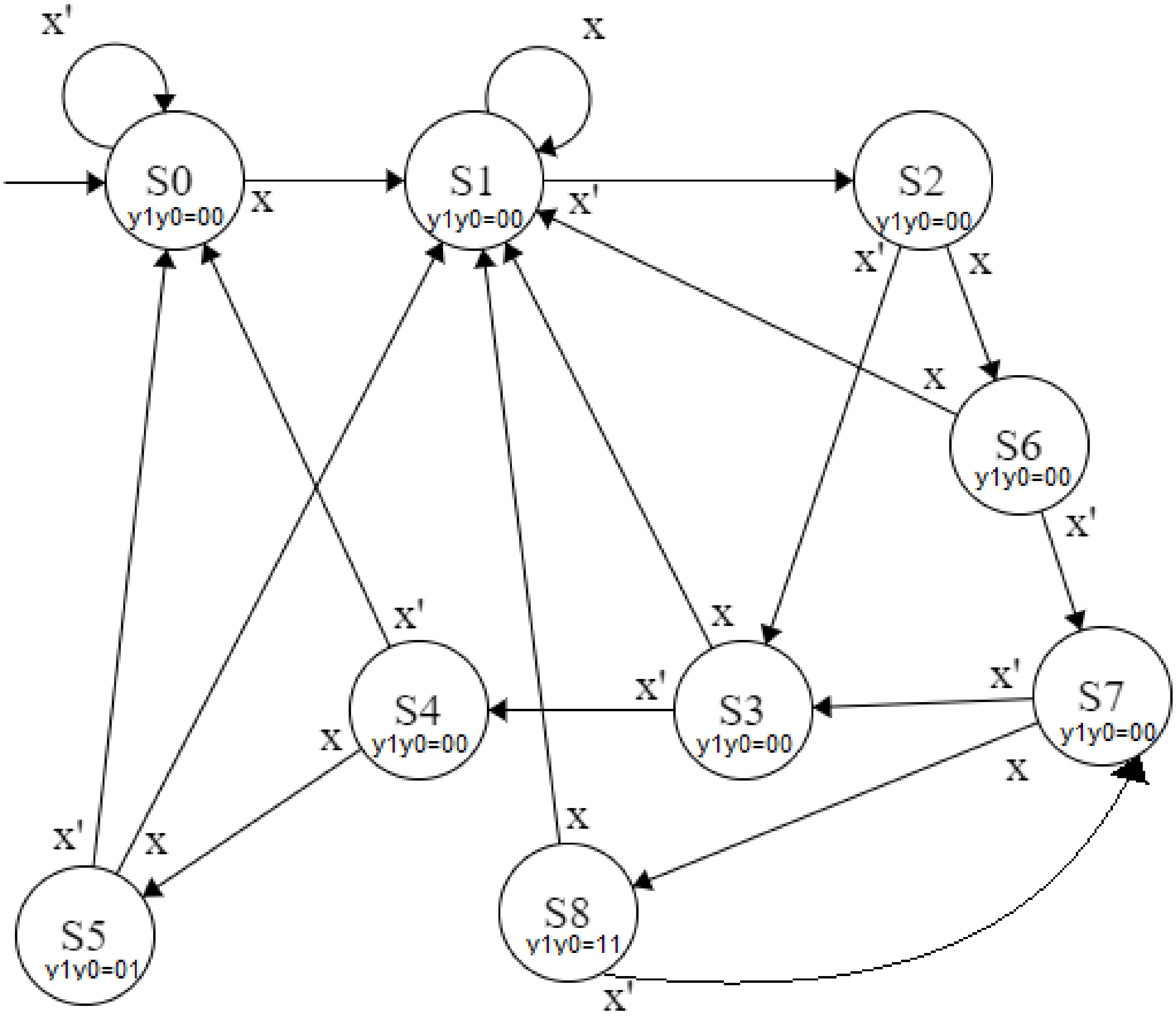
Student Names: EGE CAN KAYA
Student IDs: 2018400018
Group ID: 29
Session: 2

