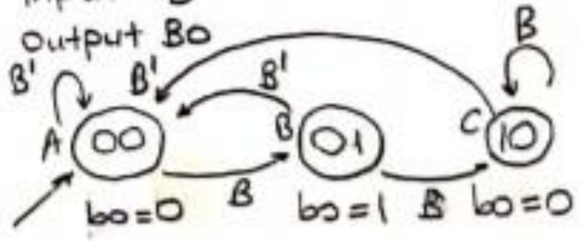
State Diagrams :

input B

output Bo



Truth Table

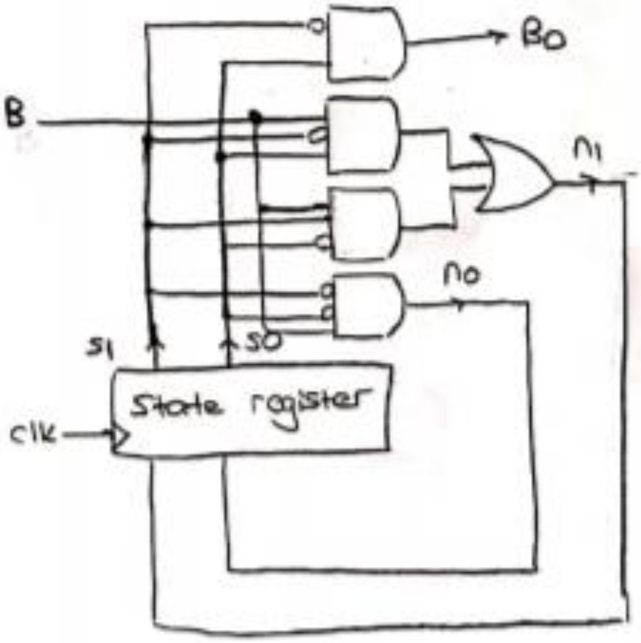
	s1	s0	B	n1	n0	Bo
A	0	0	0	0	0	0
	0	0	1	0	1	0
B	0	1	0	0	0	1
	0	1	1	1	0	1
C	1	0	0	0	0	0
	1	0	1	1	0	0
Unused	1	1	0	0	0	0
	1	1	1	0	0	0

$n1 = s1's0B + s1s0'B$

$n0 = s1's0'B$

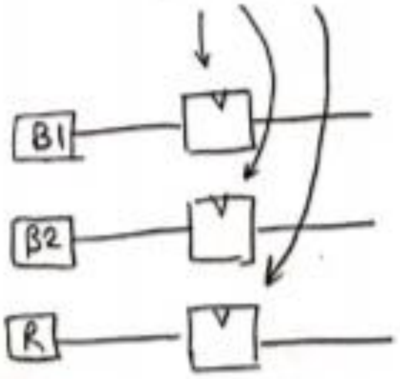
$Bo = s1's0B + s1's0B'$

$= s1's0$



This diagram helps us to convert button press to single cycle duration, regardless of length of time that button was actually pressed

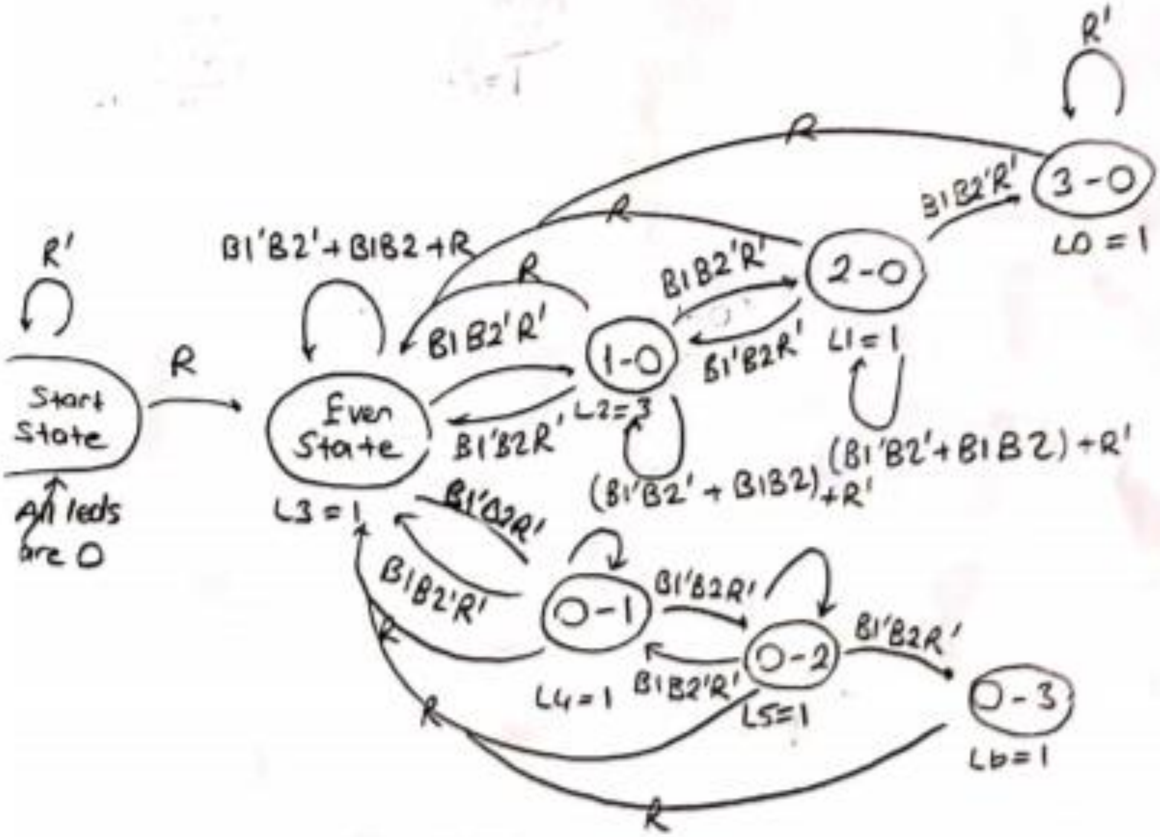
with help of this, in our game, when a player holds the button, cycle just one time will be 1

