Homework 4: Finite State Machine Simplifications

Objective

The objective of this homework assignment is to have you design and simplify a moderately complex finite state machine.

Assignment

For this project you will design and implement in Logisim a Mealy finite state machine with one input bit x and one output bit z. The machine's output z is 1 whenever the input sequence …0111 or …1000 has been detected. The patterns may overlap. For example, if the machine is given input 0111000, the output would be 0001001. [Adapted from *Contemporary Logic Design*, Randy H. Katz, Benjamin/Cummings Publishing, 1994.]

In order to complete the project, you must accomplish the following tasks. You will submit both your design work (on paper) and the Logisim implementation of the resulting finite state machine via submit on GL.

1. The transition diagram for the finite state machine described above is given to you on the next page. Use the state reduction algorithm discussed in class to minimize the number of states in your machine. You should fill in the table of distinguished states on the following page. Some entries are already filled to get you started: we know that states K and L are unique because they are the only states that output 1 on input 1 and 0, respectively.

   At the end of the reduction process you should have 7 states in your machine. Name your new states P, Q, R, S, T, U and V. For each state in the new machine, indicate its corresponding set of equivalent states from the original finite state machine.

2. Draw the transition diagram of your reduced finite state machine. Check it carefully, since an error in this diagram will invalidate the rest of your work.

3. Try 3 *different* state assignments for the reduced finite state machine. A state assignment labels each of the 7 states with a unique bit pattern. There are many ways to do this and the choice of bit patterns can affect the size of the final circuit. Use the heuristics for state assignment presented in class. (Remember that the initial state should be 000 or 111.) Do this step carefully as it will have a large effect on the complexity of your final circuit.

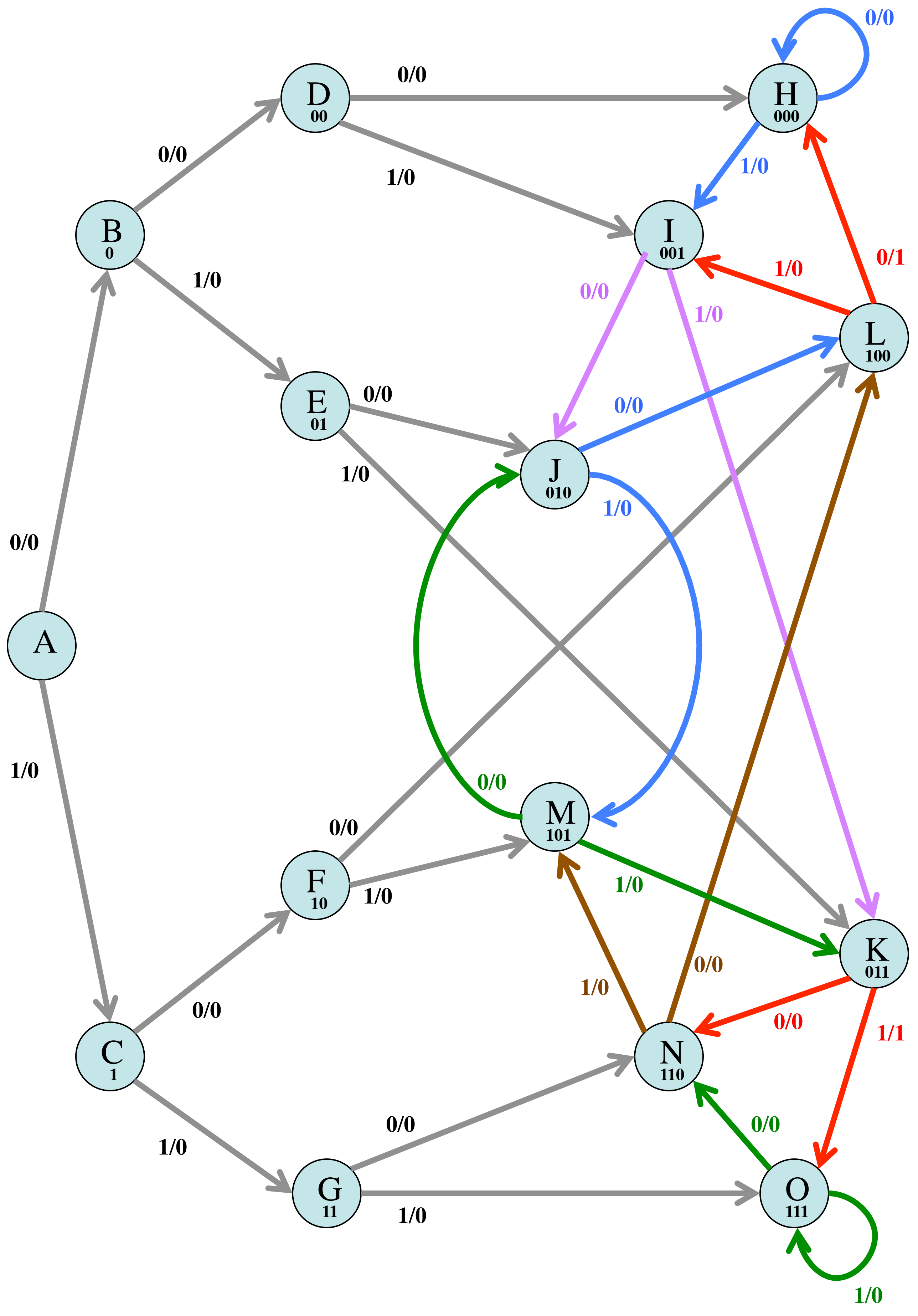
   Hint: follow the loops in the FSM and try to assign bit patterns such that in each step of the loop only one state bit changes. You might not be able to achieve this for *every* step of the loop, but you can try to have only one state bit change for *most* steps of the loop.

