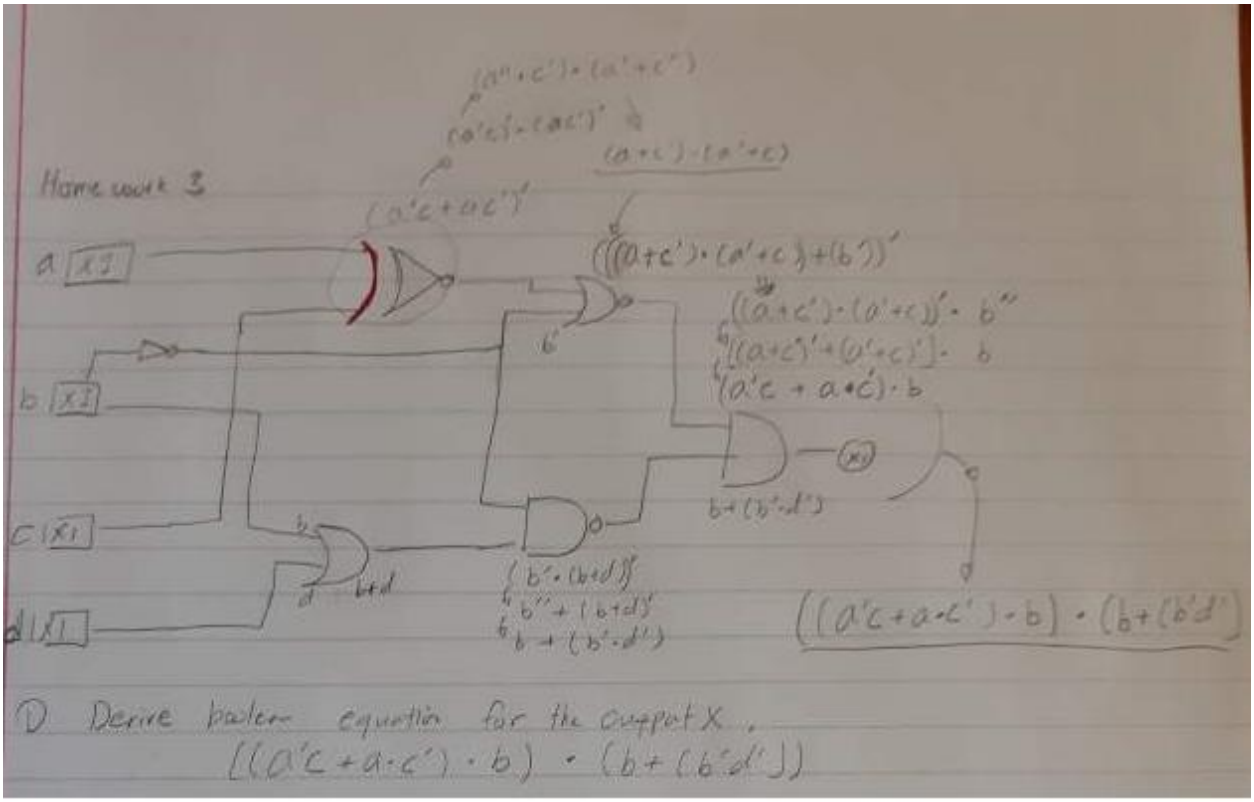
CSS 422

Jun Hyung Park

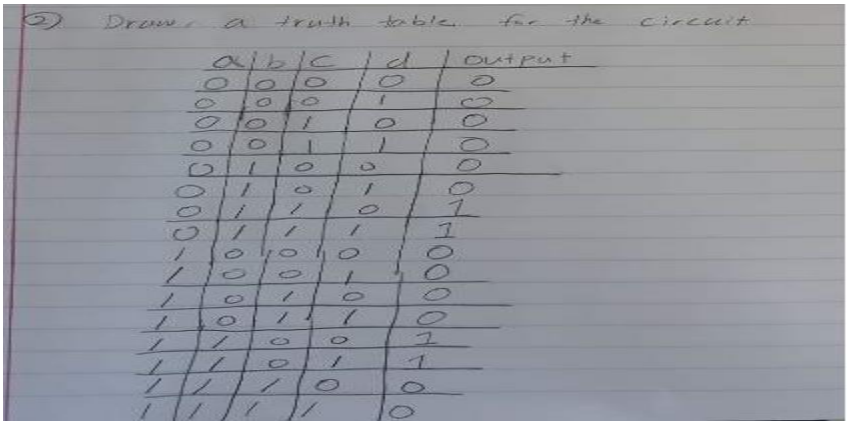
Homework 3

Q1.(5 pts) Consider the logic gate circuit shown below.

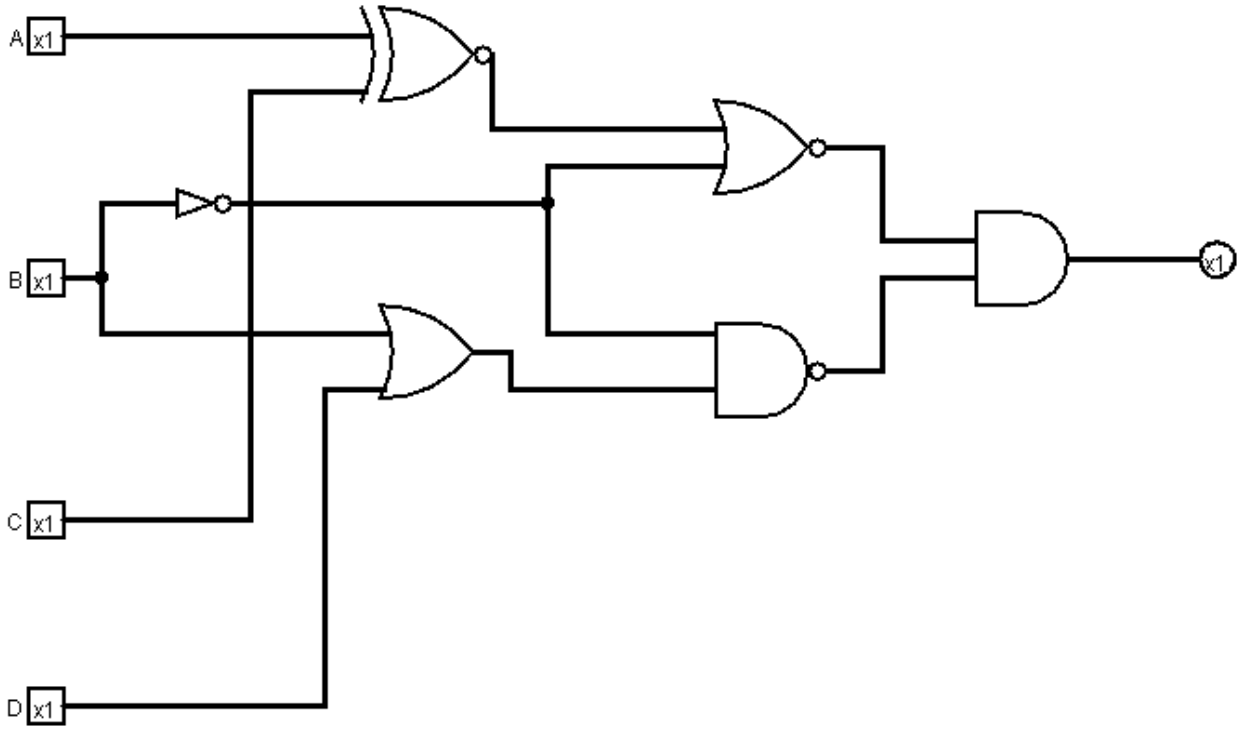
(1) Derive a Boolean equation for the output X. You don't need to simplify this equation, but feel free to try!



