül

171044007

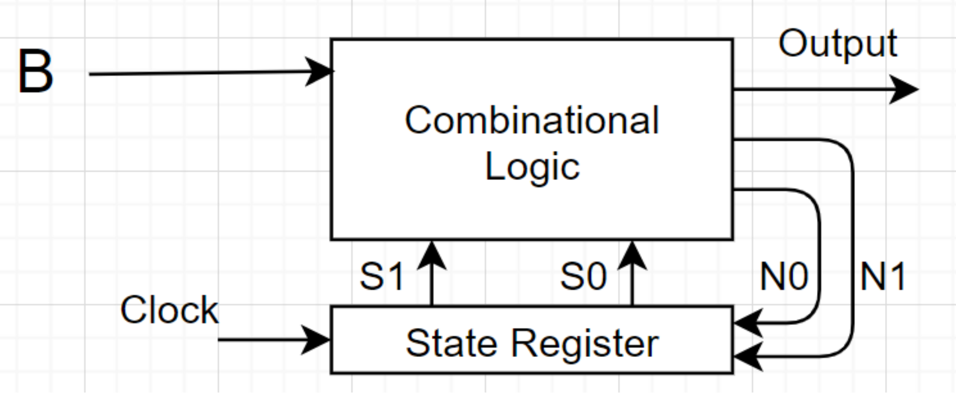
While starting the project, I wrote an FSM for the button to solve the pressed state of the buttons.

If the player remains pressed for a long time, the system will only send 1 pulse, even though it still presses the button.