4. For each state assignment, fill in a truth table for the finite state machine. In these truth tables, s2, s1 and s0 are the state bits, x is the input bit and z is the output bit. Then s2', s1' and s0' are the next states to be stored in the flip flops.

5. Use the Karnaugh maps provided to simplify the Boolean formulas for each state assignment.

6. Decide which of the 3 state assignments gives the simplest circuit. *Hint:* it is possible to implement this finite state machine with just 14 gates. If your circuit requires many more gates, redo the simplification steps.

7. Implement the resulting circuit in Logism. You may start by modifying the sequence detector shown in class.


A
B
0
C
1
D
00
E
01
F
10
G
11
H
000
I
001
J
010
K
011
L
100
M
101
N
110
O
111
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/0
1/0
0/1
1/0
0/0
1/0
0/0
1/0
0/0
1/1
0/0
1/0

CMSC 313 Homework 4: State Reduction

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	╳										X	X			
B	╳	╳									X	X			
C	╳	╳	╳								X	X			
D	╳	╳	╳	╳							X	X			
E	╳	╳	╳	╳	╳						X	X			
F	╳	╳	╳	╳	╳	╳					X	X			
G	╳	╳	╳	╳	╳	╳	╳				X	X			
H	╳	╳	╳	╳	╳	╳	╳	╳			X	X			
I	╳	╳	╳	╳	╳	╳	╳	╳	╳		X	X			
J	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	X	X			
K	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	X	X	X	X
L	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	X	X	X
M	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳		
N	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	
O	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳	╳

Indicate the states of the new finite state machine as subsets of states of the original machine

P = { **}**	**T = {** **}**
Q = { **}**	**U = {** **}**
R = { **}**	**V = {** **}**
S = { **}**	

CMSC 313 Homework 4

Name: ________________________________

Minimized 7-State Transition Diagram:

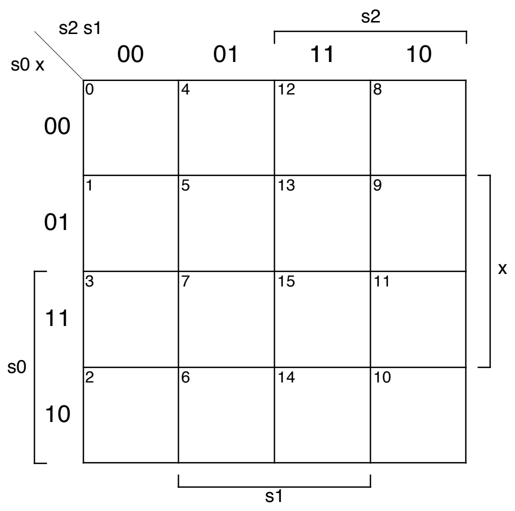
State Assignment #1

State	Bit Pattern
P	
Q	
R	
S	
T	
U	
V	

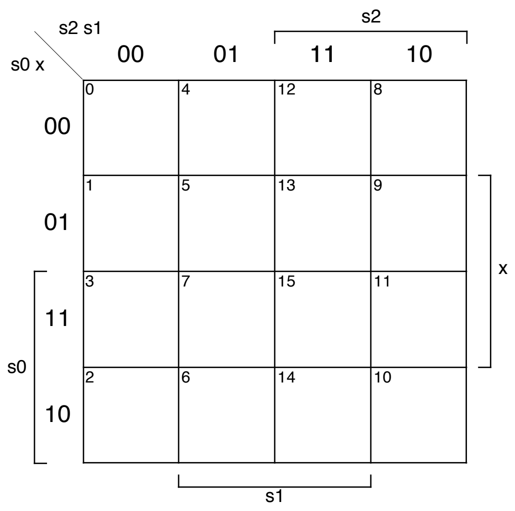
Truth Table #1

m	s2	s1	s0	x	s2'	s1'	s0'	z
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

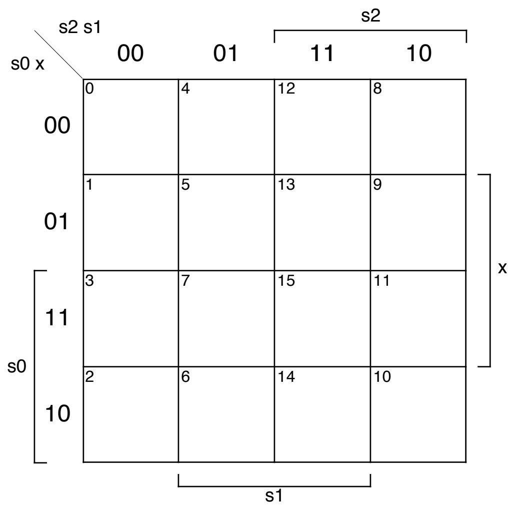
State Assignment #1 Karnaugh Maps



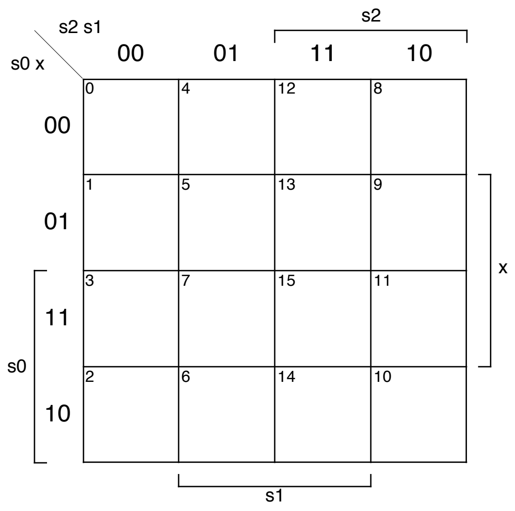
s2’ =



s1’ =



s0’ =



z =

State Assignment #2

State	Bit Pattern
P	
Q	
R	
S	
T	
U	
V	

Truth Table #2

m	s2	s1	s0	x	s2'	s1'	s0'	z
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

State Assignment #2 Karnaugh Maps

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

s2' =

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

s1' =

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

s0' =

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

z =

State Assignment #3

State	Bit Pattern
P	
Q	
R	
S	
T	
U	
V	

Truth Table #3

m	s2	s1	s0	x	s2'	s1'	s0'	z
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

State Assignment #3 Karnaugh Maps

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

s2' =

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

s1' =

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

s0' =

s0 x \ s2 s1	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

z =

CMSC 313 Homework 4

Which state assignment will you use in Logisim? Why?