(2) Draw a truth table for the circuit.



(3) Make the same circuit with logisim . **Submit the image (export image) and circuit file.**



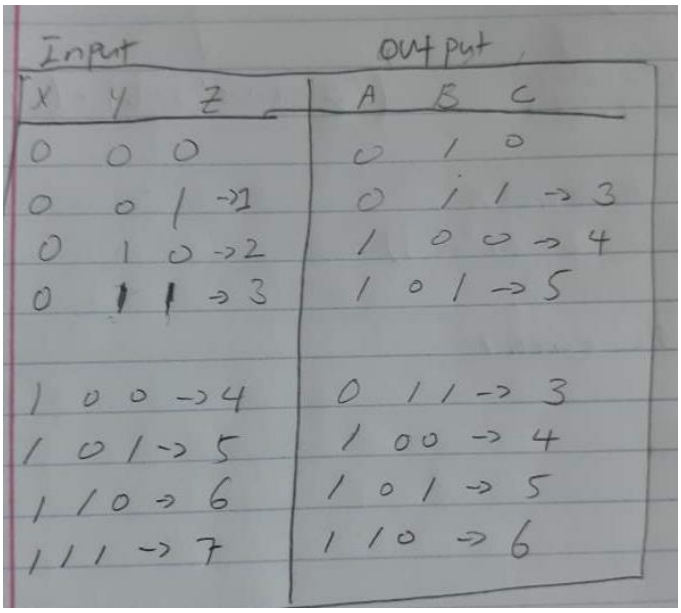
File had been attached in Zip folder

(4) Generate the truth table with logisim(In logisim, project-->analyze circuit-->table). **Copy & past the table in your submission.**

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

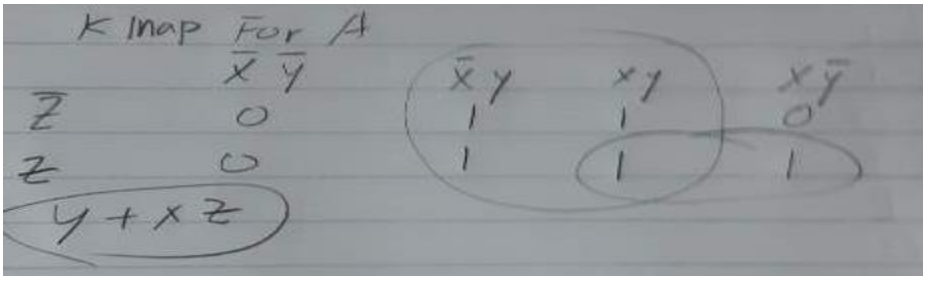
Q2. (5 pts) Design a combinational circuit system.

(1) x,y,z are inputs and A,B,C are outputs.Draw a truth table for the given function.



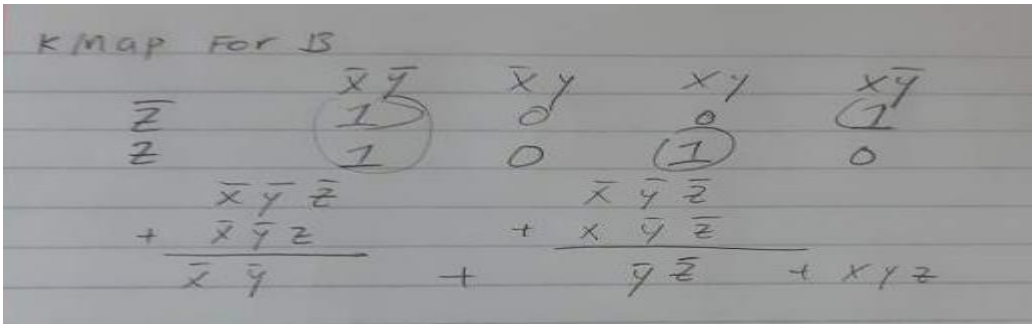
(2) Based on the truth table you draw, build Karnaugh maps for the output A,B,C.

Output For A



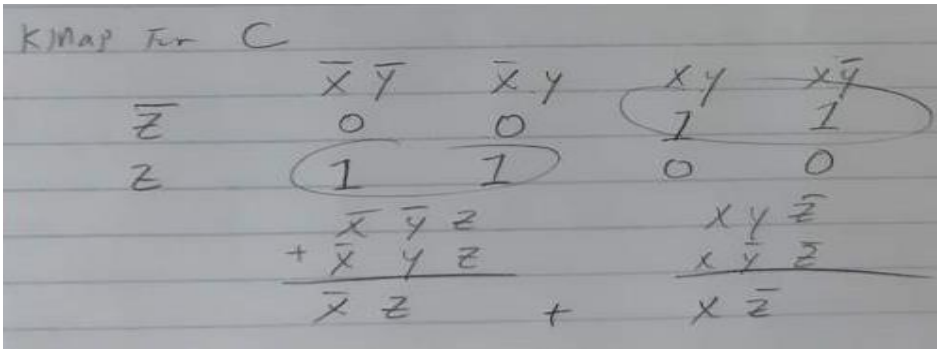
Y + XZ

Output For B



~X~Y + ~Y~Z + XYZ

Output For C



~XZ + X~Z

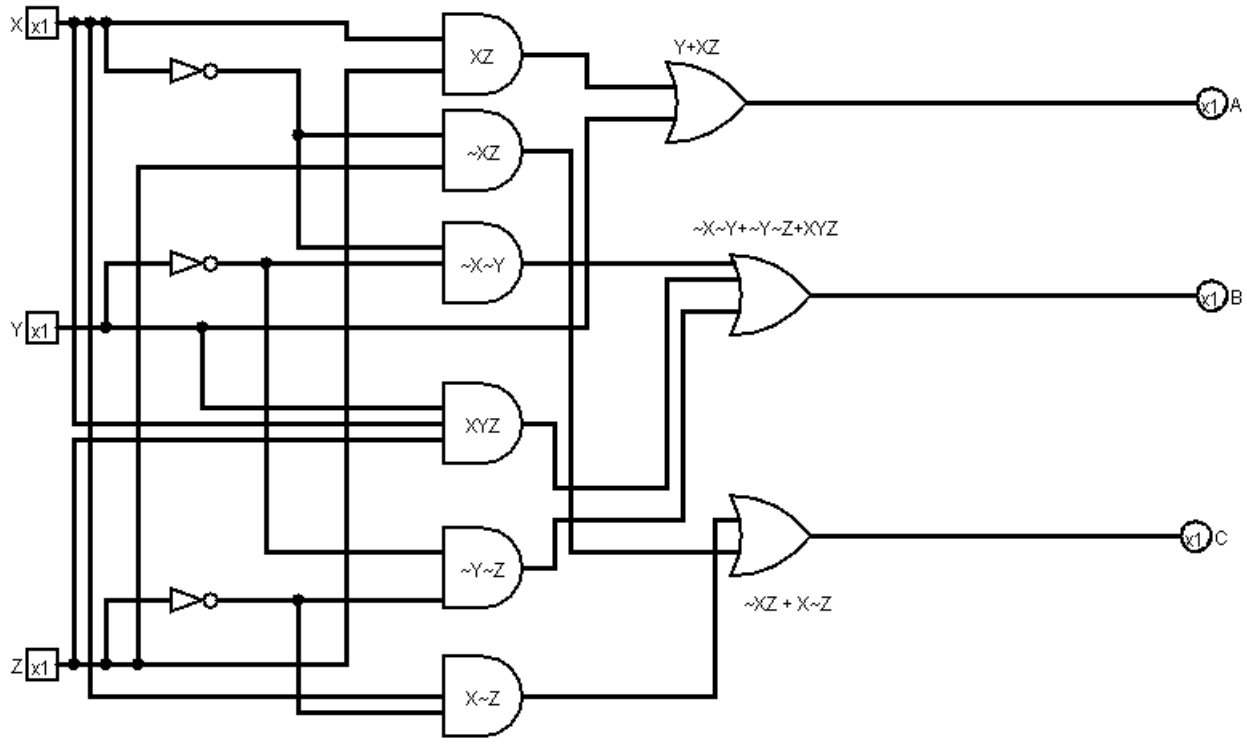
(3) Derive (as simple as possible) Boolean equations for A,B,C using the Karnaugh maps