FSM for the button that works the way we want



Controller for Button FSM



Then truth table for Button FSM

INPUTS **OUTPUTS**

S1	S0	B	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

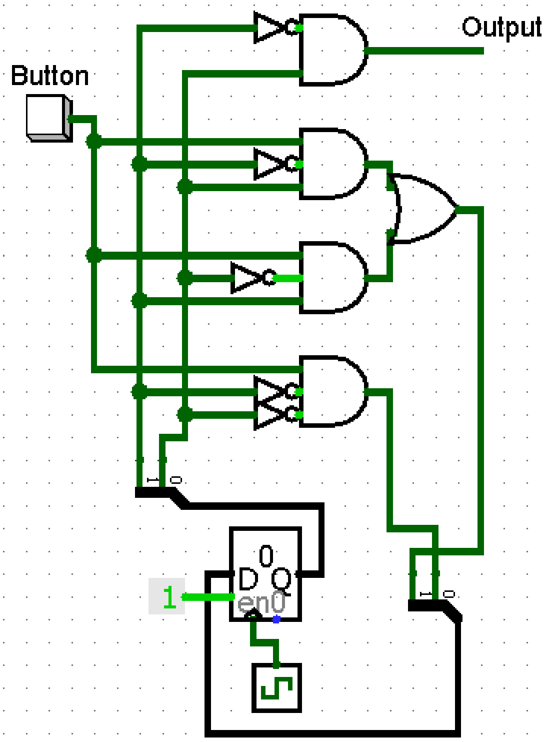
Boolean Equations of truth table

n1 = S1’S0B+S1S0’B

n0 = S1’S0’B

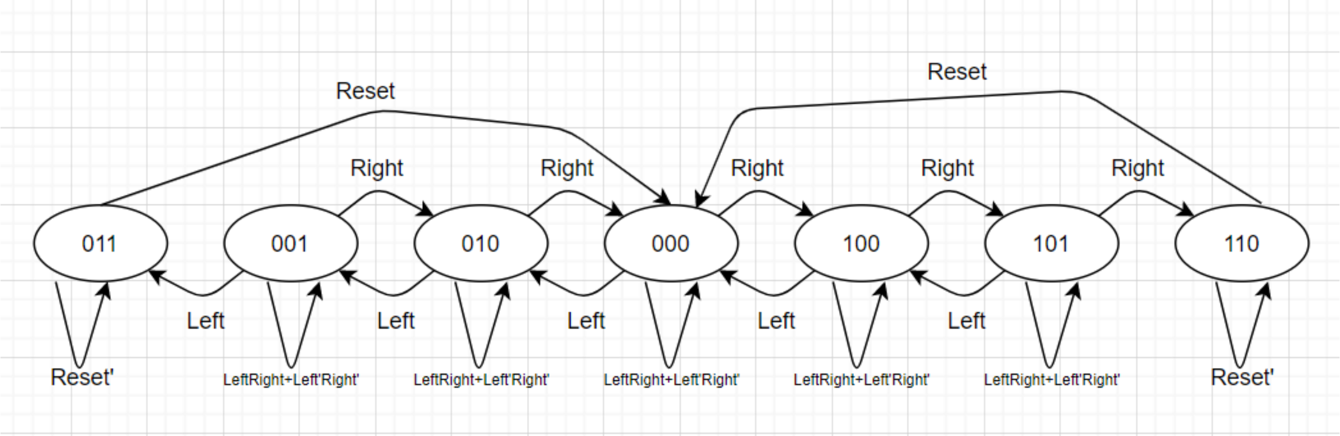
O = S1’S0

Then Design On Logisim

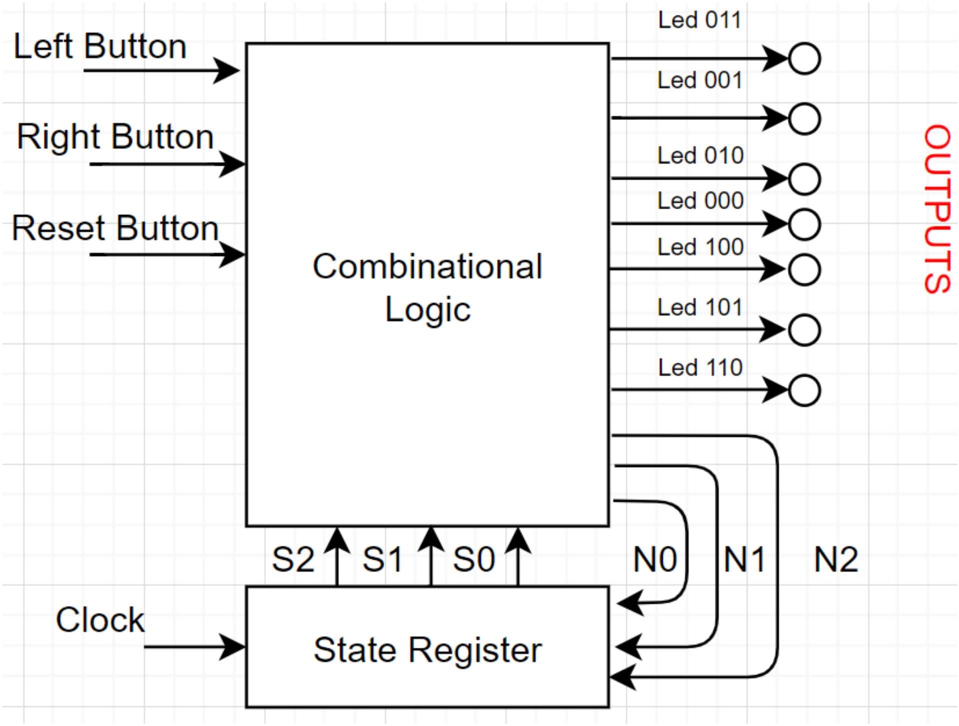


After I made the button in a way that would not affect the game if it was pressed for a long time, I went to design the main LED game.

FSM for LED GAME



Controller for LED Game FSM



Then truth table for LED Game FSM

Note : **L = left input , R = right input, X = reset input.**

S2,S1,S0 current state and n2,n1,n0 next state.

L2 = Led left 2, R2 = Led Right 2, M = Led middle

Respectively L2, L1, L0, M, R0, R1, R2

N= Value that does not affect the result in input. 1 or 0, does not matter.