CMPE 240 2020 Experiment 4 Preliminary Work

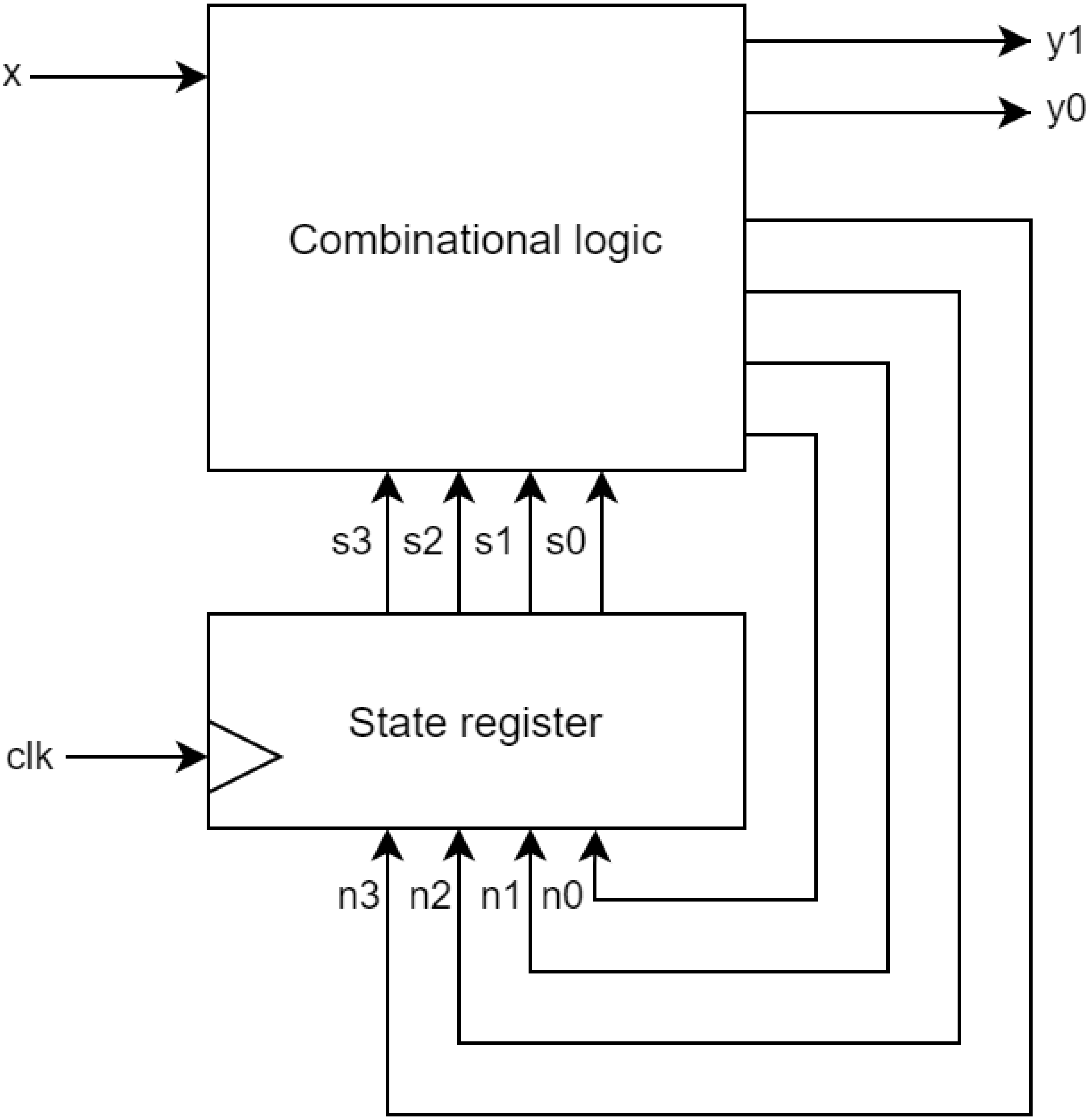
(For illustrations you can use any drawing tool that you want including Microsoft Word Shapes. Do not use scanned images of hand drawn state machines and architecture diagrams.)
(For tables please use insert table feature of Microsoft Word)

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



Student Names: EGE CAN KAYA
Student IDs: 2018400018
Group ID: 29
Session: 2

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: EGE CAN KAYA
Student IDs: 2018400018
Group ID: 29
Session: 2

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
S0	0000
S1	0001
S2	0010
S3	0011
S4	0100
S5	0101
S6	0110
S7	0111
S8	1000