A) Y + XZ

B) ~X~Y + ~Y~Z + XYZ

C) ~XZ + X~Z

(4) Based on the Boolean equations, draw the logical gate diagram (circuit) for this system in Logisim. **Submit the image (export image) and circuit file.**



Circuit file will be submitted in zip folder.

(5) Test your circuit with the Logisim simulation and generate the truth table (In logisim, project-->analyze circuit-->table). **Copy & past the table, and compare it with your own table from sub-question (2)**

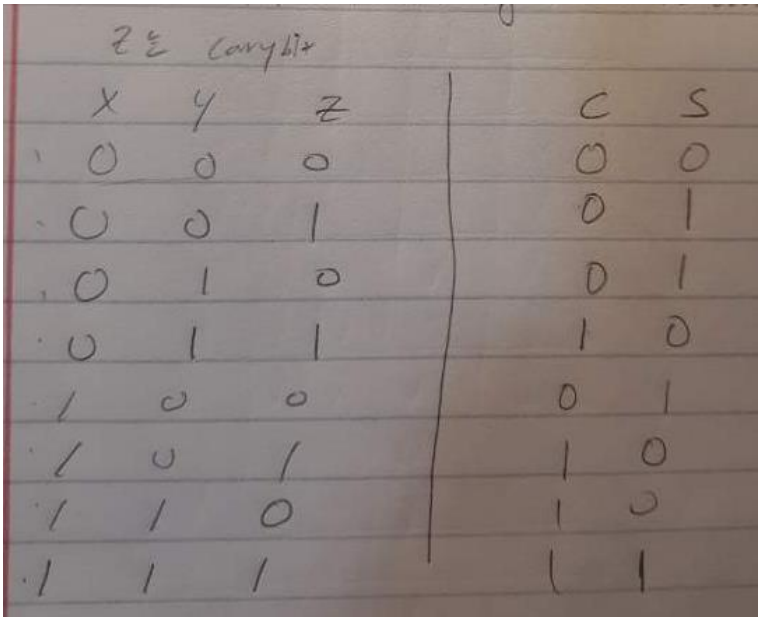
X	Y	Z	A	B	C
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

I did compare, it looks same.

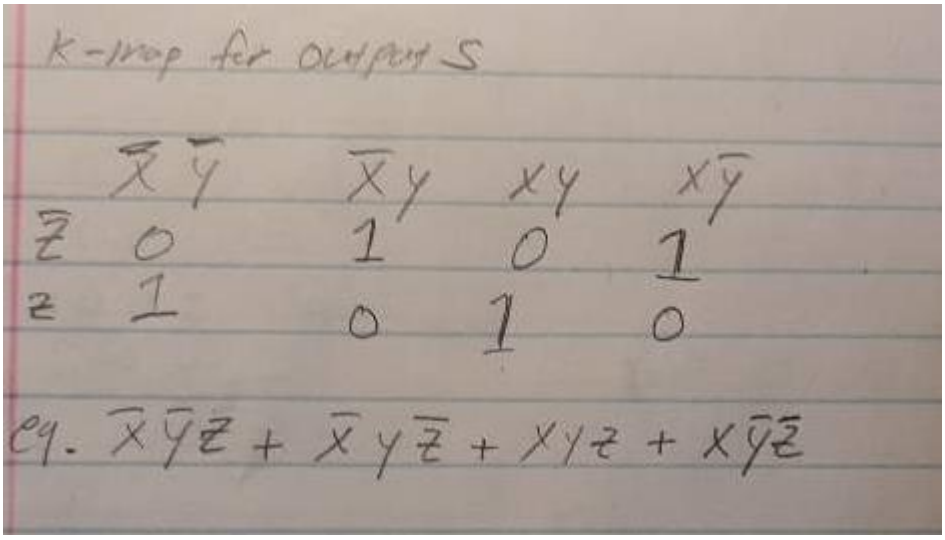
Q3. (5 pts) Combinational circuit and Full Adder.

A full-adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs, x, y, z, and two outputs, C and S. Two of the input, that is, x and y, represent the two significant bits to be added. The third input, z, represents the carry from the previous lower significant position. The output S denotes the sum of two bits and C denotes carry. **Answer the following sub-questions.**

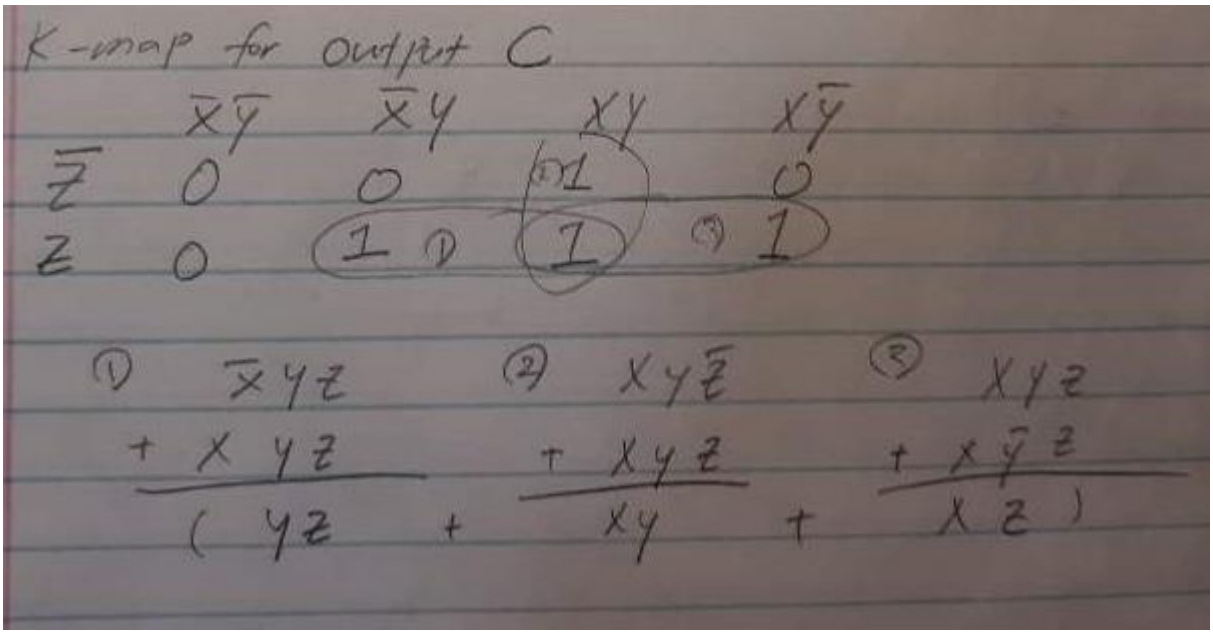
(1) Construct a truth table for Full-Adder



