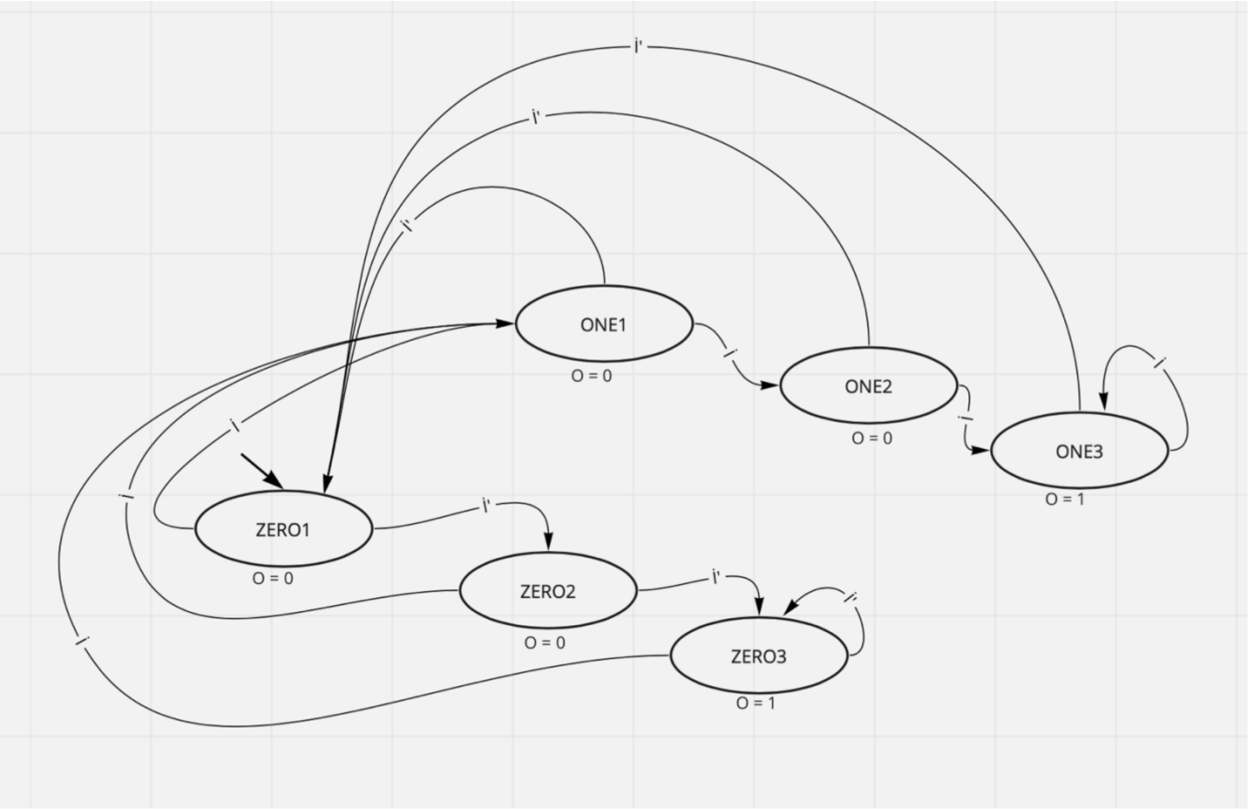
Student Names: Orkun Mahir Kılıç – Esad Yusuf Atik
Student IDs: 2019400210 - 2020400261
Group ID: 9
Session: 1

CMPE 240 2022 Experiment 5 Preliminary Work

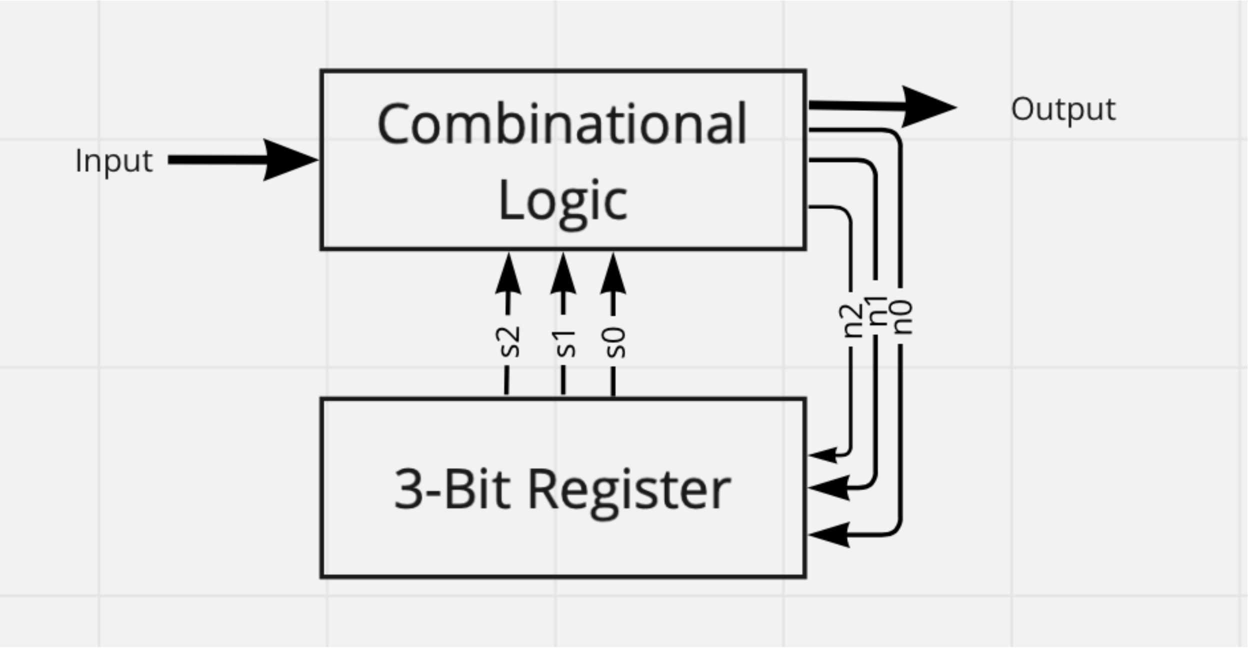
(For illustrations you can use any drawing tool that you want including Microsoft Word Shapes. Do not use scanned images of hand drawn state machines and architecture diagrams.)
(For tables please use insert table feature of Microsoft Word)