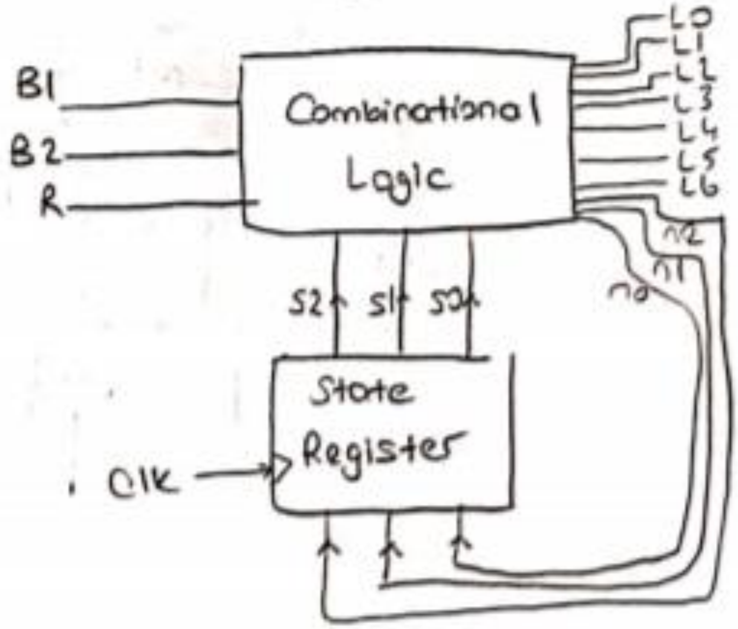


Main Game

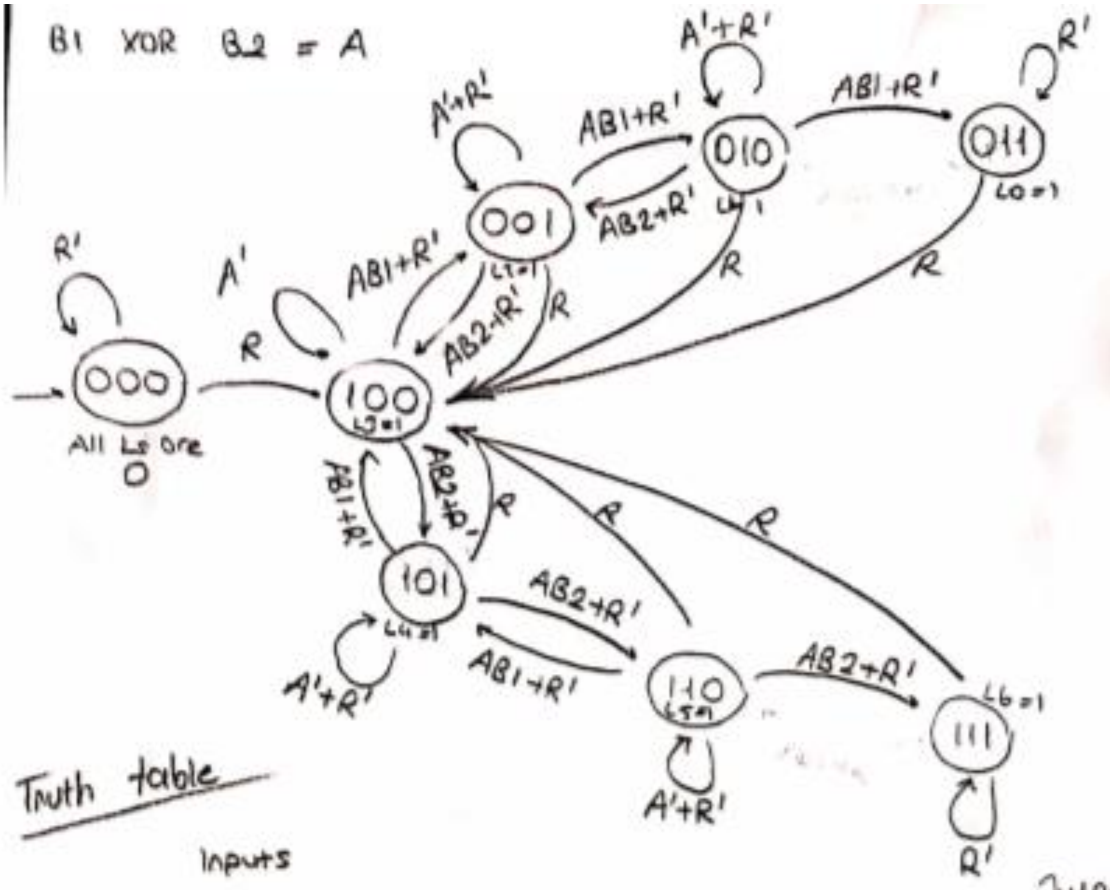
Inputs : B1, B2, R

Outputs : L0, L1, L2 L3, L4, L5, L6





B1 XOR B2 = A



Truth table

Inputs						Outputs									
S2	S1	S0	B1	B2	R	n2	n1	n0	L0	L1	L2	L3	L4	L5	L6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0
0	0	1	1	1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0

Start State

1-0 State

s2	s1	s0	B1	B2	R	n2	n1	n0	L0	L1	L2	L3	L4	L5	L6	
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	0	1	0	0	0	0	0	2-0 State
0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	
0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	
0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	
0	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	
0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	
0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	
0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	3-0 State
0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	
0	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	
0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	1	0	0	0	0	0	0	
0	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	
1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	
1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	Even state
1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	
1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	
1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	
1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	
1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	
1	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	

s2	s1	s0	B1	B2	R	n0	n1	n2	L0	L1	L2	L3	L4	L5	L6	
1	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	
1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	
1	0	1	0	1	0	1	1	0	0	0	0	0	1	0	0	
1	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0-1 State
1	0	1	1	0	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	0	
1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	
1	0	1	1	1	1	1	0	0	0	0	0	0	1	0	0	
1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	
1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	
1	1	0	0	1	0	1	1	1	0	0	0	0	0	1	0	
1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0-2 State
1	1	0	1	0	0	1	0	1	0	0	0	0	0	1	0	
1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	
1	1	0	1	1	0	1	1	0	0	0	0	0	0	1	0	
1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	
1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	
1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	
1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	1	0-3 State
1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	
1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	1	
1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	
1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	1	
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	

Boolean Expressions:

$L0 = S2'S1S0$ $\quad L4 = S2S1'S0$

$L1 = S2'S1S0'$ $\quad L5 = S2S1S0'$

$L2 = S2'S1'S0$ $\quad L6 = S2S1S0$

$L3 = S2S1'S0'$

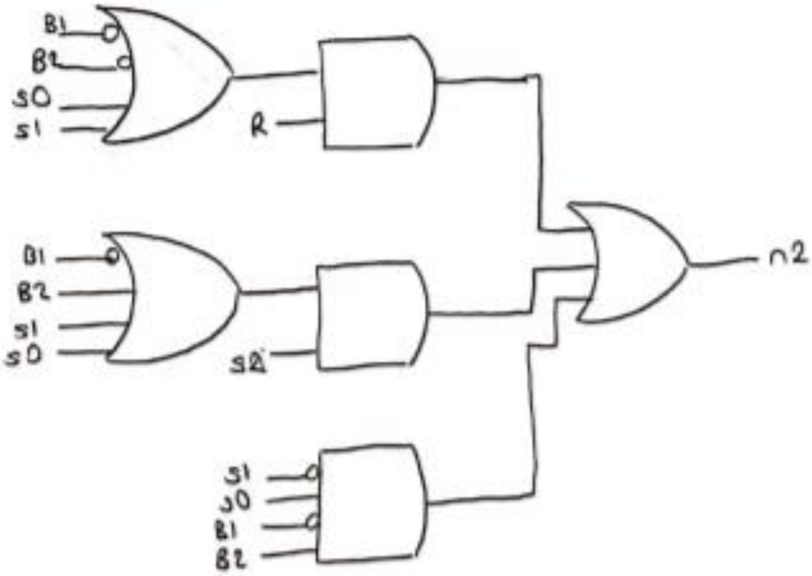
$n2 = B1'R + B2'R + S0R + S1R + S2B1' + S2B2 + S2S1 + S1'S0B1'B2 + S2S0$

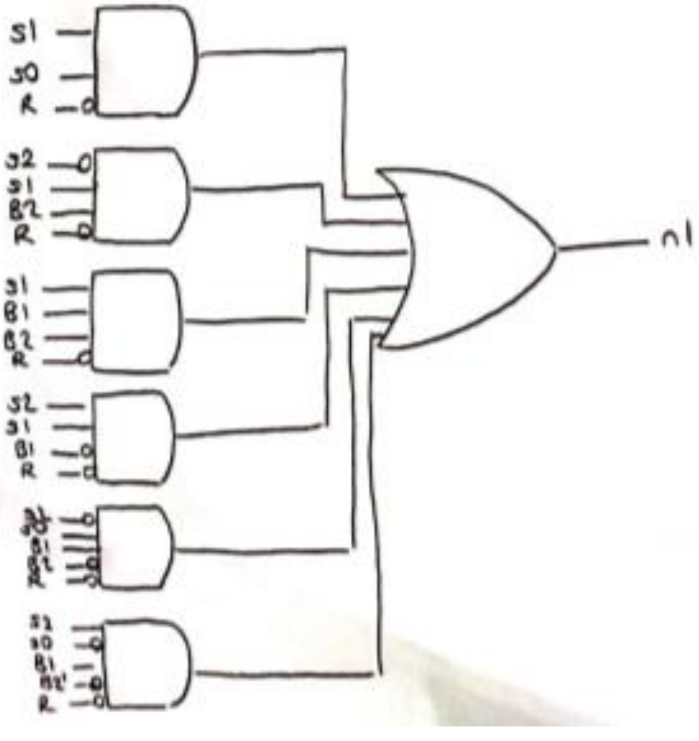
$R(B1' + B2' + S0 + S1) + S2(B1' + B2 + S1 + S0) + S1'S0B1'B2$

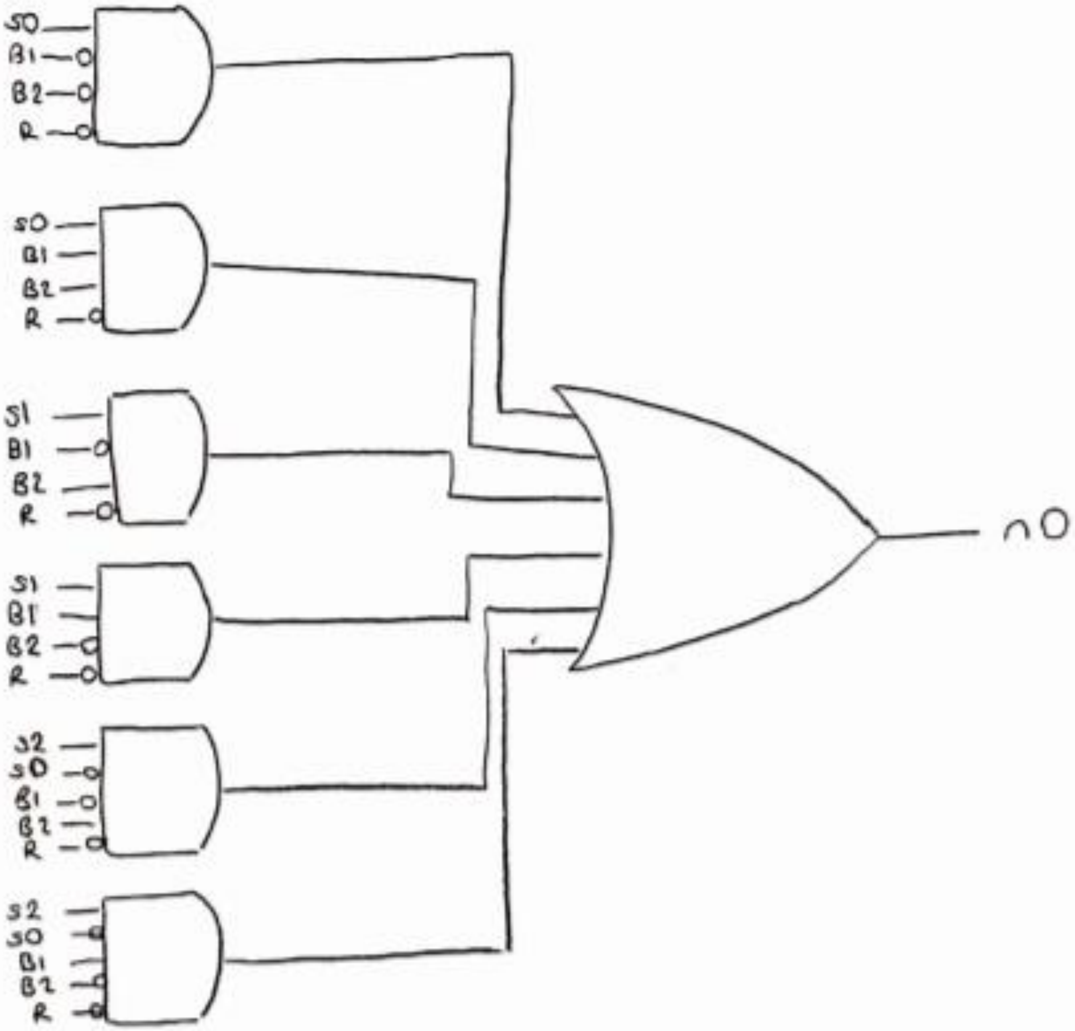
$n1 = S1S0R' + S2'S1B2R' + S1B1B2R' + S2S1B1'R' + S2'S0B1B2'R' + S2S0B1'B2R'$