(2) Based on a truth table, construct a K-map for the output S and derive a Boolean equation using K-map. Make the equation as simple as possible



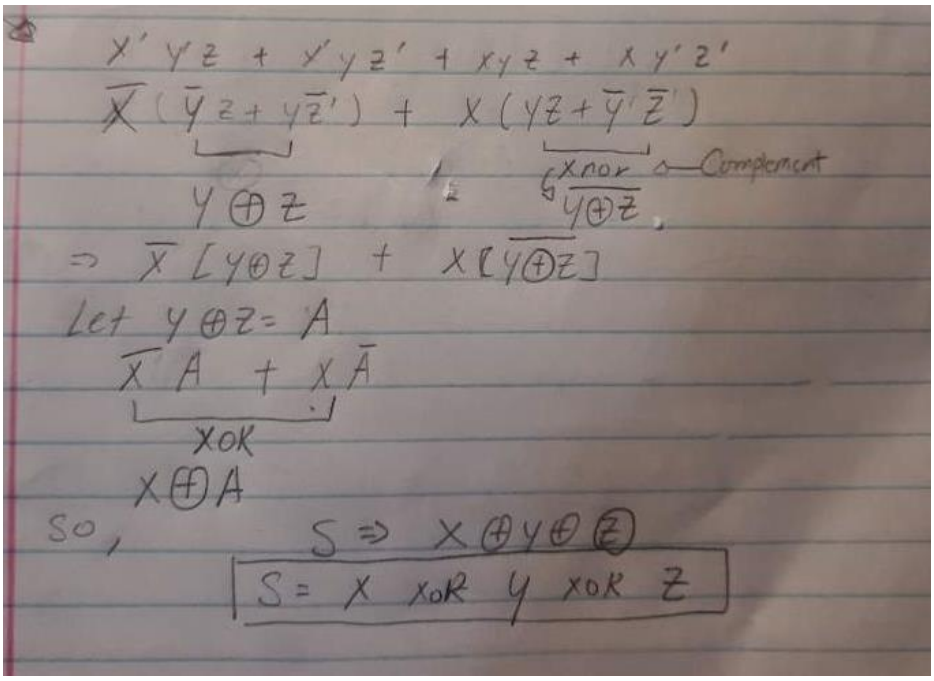
(3) Based on a truth table, construct a K-map for the output C and derive a Boolean equation using K-map. Make the equation as simple as possible



(4) By algebraic manipulation, show that S can be expressed as the exclusive-OR of the three input variables. That is, show that,

S = (x XOR y) XOR z.

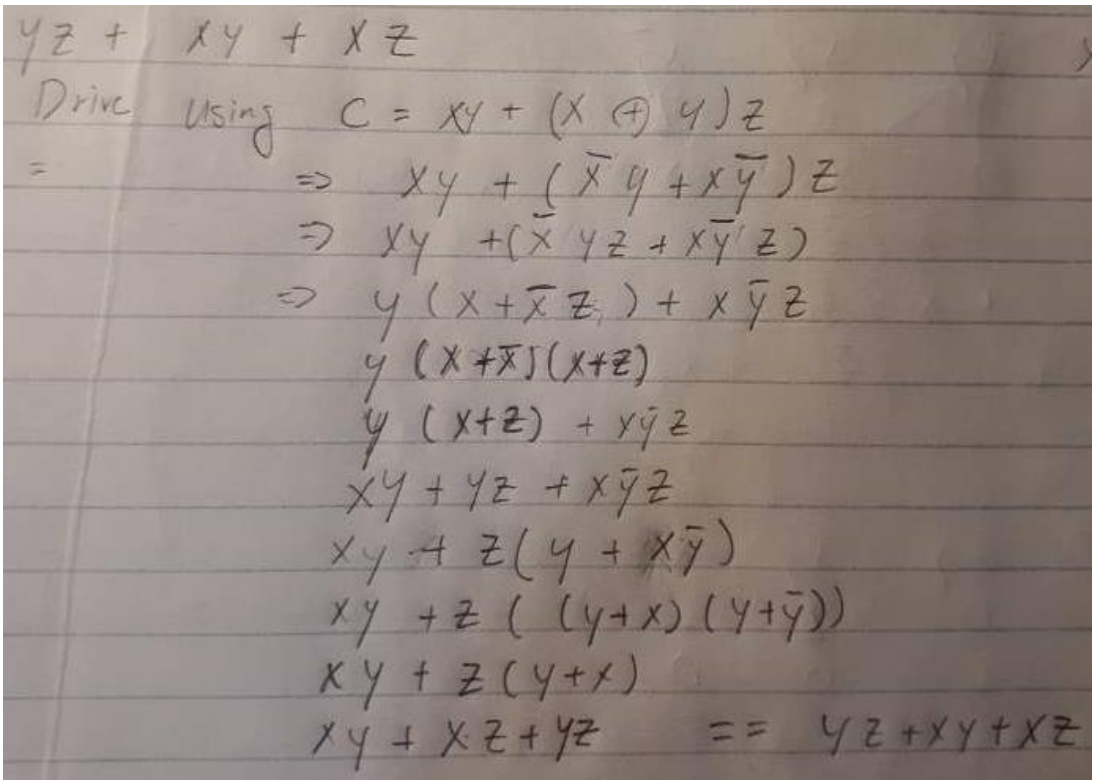
You can prove by deriving from S = (x XOR y) XOR z to the answer you got in the question (2)



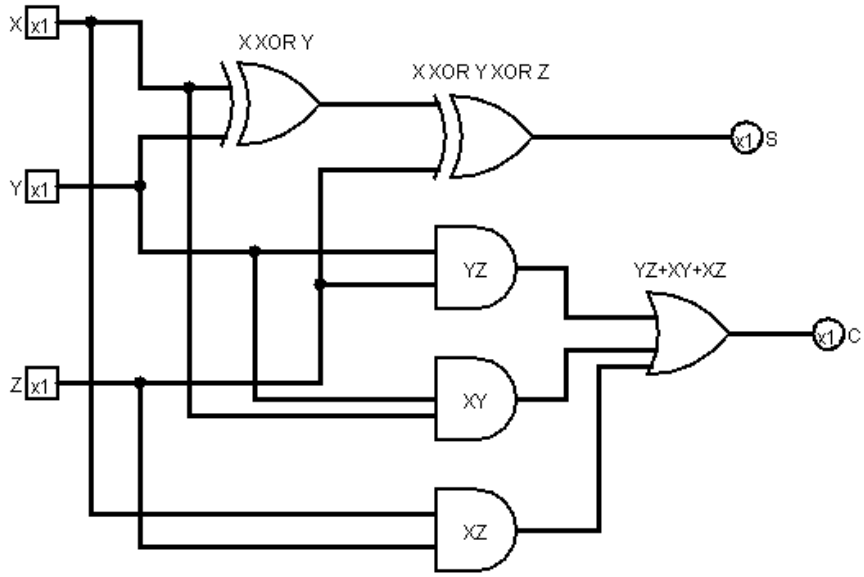
(5) By algebraic manipulation, show that C can be expressed as the following term.

