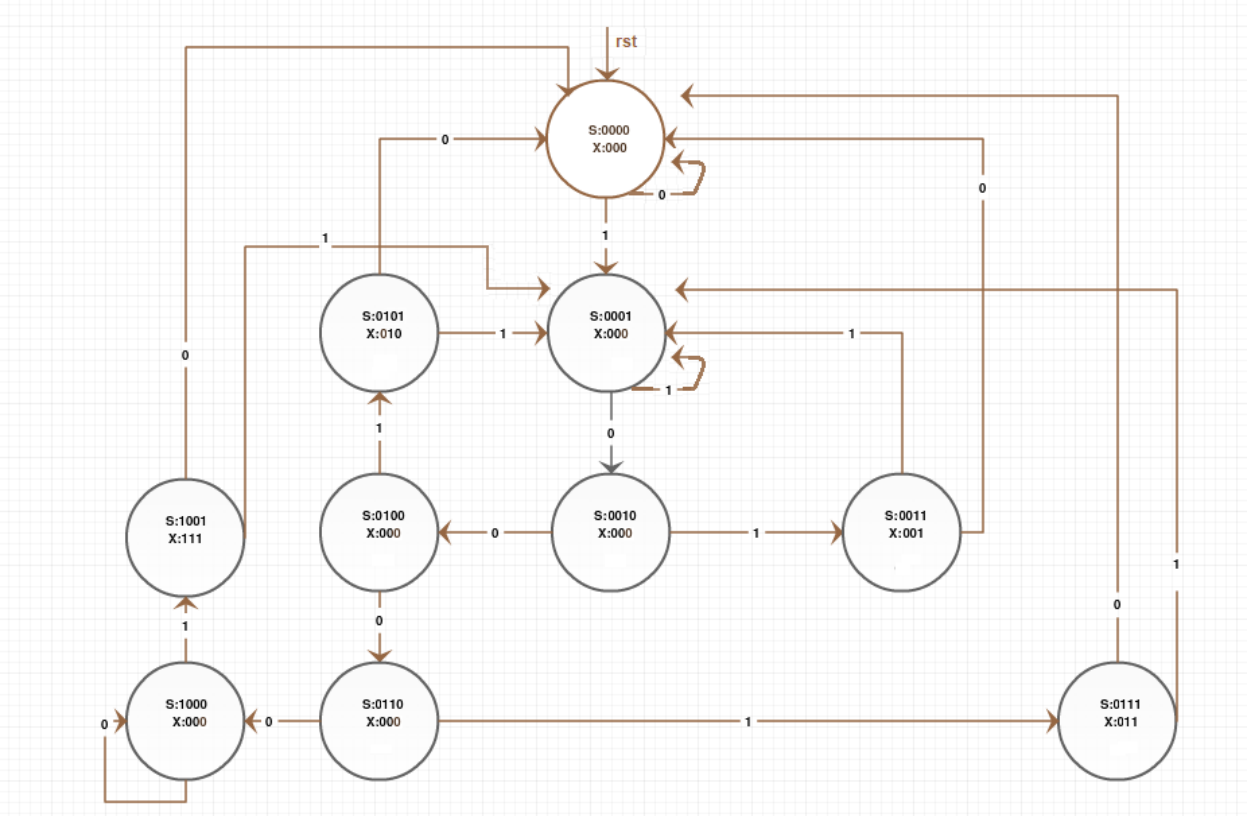
Student Names: Ahmet Emir KOCAAĞA, Mete Han KURT
Student IDs: 2017400276, 2016400339
Group ID: 1
Session: FF34