INPUTS **NEXT STATE** **LEDS**

	S2	S1	S0	L	R	X	n2	n1	n0	L2	L1	L0	M	R0	R1	R2
LED L2	0	1	1	N	N	0	0	1	1	1	0	0	0	0	0	0
	0	1	1	N	N	1	0	0	0	1	0	0	0	0	0	0
LED L1	0	0	1	0	0	N	0	0	1	0	1	0	0	0	0	0
	0	0	1	0	1	N	0	1	0	0	1	0	0	0	0	0
	0	0	1	1	0	N	0	1	1	0	1	0	0	0	0	0
	0	0	1	1	1	N	0	0	1	0	1	0	0	0	0	0
LED L0	0	1	0	0	0	N	0	1	0	0	0	1	0	0	0	0
	0	1	0	0	1	N	0	0	0	0	0	1	0	0	0	0
	0	1	0	1	0	N	0	0	1	0	0	1	0	0	0	0
	0	1	0	1	1	N	0	1	0	0	0	1	0	0	0	0
LED M	0	0	0	0	0	N	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	1	N	1	0	0	0	0	0	1	0	0	0
	0	0	0	1	0	N	0	1	0	0	0	0	1	0	0	0
	0	0	0	1	1	N	0	0	0	0	0	0	1	0	0	0

LED **R0**	1	0	0	0	0	N	1	0	0	0	0	0	0	1	0	0
	1	0	0	0	1	N	1	0	1	0	0	0	0	1	0	0
	1	0	0	1	0	N	0	0	0	0	0	0	0	1	0	0
	1	0	0	1	1	N	1	0	0	0	0	0	0	1	0	0
LED **R1**	1	0	1	0	0	N	1	0	1	0	0	0	0	0	1	0
	1	0	1	0	1	N	1	1	0	0	0	0	0	0	1	0
	1	0	1	1	0	N	1	0	0	0	0	0	0	0	1	0
	1	0	1	1	1	N	1	0	1	0	0	0	0	0	1	0
LED **R2**	1	1	0	N	N	0	1	1	0	0	0	0	0	0	0	1
	1	1	0	N	N	1	0	0	0	0	0	0	0	0	0	1

Boolean Equations of truth table

n2 = s2’s1’s0’L’R + s2s1’s0’L’R’ + s2s1’s0’L’R + s2s1’s0’LR + s2s1’s0 + s21s0’X

n1 = s2’s1s0X’ + s2’s1’s0L’R + s2’s1’s0LR’+ s2’s1s0’L’R’ + s2’s1s0’LR + s2’s1’s0’LR’ + s2s1’s0L’R + s2s1s0’X’

n0 = s2’s1s0r’ + s2’s1’s0L’R’ + s2’s1’s0LR’ + s2’s1’s0LR + s2’s1s0’LR’ + s2s1’s0’L’R + s2s1’s0L’R’ + s2s1’s0LR

Then Simplified version of n2, n1, n0

n2 = s1’s0’L’R + s2s1s0’X’ + s2s1’L’ + s2s1’X + s2s1’s0

n1 = s2’s1s0’L’R’ + s2’s1s0’LR + s2’s1’LR’ + s1’s0L’R + s2s1s0’X’ + s2’s1s0X’

n0 = s2’s1s0’LR’ + s2s1’s0’L’R + s1’s0L’R’ + s2’s1s0X’ +s1’s0LR + s2’s1’s0L

Boolean Equations for Leds

L2(Led Left2) = s2’s1s0

L1(Led Left1) = s2’s1’s0

L0(Led Left0) = s2’s1s0’