C = xy + (x XOR y)z.

You can prove by deriving from C = xy + (x XOR y)z. to the answer you got in the question (3)



(6) Based on the (4) (5), draw a circuit for the full-adder in Logisim simulator and submit the **image (export the image)** and **circuit file**.



Circuit file will be attached in folder

Q4. (5 pts) Design a sequential circuit.

A sequential circuit has one D flip-flop and one JK-flip-flop, two inputs x and y, and one output z. A and B are the outputs of each D flip-flop, and JK-flip-flop, respectively. The flip-flop ***input*** equations and the circuit output are as follows. Here D_A is the D input of the D-flip flop of A, and J_B, K_B is the J and K input of the JK-flip flop of B.

$D_A = \sim xy + yB$

$J_B = \sim xB + xy$

$K_B = yB + \sim xA$

$z = y + x\sim y$

(1) Draw the logic diagram of the circuit and test it with Logisim. Please attach the **circuit image, circuit file**, and attach the **generated table**.

The circuit file will be attached.

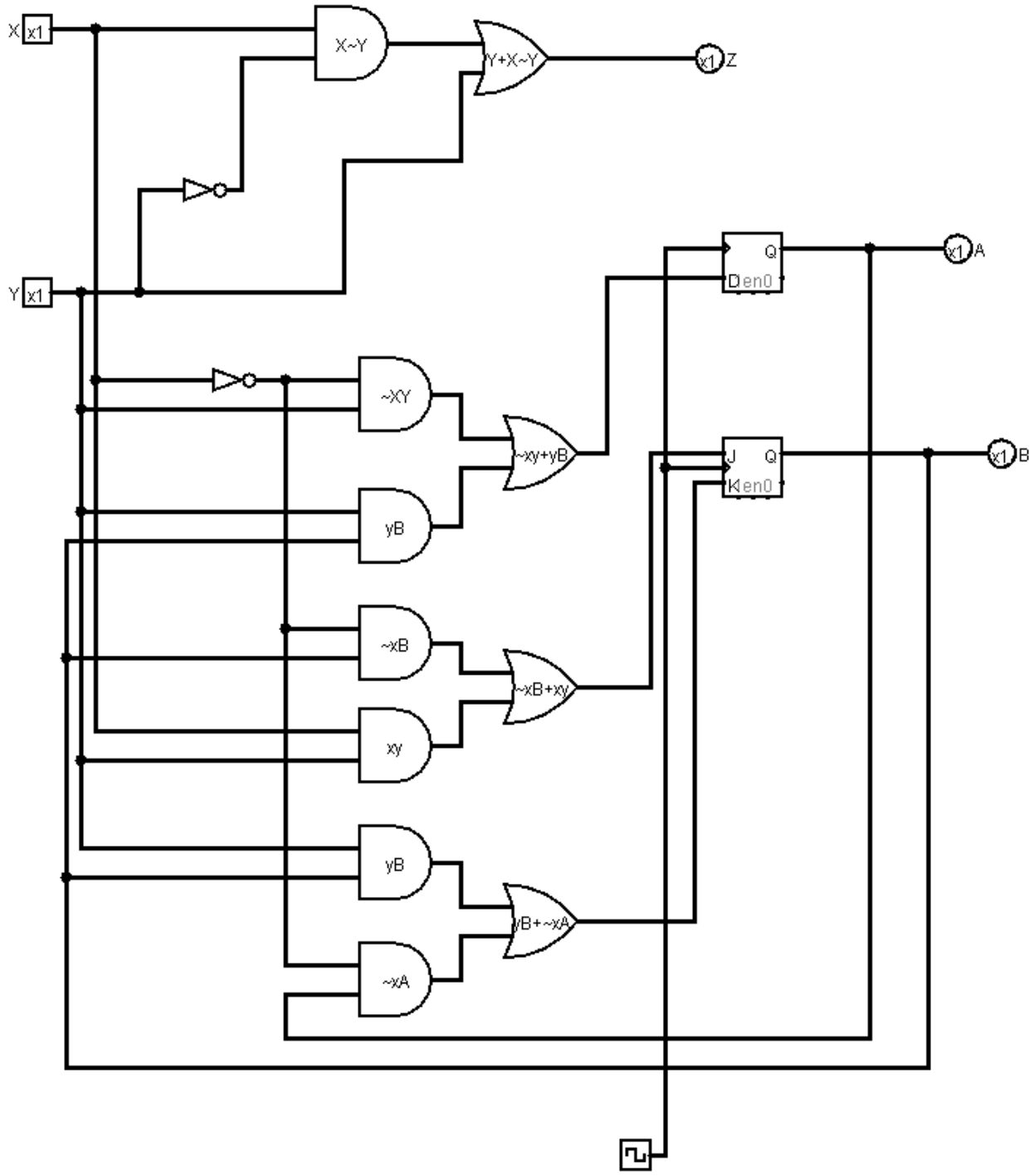
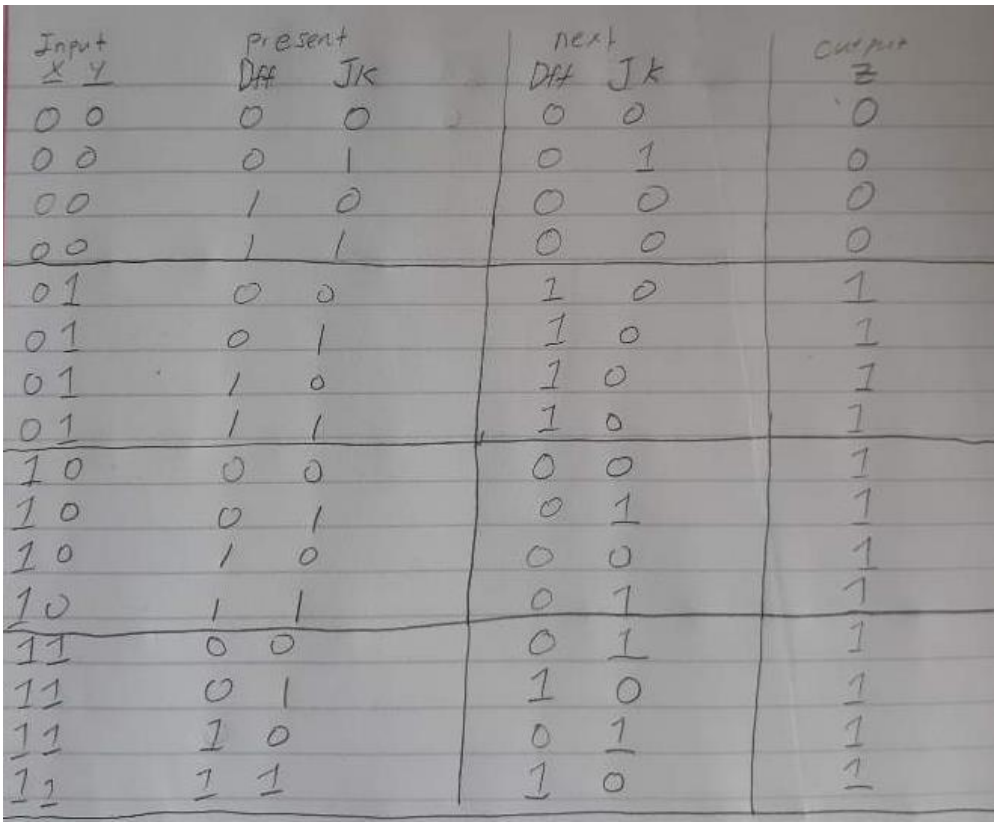