$n0 = S0B1'B2'R' + S0B1B2R' + S1B1'B2R' + S1B1B2'R' + S2S0'B1'B2R' + S2S0'B1B2'R'$

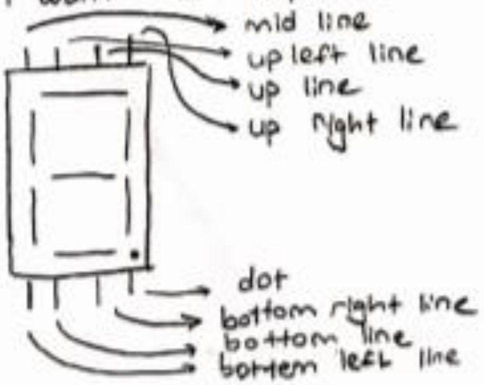
Combinational Logic





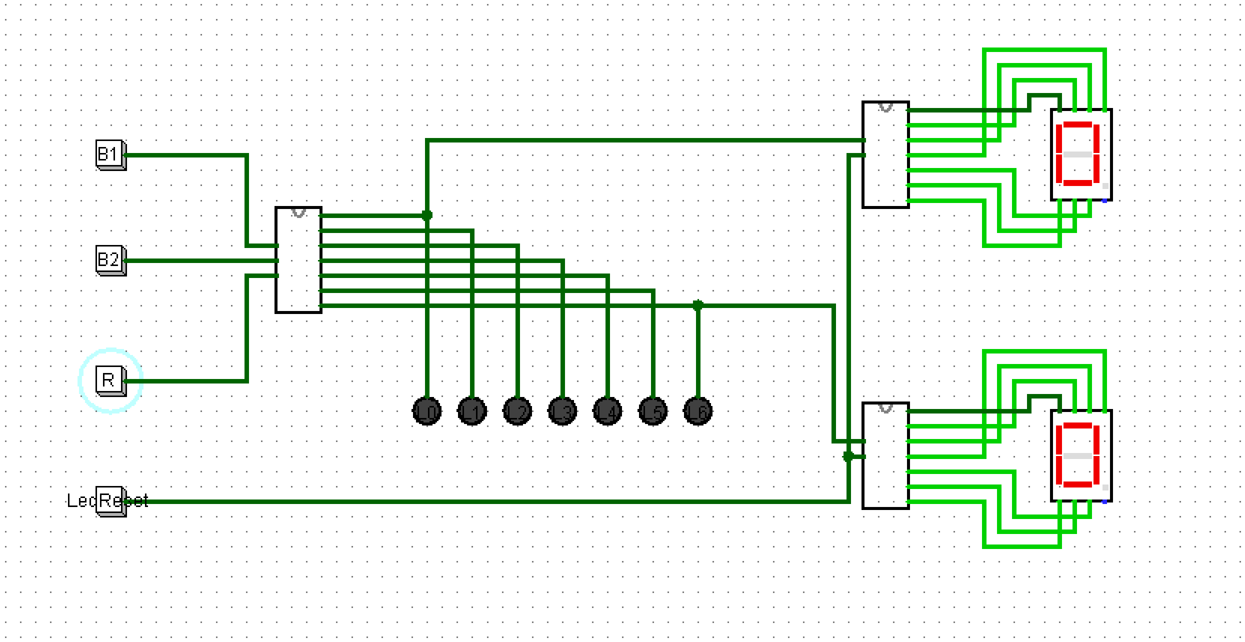


I want to implement a FSM that uses 7-segment display