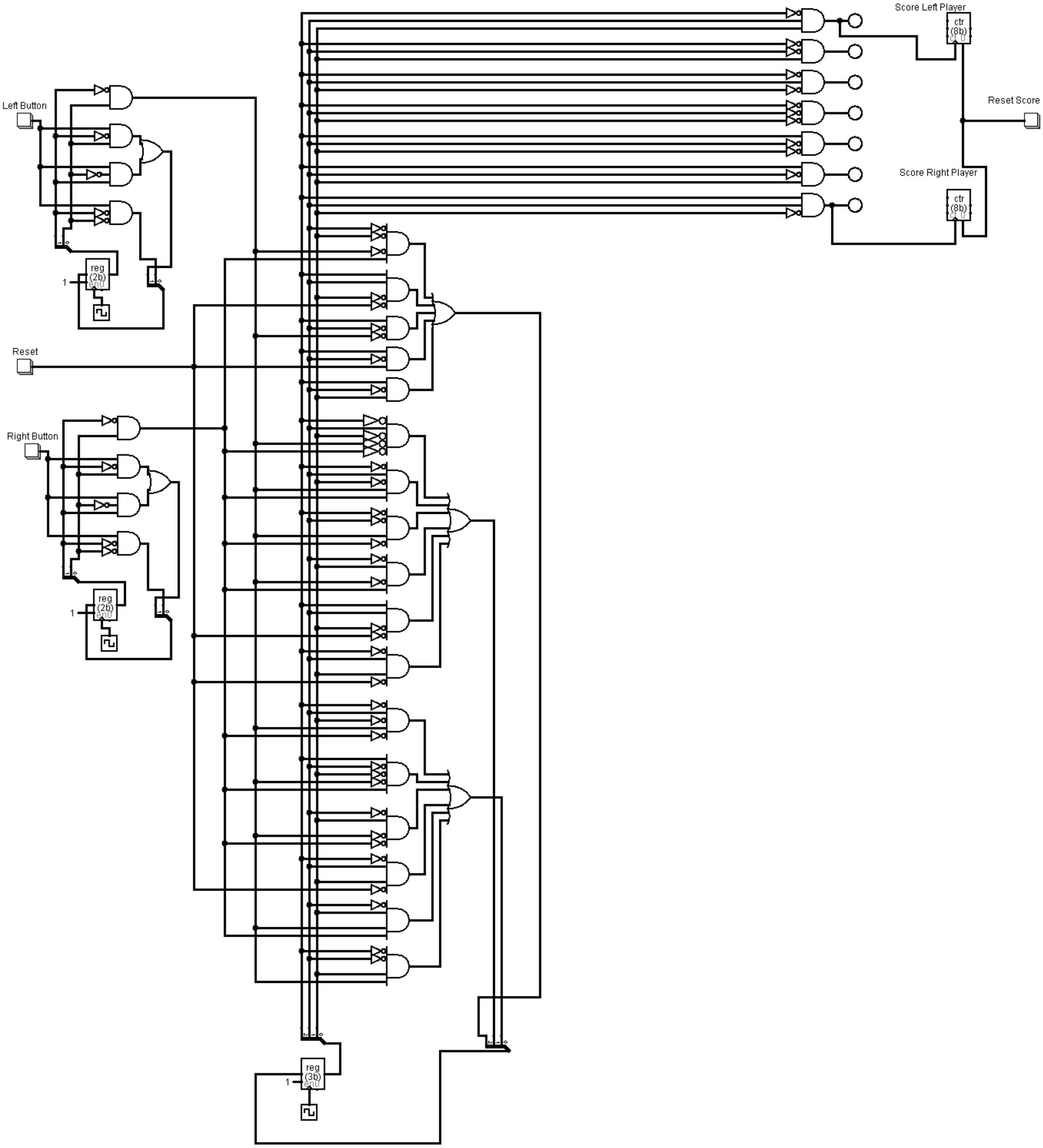
M(Led Middle) = s2’s1’s0’

R0(Led Right0) = s2s1’s0’

R1(Led Right1) = s2s1’s0

R2(Led Right2) = s2s1s0’

Then Design On Logisim


Score Left Player
ctr
(8b)
Left Button
reg
(2b)
1
Reset Score
Score Right Player
ctr
(8b)
Reset
Right Button
reg
(2b)
1
reg
(3b)
1