CMPE 240 2019 Experiment 4 Preliminary Work

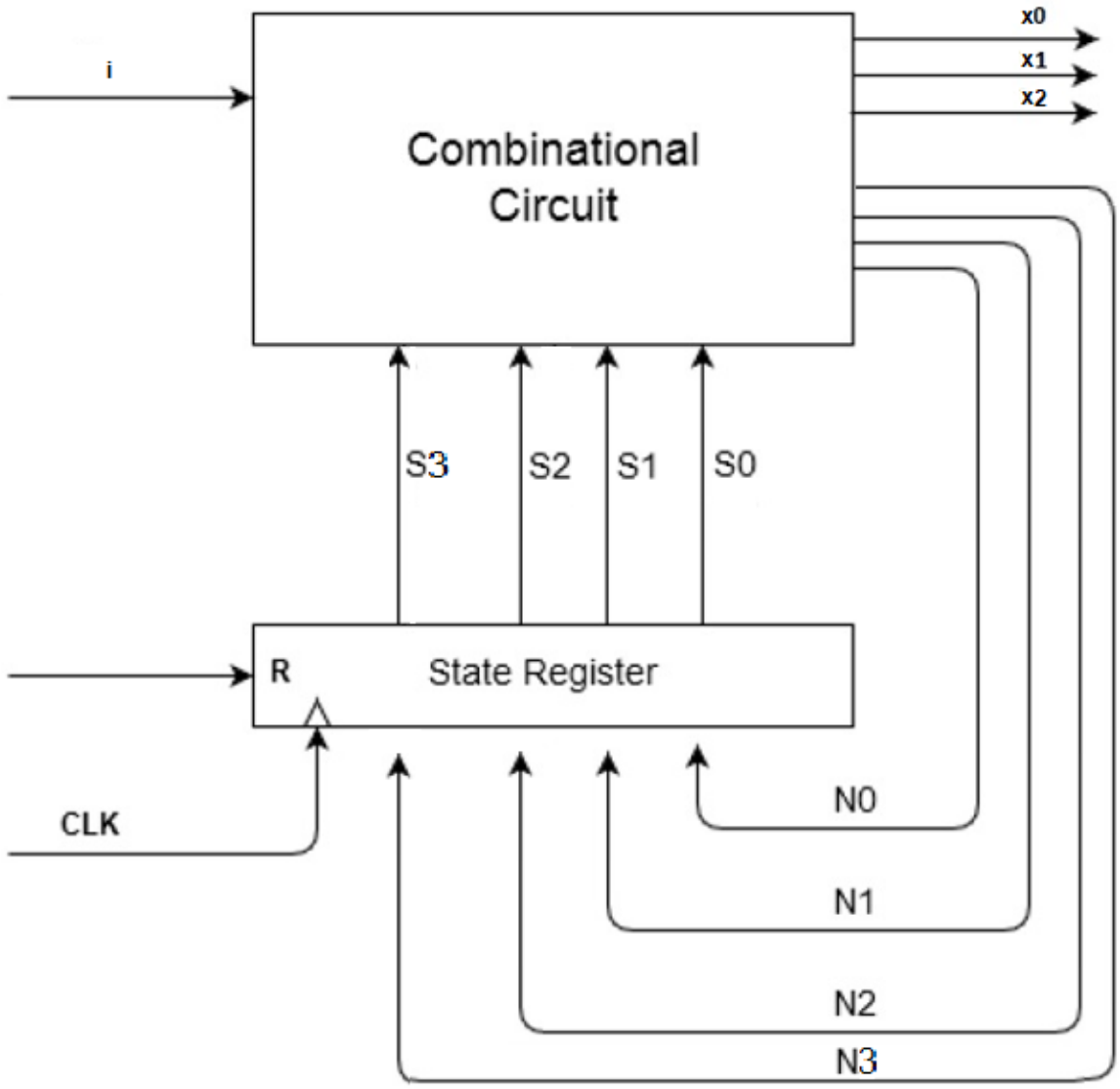
(For illustrations you can use any drawing tool that you want including Microsoft Word Shapes. Do not use scanned images of hand drawn state machines and architecture diagrams.)
(For tables please use insert table feature of Microsoft Word)

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



Student Names: Ahmet Emir KOCAAĞA, Mete Han KURT
Student IDs: 2017400276, 2016400339
Group ID: 1
Session: FF34

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.



Student Names: Ahmet Emir KOCAAĞA, Mete Han KURT
Student IDs: 2017400276, 2016400339
Group ID: 1
Session: FF34

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
S(0)	0000
S(1)	0001
S(2)	0010
S(3)	0011
S(4)	0100
S(5)	0101
S(6)	0110
S(7)	0111
S(8)	1000
S(9)	1001
S(10)	1010
S(11)	1011
S(12)	1100
S(13)	1101
S(14)	1110
S(15)	1111

Student Names: Ahmet Emir KOCAAĞA, Mete Han KURT
Student IDs: 2017400276, 2016400339
Group ID: 1
Session: FF34

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	INPUTS	NEXT STATE	OUTPUTS
0000	0	0000	000
0000	1	0001	000
0001	0	0010	000
0001	1	0001	000
0010	0	0100	000
0010	1	0011	000
0011	0	0000	001
0011	1	0001	001
0100	0	0110	000
0100	1	0101	000
0101	0	0000	010
0101	1	0001	010
0110	0	1000	000
0110	1	0111	000
0111	0	0000	011
0111	1	0001	011
1000	0	1000	000
1000	1	1001	000
1001	0	0000	111
1001	1	0001	111
1010	0	0000	000
1010	1	0000	000
1011	0	0000	000
1011	1	0000	000
1100	0	0000	000
1100	1	0000	000
1101	0	0000	000
1101	1	0000	000

Student Names: Ahmet Emir KOCAAĞA, Mete Han KURT
Student IDs: 2017400276, 2016400339
Group ID: 1
Session: FF34

1110	0	0000	000
1110	1	0000	000
1111	0	0000	000
1111	1	0000	000

Student Names: Ahmet Emir KOCAAĞA, Mete Han KURT
Student IDs: 2017400276, 2016400339
Group ID: 1
Session: FF34

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).