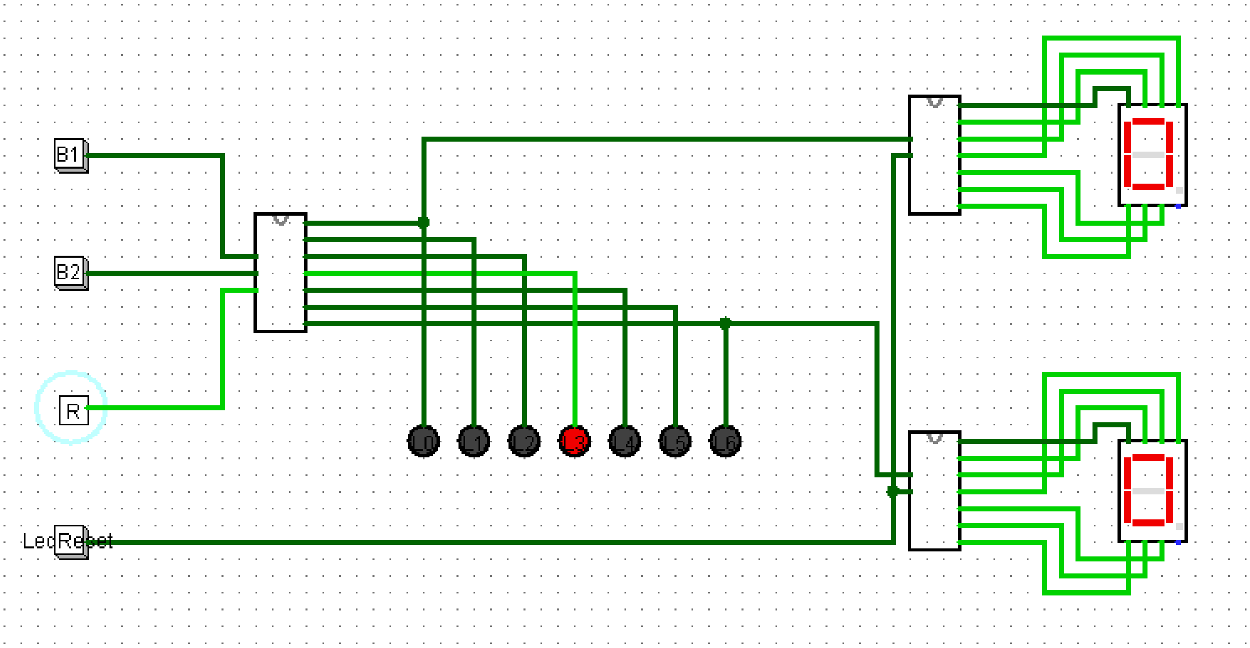


There will be two inputs and 7 outputs and state register will be 3-bit

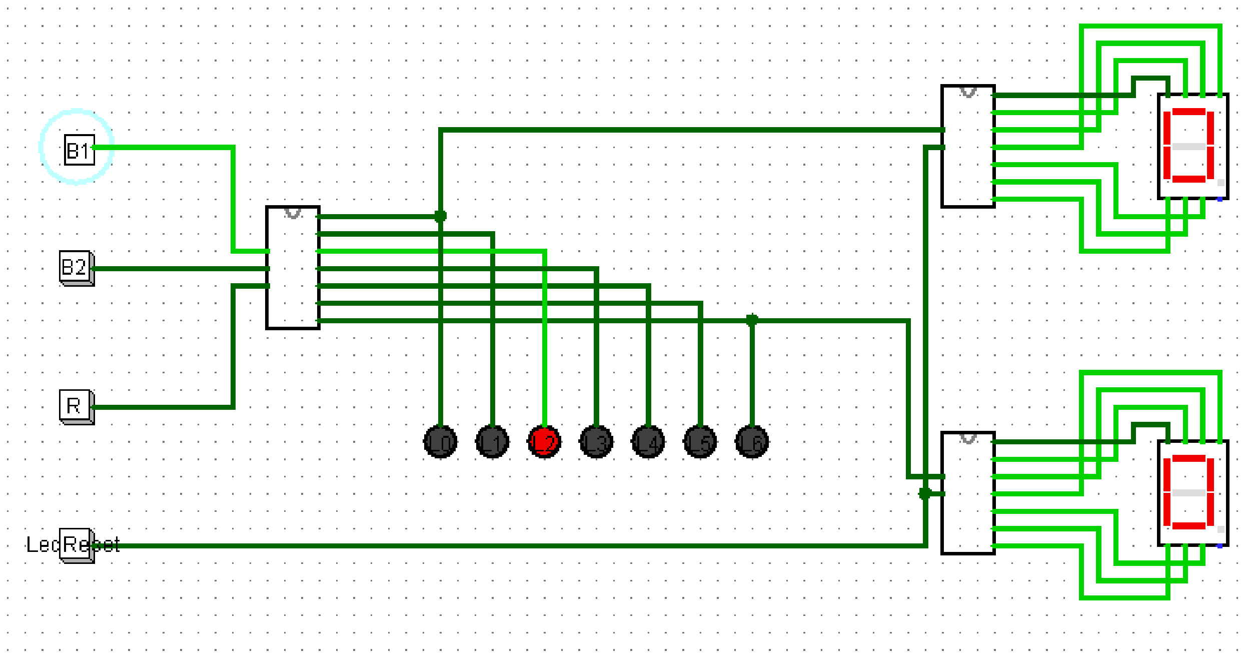
Start State: No leds are not open