Table it generates in Logisim

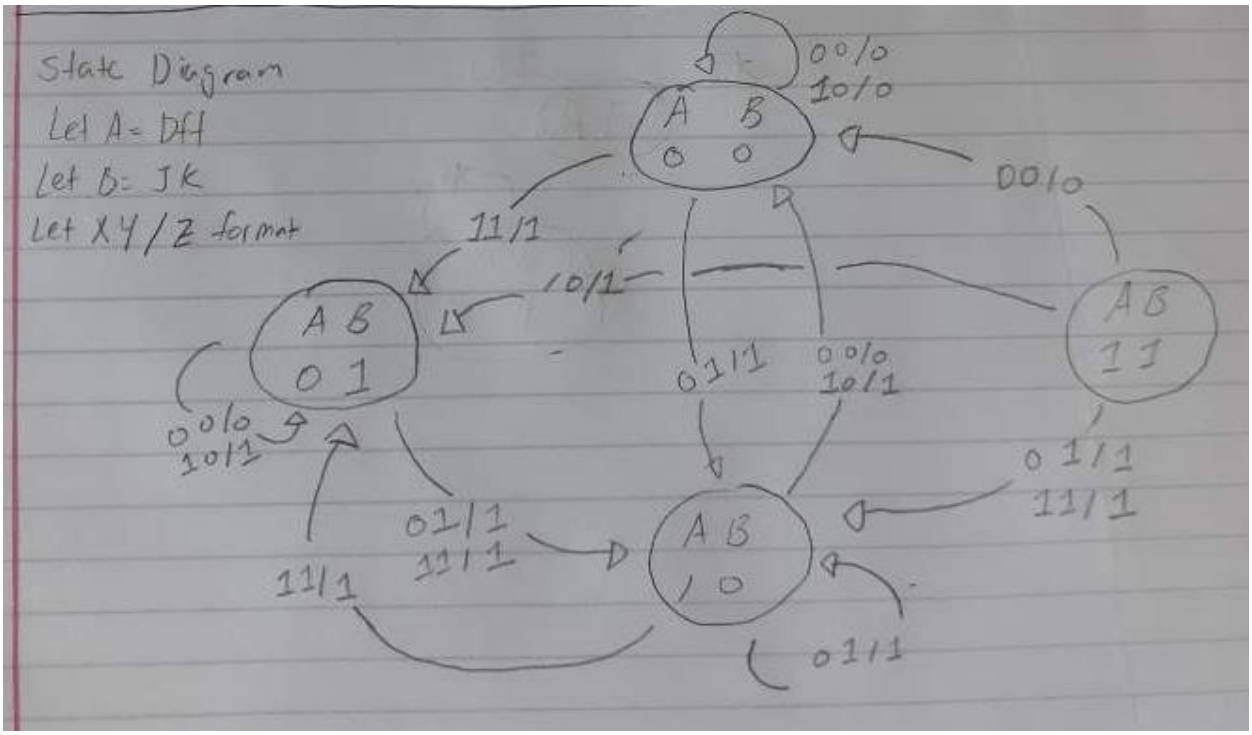
X	Y	Z	A	B
0	0	0	0	0
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

State Table I worked on by Testing



Input X	Input Y	present Dff	present Jk	next Dff	next Jk	output z
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	0	0	1
1	0	0	1	0	1	1
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	0	1	1
1	1	1	1	1	0	1

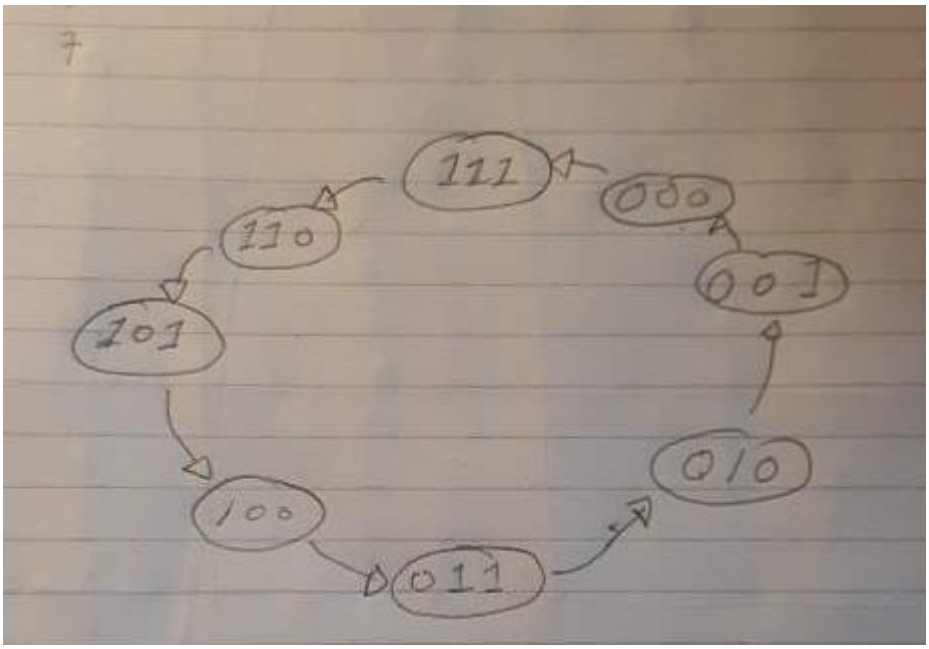
(2) Construct a state diagram of this circuit.