Step 1: Capture the FSM: Create and draw the finite state machine that describes the desired behavior of the controller.



Student Names: Orkun Mahir Kılıç – Esad Yusuf Atik
Student IDs: 2019400210 - 2020400261
Group ID: 9
Session: 1

Step 2: Create the architecture: Create and draw standard architecture by a using state register of the appropriate width and combinational logic. Refer to book or lecture slides. Use the same convention.



Step 3: Encode the states: Assign a unique binary number to each state. Each binary number representing a state is known as an encoding. Any encoding will do as long as each state has a unique encoding. (The content of the following table is an example. Rename the states according to the ones you specified in the first section. Also use any encoding you want.)

STATE NAME	ENCODING
ZERO1	000
ZERO2	001
ZERO3	010
ONE1	011
ONE2	100
ONE3	101

Student Names: Orkun Mahir Kılıç – Esad Yusuf Atik
Student IDs: 2019400210 - 2020400261
Group ID: 9
Session: 1

Step 4: Create the state table: Create a truth table for the combinational logic such that the logic will generate the correct FSM outputs and next state signals. Ordering the inputs with state bits first makes this truth table describe the state behavior, so the table is a state table. (Update the table according to the number of state variables that you have used.)

CURRENT STATE s2s1s0	**INPUTS(X)**	**NEXT STATE N2n1n0**	**OUTPUTS(Y)**
000	0	001	0
000	1	011	0
001	0	010	0
001	1	011	0
010	0	010	1
010	1	011	0
011	0	000	0
011	1	100	0
100	0	000	0
100	1	101	0
101	0	000	0
101	1	101	1

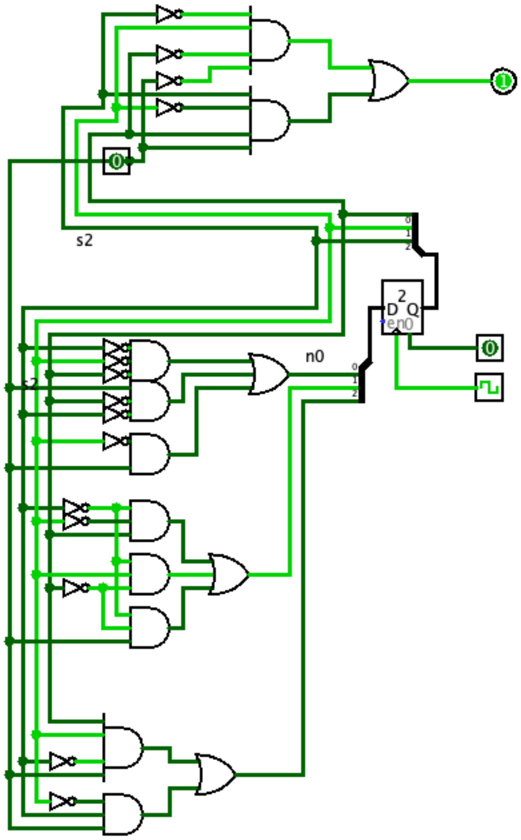
N0 = s2’s1’s0’x’ +s2’s1’s0’x +s2’s1’s0x +s2’s1s0’x+s2s1’s0’x +s2s1’s0x
=S2’S1’S0’ +S2’S0’X+S1’X

Output -> s2’s1s0’i’ + s2s1’s0i
N2 = s2’s1s0i + s2s1’s0’i + s2s1’s0i
= s2’s1s0i + s2s1’i(s0’+s0)
= s2’s1s0i + s2s1’i
N1 -> S2’s1’s0 + s2’s1s0’ + s2’s0’x

N0 -> S2’S1’S0’ +S2’S0’X+S1’X

Student Names: Orkun Mahir Kılıç – Esad Yusuf Atik
Student IDs: 2019400210 - 2020400261
Group ID: 9
Session: 1

Step 5: Draw the combinational logic: Implement the combinatorial logic using any method (You do not need to draw the inside circuit of multiplexers or decoders if you are using any. You can show those as blocks).


1
0
s2
0
1
2
2
D
Q
en0
0
n0
0
1
2