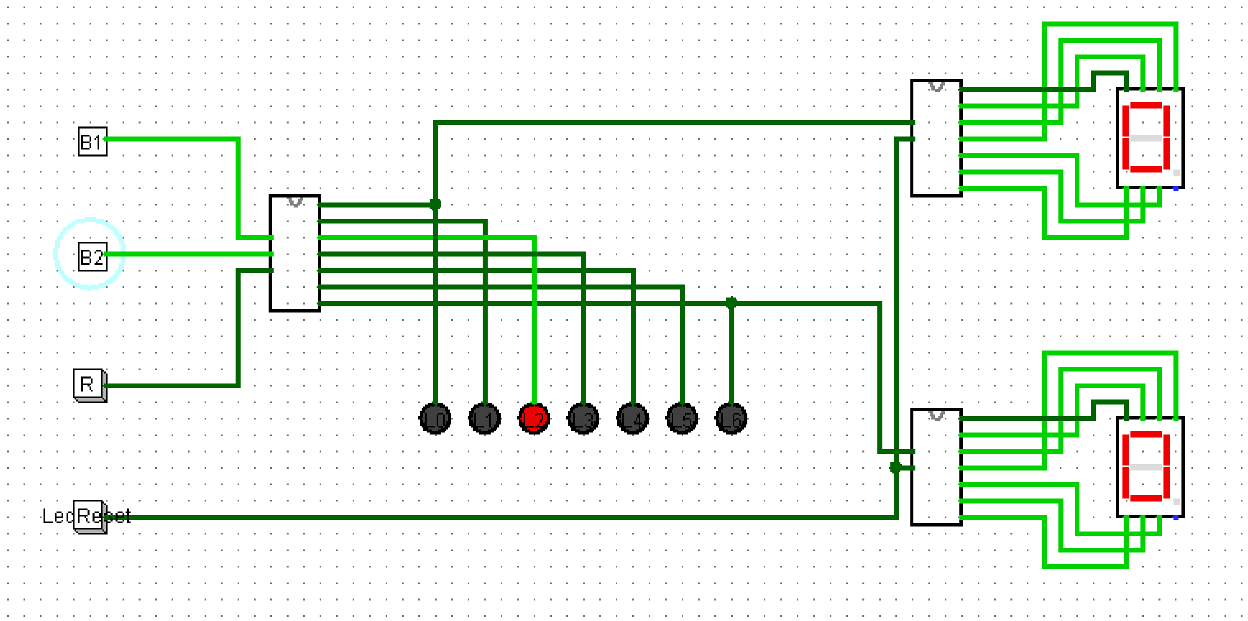
Press Reset: Game Start



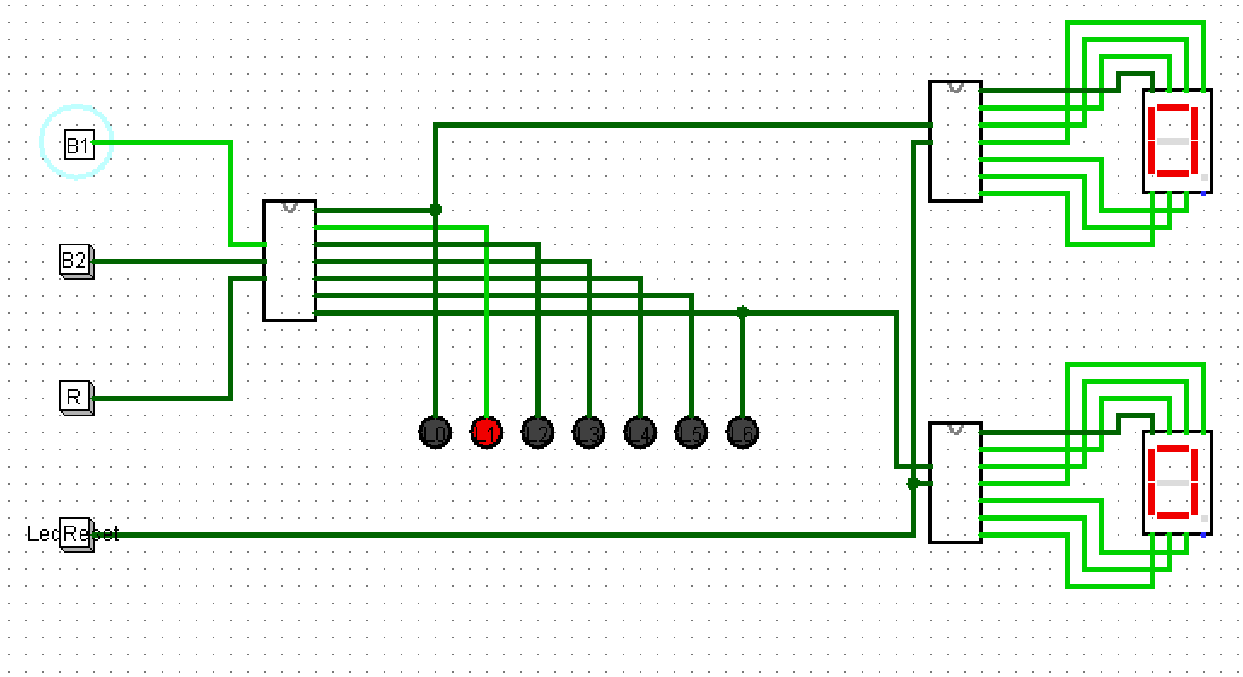
Press B1: Goes the left side



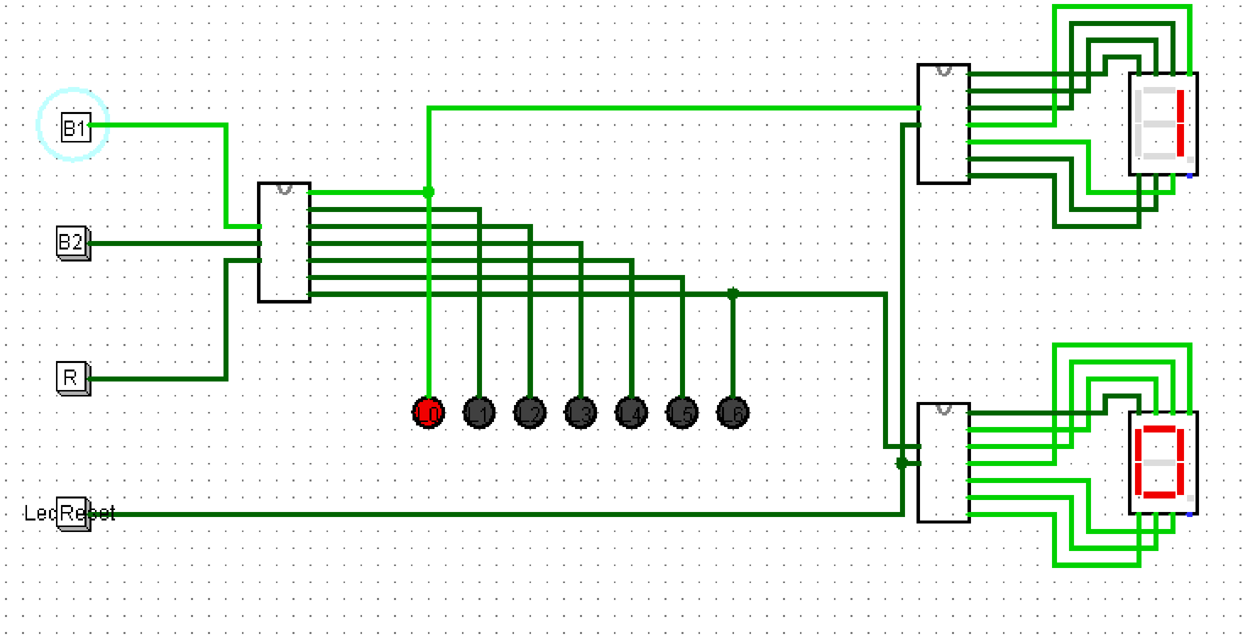
Press B1 and B2: Nothing changed