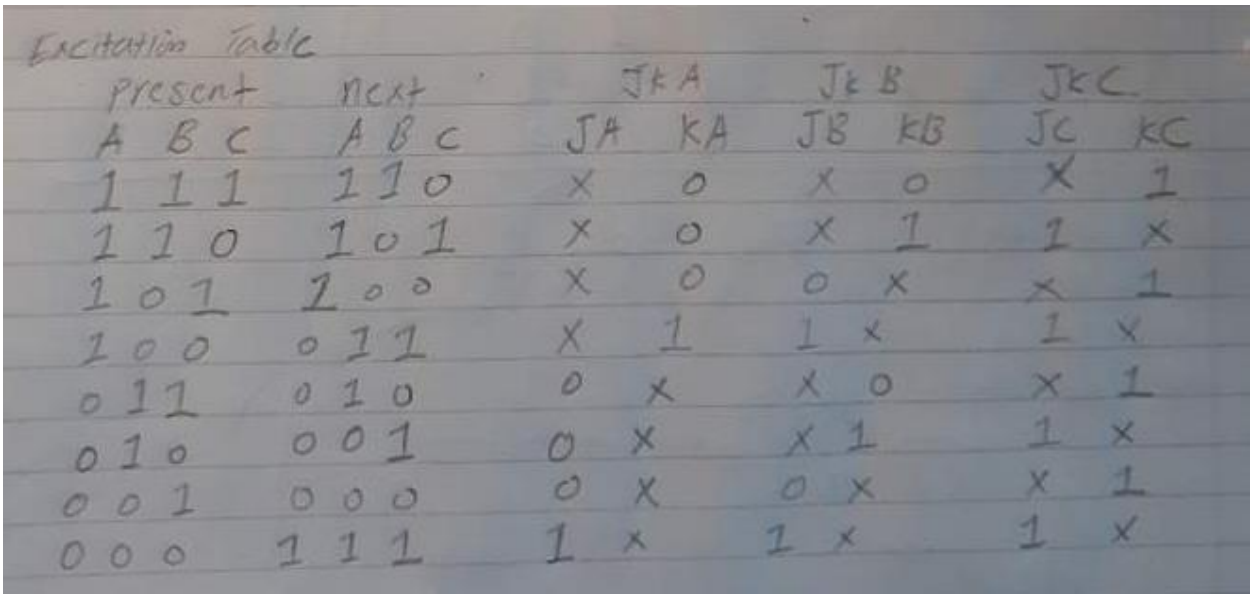
Q5. (5 pts) Design a system

Design a system with the following state changes: This is a sequential circuit with three flip-flops. The state sequence is changed with a clock as in the order of, 111,110,101,100,011,010,001,000,111 and repeat. Use JK flip-flops.