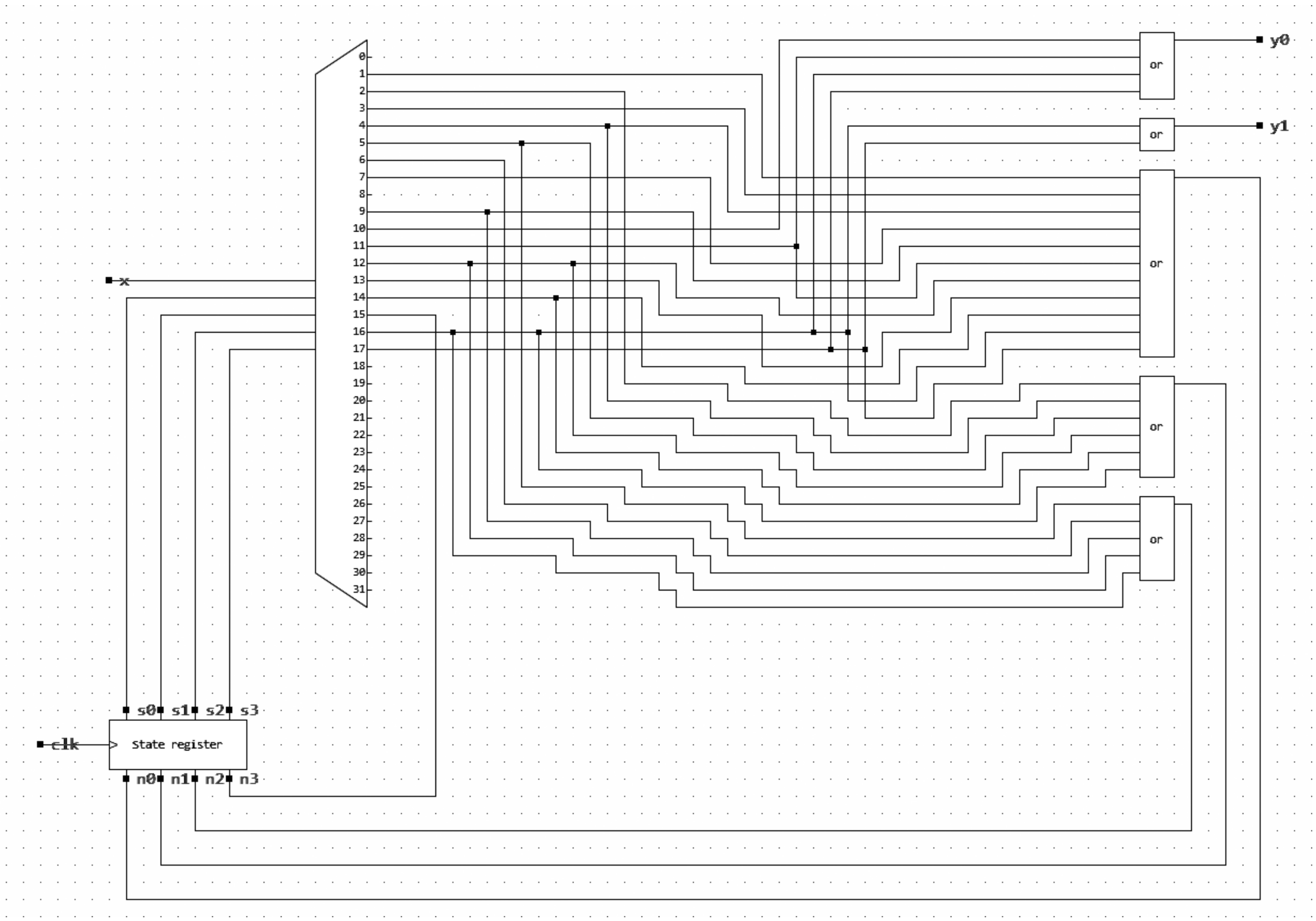
Student Names: EGE CAN KAYA
Student IDs: 2018400018
Group ID: 29
Session: 2

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(X)	NEXT STATE	OUTPUTS(Y1Y0)
0000	0	0000	00
0000	1	0001	00
0001	0	0010	00
0001	1	0001	00
0010	0	0011	00
0010	1	0110	00
0011	0	0100	00
0011	1	0001	00
0100	0	0000	00
0100	1	0101	00
0101	0	0000	01
0101	1	0001	01
0110	0	0111	00
0110	1	0001	00
0111	0	0011	00
0111	1	1000	00
1000	0	0111	11
1000	1	0001	11
1001	0	0000	00
1001	1	0000	00
1010	0	0000	00
1010	1	0000	00
1011	0	0000	00
1011	1	0000	00
1100	0	0000	00
1100	1	0000	00
1101	0	0000	00
1101	1	0000	00
1110	0	0000	00
1110	1	0000	00
1111	0	0000	00
1111	1	0000	00

Student Names: EGE CAN KAYA
Student IDs: 2018400018
Group ID: 29
Session: 2

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).


y0
y1
or
or
or
or
or
x
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
s0
s1
s2
s3
clk
State register
n0
n1
n2
n3