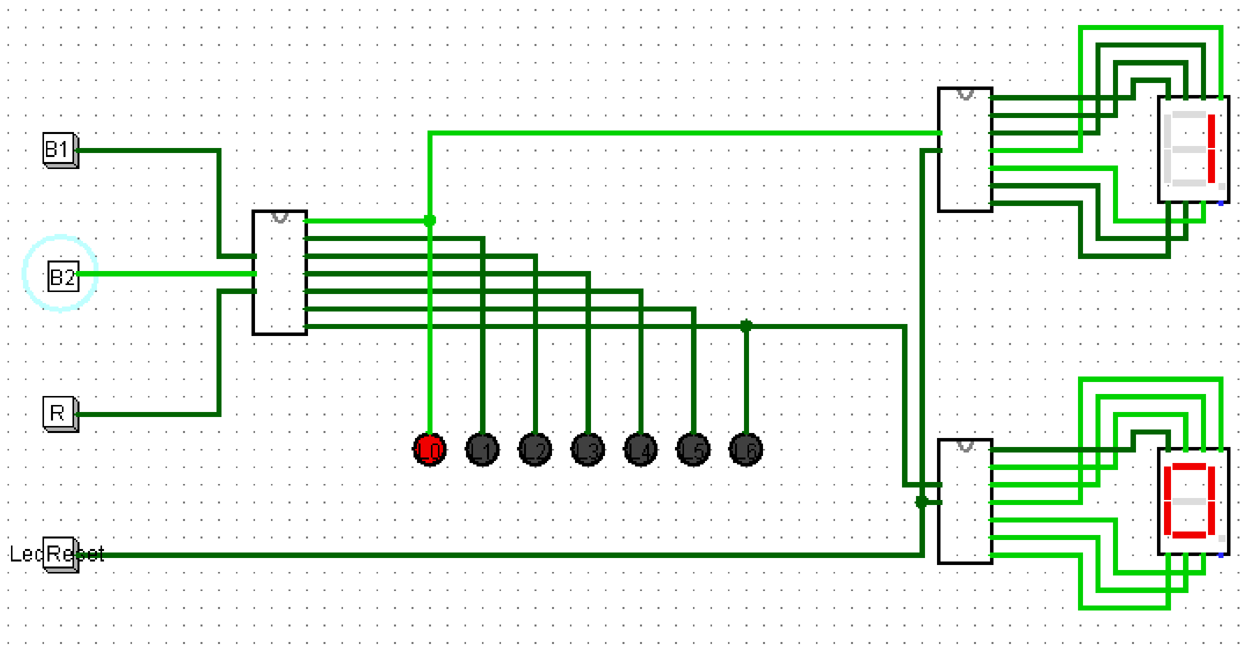
Press B1: Goes left



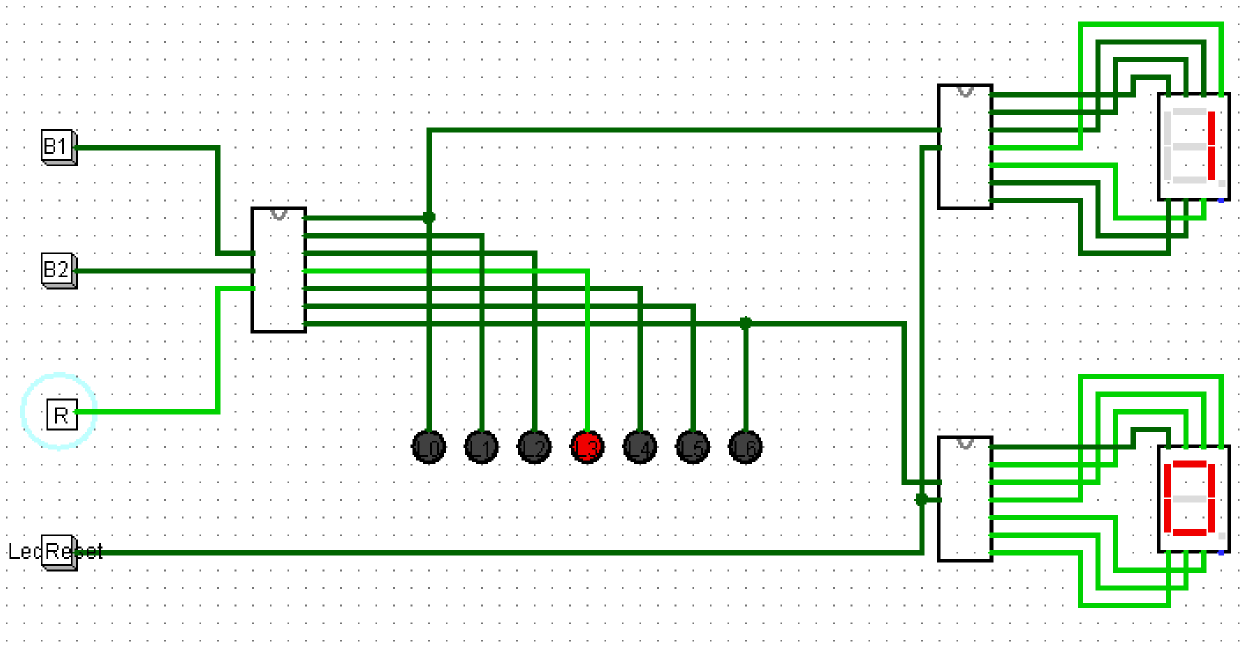
Press B1: Goes L0 and first led displays 1



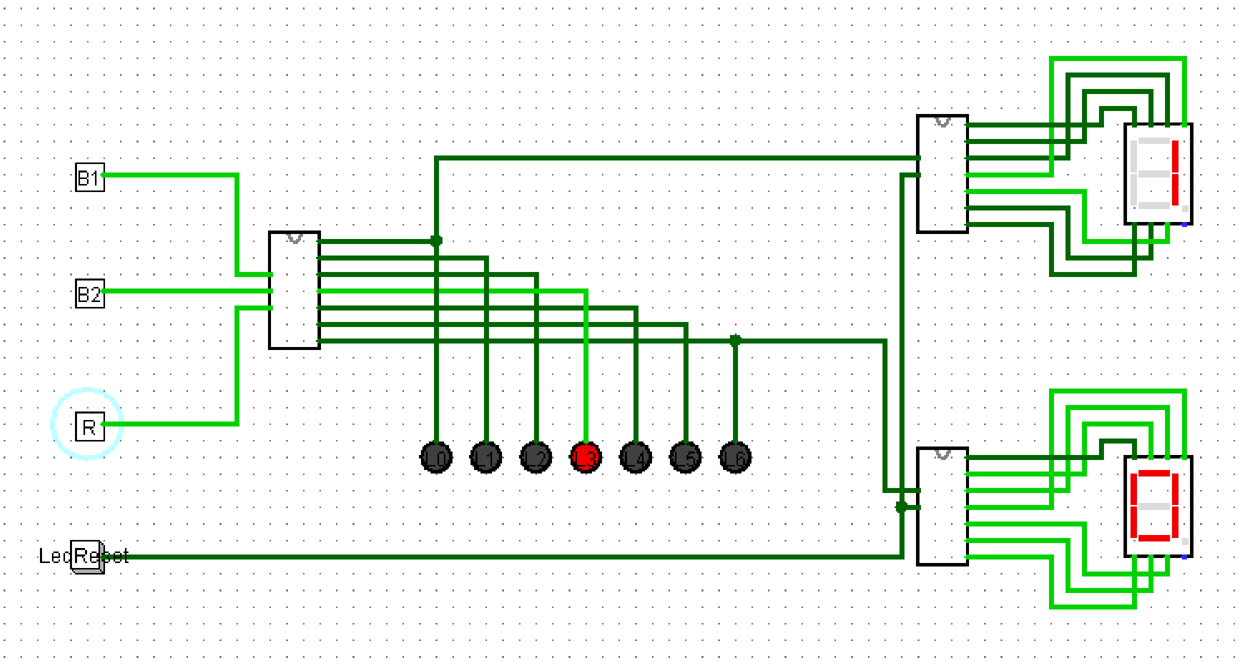
Press B2: Nothing changes because player 1 won.