(1) Draw a state diagram



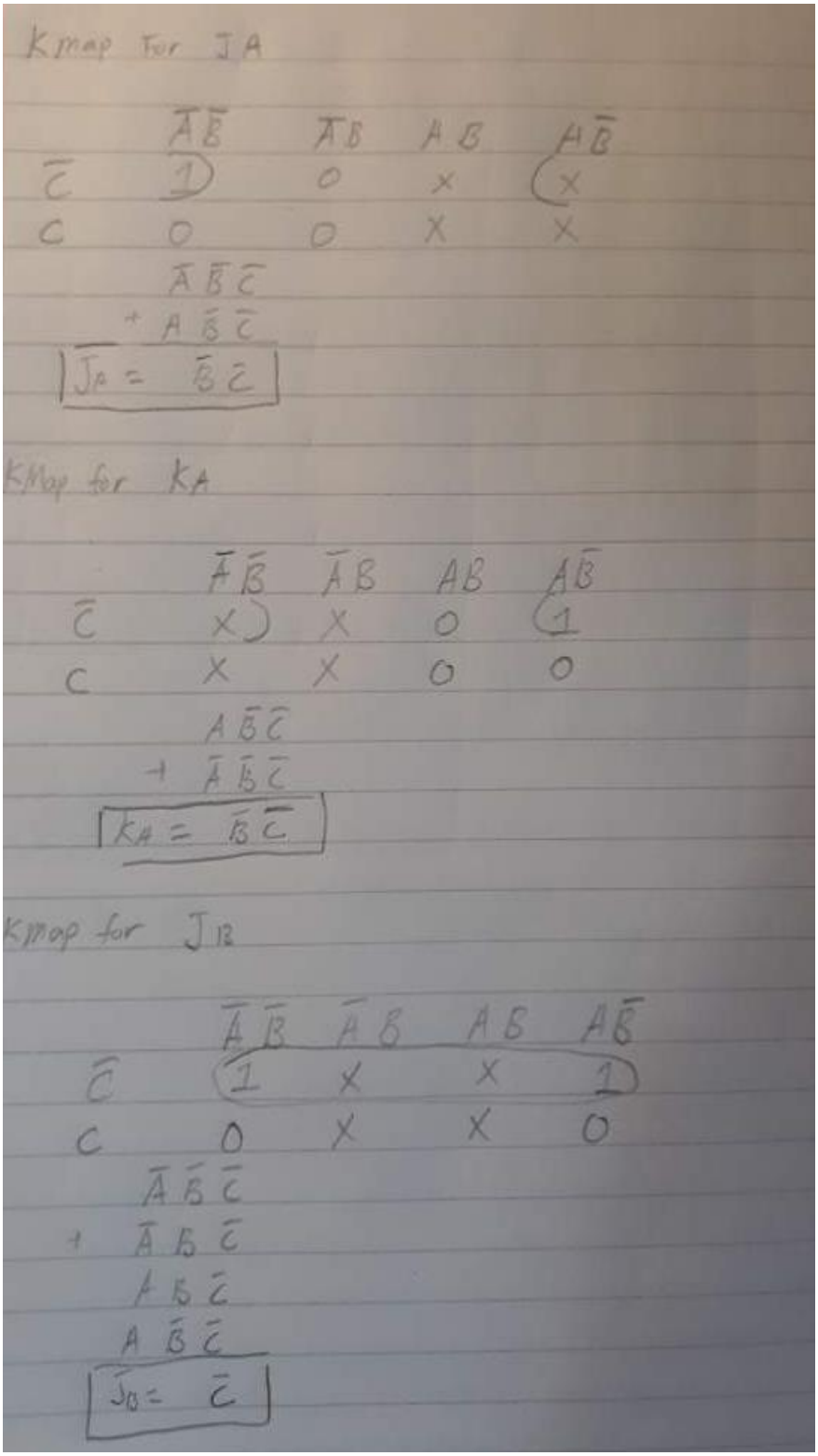
(2) Construct an excitation table

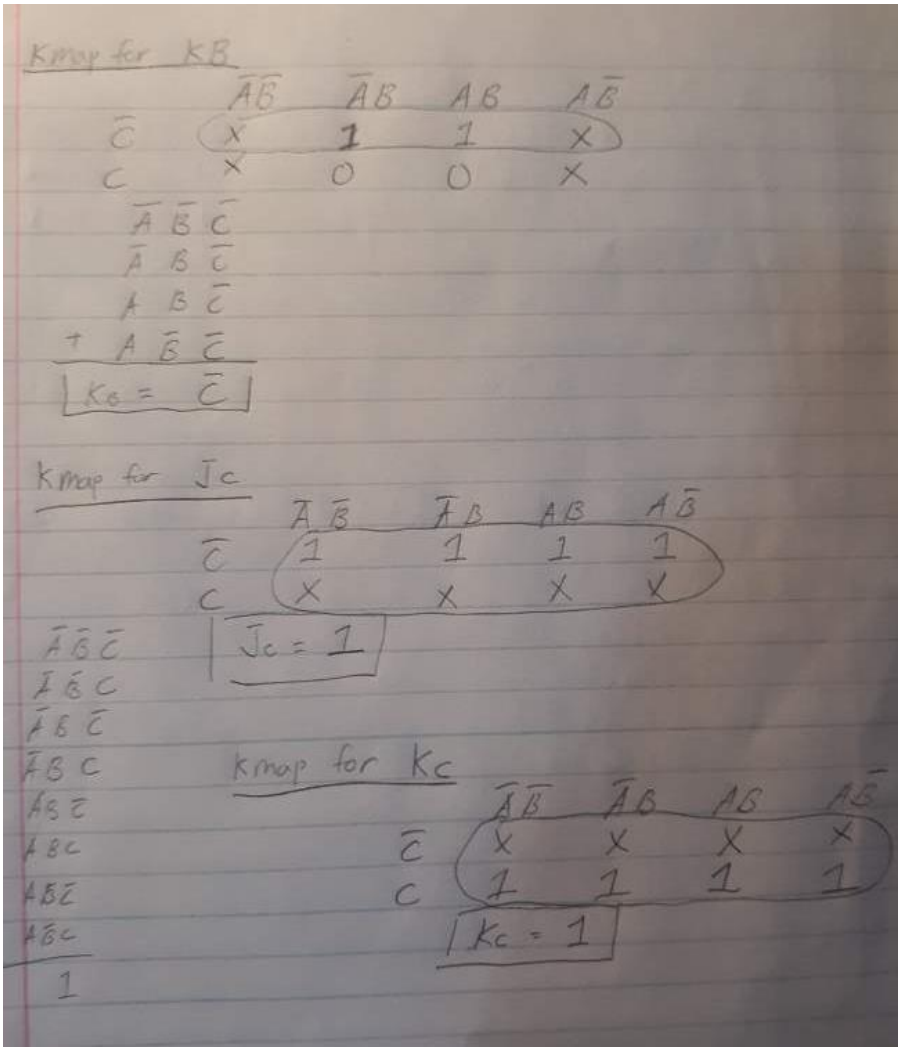


Excitation Table

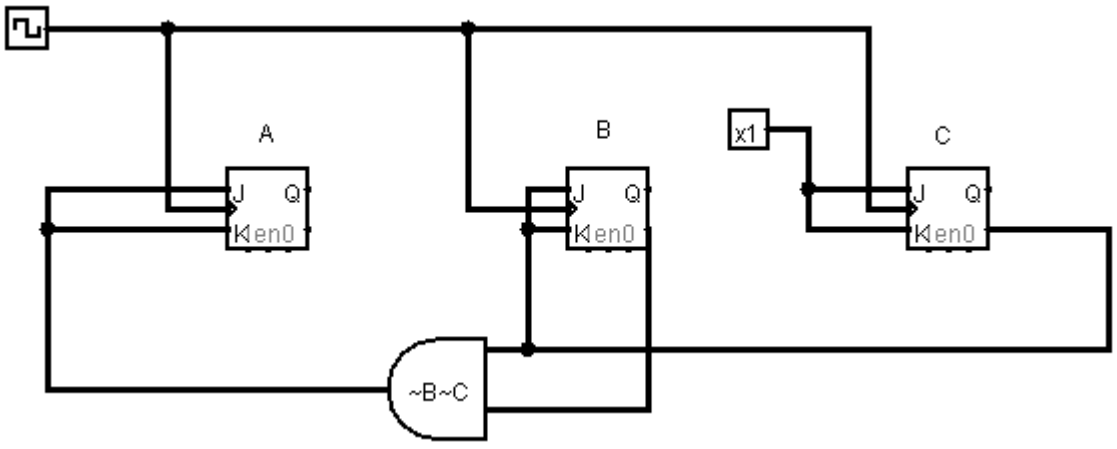
Present			Next			JK A		JK B		JK C	
A	B	C	A	B	C	JA	KA	JB	KB	JC	KC
1	1	1	1	1	0	x	0	x	0	x	1
1	1	0	1	0	1	x	0	x	1	1	x
1	0	1	1	0	0	x	0	0	x	x	1
1	0	0	0	1	1	x	1	1	x	1	x
0	1	1	0	1	0	0	x	x	0	x	1
0	1	0	0	0	1	0	x	x	1	1	x
0	0	1	0	0	0	0	x	0	x	x	1
0	0	0	1	1	1	1	x	1	x	1	x

(3) Draw K-maps and derive Boolean equations using K-maps. Make the equations as simple as possible





(4) Draw the system in Logisim simulator and include the **circuit image and circuit file** .



Circuit file will be provided.

(5) Test the system and include the **generated table** .

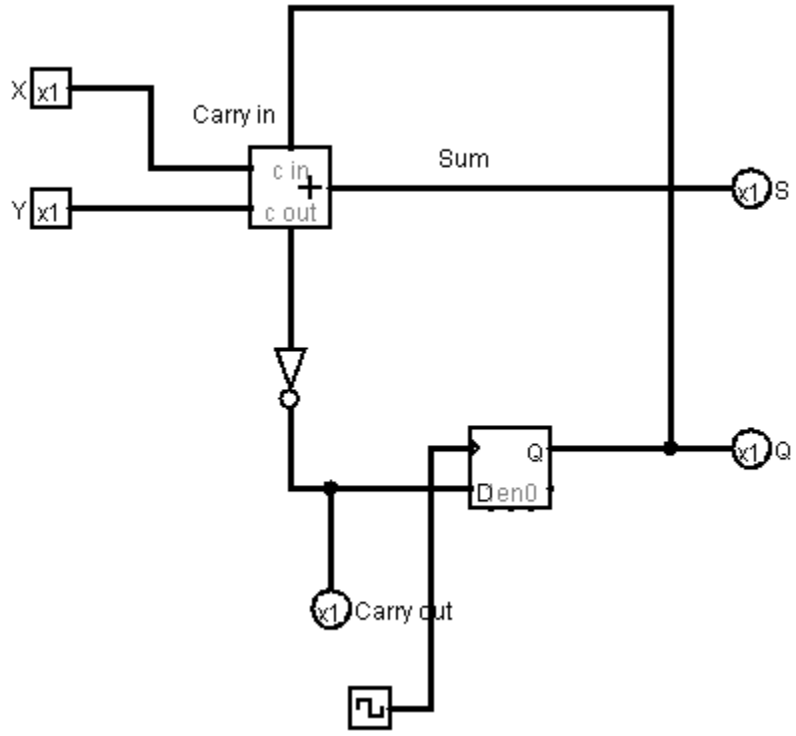
A	B	C
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0
1	1	1

The output of the table matches with FSM, therefore it is correct.

Extra points (5 pts). Sequencial circuit and a full-adder

The following sequential circuit includes a full-adder (described in the previous question). Inputs are X, Y and carry-in, and outputs are the next state of S and Q.

(1) Implement the sequential circuit in Logisim simulator and submit the **image** and **circuit** file.



(2) Complete the following truth table for the following sequential circuit: Note that the Carry out signal is output, not the input. The carry in signal is the same as the Q. You can change the Carry-in bit by clicking the D-FF.

X	Y	Carry-in (or Q before clock)	S (before clock)	Carry-out (before clock)	S (after clock)	Carry-out (after clock)
0	0	0	0	1	1	1
0	0	1	1	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	0
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	1	0	0	0