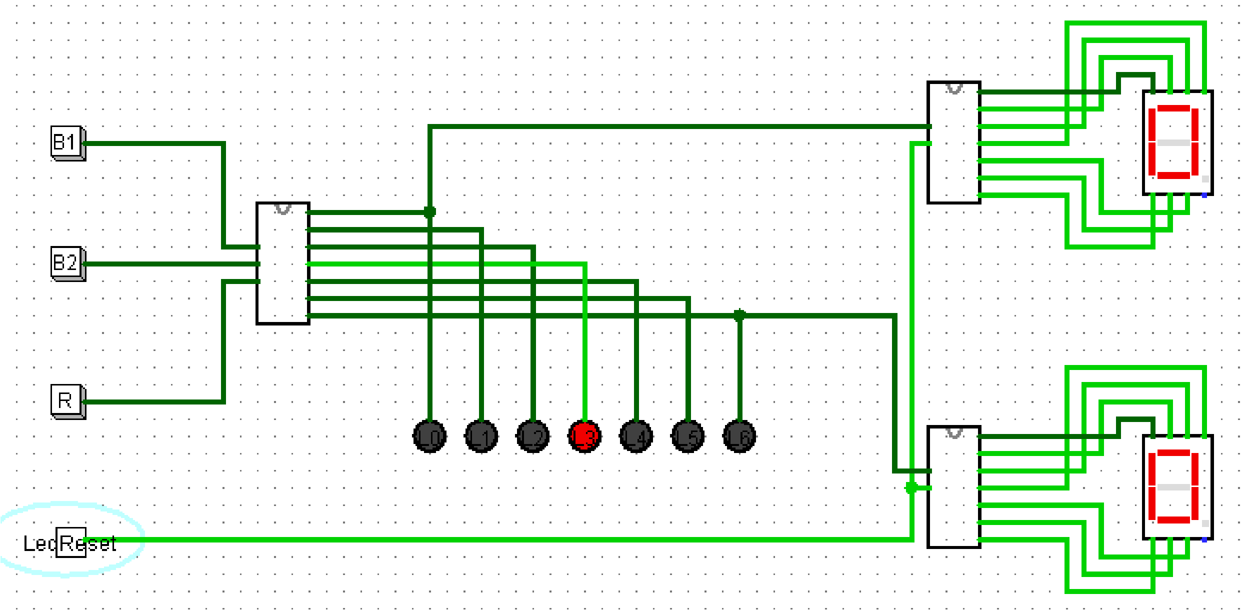
Press R: Resets light



Press B1, B2 and R: Resets light

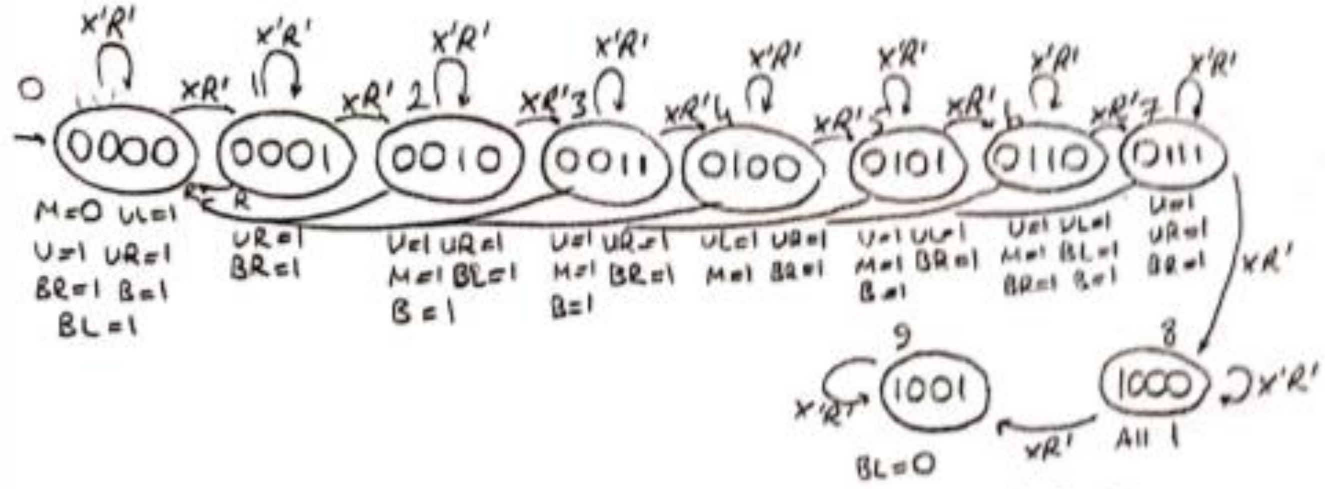


Press Led_Reset: Both leds display 0



Inputs : R , X, S3,S2, S1,S0

Outputs : n3,n2,n1,n0, M, UL, U, UR, BR, B, BL



s3	s2	s1	s0	R	X	n3	n2	n1	n0	M	UL	U	UR	BR	B	BL	
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0
0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	0	0	0	1	1	0	0	0	0								
0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	
0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	0	0	1	1	1	0	0	0	0								
0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	
0	0	1	0	0	1	0	0	1	1	1	0	1	1	0	1	1	2
0	0	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	0	1	0	1	1	0	0	0	0								
0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	
0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	0	3
0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	0	1	1	1	1	0	0	0	0								
0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	
0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	4
0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	1	0	0	1	1	0	0	0	0								

s3	s2	s1	s0	R	x	n3	n2	n1	n0	M	UL	U	UR	BR	B	BL	
0	1	0	1	0	0	0	1	0	1	1	1	1	1	1	1	1	
0	1	0	1	0	1	0	1	1	0	1	1	1	0	1	1	0	
0	1	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	5
0	1	0	1	1	1	0	0	0	0	1	1	1	1	1	1	1	
0	1	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1	
0	1	1	0	0	1	0	1	1	1	1	1	1	0	1	1	1	6
0	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	
0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	
0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	0	0	7
0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	
0	1	1	1	1	1	0	0	0	0								
	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	8
	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1	
	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	
	0	0	0	1	1	0	0	0	0								
	0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	1	9
	0	0	1	0	1	1	0	0	1	1	1	1	1	1	1	0	
	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	
	0	0	1	1	1	0	0	0	0								

I didnot convert this to boolean expressions. In Logisim program, t's automatically converts truth table to logic gates.