Bisrat Asefaw

CSS 422 (Hardware)

Prof. Yang

Homework Problem Set #5

Q1. (6 points) 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 the Full-Adder

x	y	z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2) Based on the 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.

S table	xy	~xy	~x~y	x~y
z			1	
~z		1		1

S = ~x~yz + ~xy~z + x~y~z

3) Based on the 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.

C table	xy	~xy	~x~y	x~y
z	1	1		1
~z	1			

C = xy + yz + xz

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)

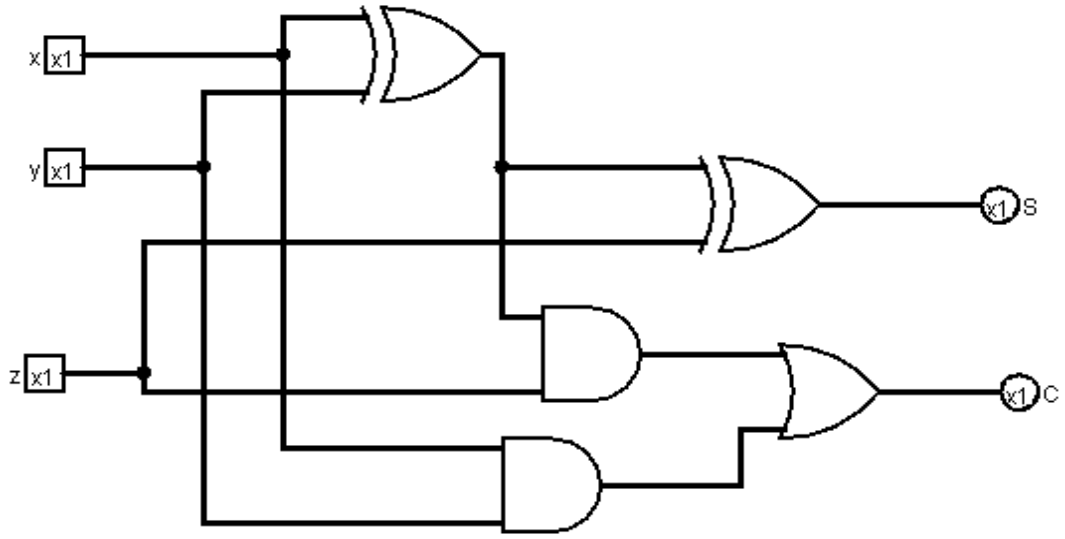
S = x XOR z XOR y
= (~(xXORy)z) + (x ⊕y)~z)
= (~(x~y+~xy)z) + ((x~y+~xy)~z)
= (x~y~z + ~xy~z) + (~(x~y + ~xy)z)
= (~(x~y)*~(~xy))z + ~xy~z + x~y~z
= ((x+~y)*(~x+y))z + ~xy~z + x~y~z
= (x~x+xy + ~x~y+y~y)z + ~xy~z + x~y~z
= (xy+~x~y)z+~xy~z + x~y~z
= xyz + ~x~yz + ~xy~z + x~y~z

5) By algebraic manipulation, show that C can be expressed as the following term C = xy + (x XOR y)z.

xy + (xXORy)z = xy + (x~y+~xy)z

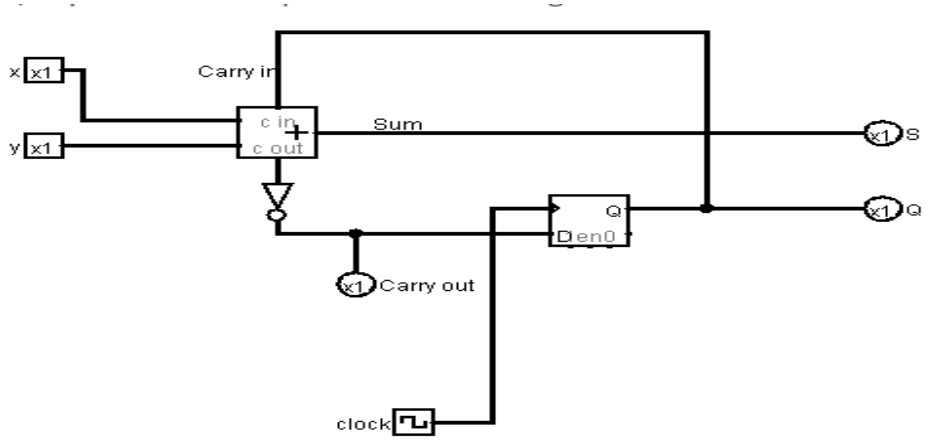
= xy + x~yz+~xyz (Note: xy = xyz + xy~z)
= xyz + xy~z + x~yz+~xyz
= x(yz+y~z+~yz)+~xyz
= x(y + z) + ~xyz (Note: y = yz+ y~z = y; z = yz + ~yz, so yz +y~z + yz +~yz, while yz + yz = yz)
= xy + xz + ~xyz + xyz (Note: Since xy + xz is here xyz is logically inferred as here)
= xy + xz + yz(x+~x)
= xy + xz + yz

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



Q2. (6 points) 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 circuit file**.



2) Complete the following truth table for the following sequential circuit: Note that the Carry out signal is an output, not an 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

Q3. (4 points) A sequential circuit has one D flip-flop and one JK flip-flop, two inputs x and y, and one output z. A is the output of D flip-flop, and B is the output of JK-flip-flop; A and B together form the "output state" of the circuit. The flip-flop *input* equations and the circuit output are as follows. Here DA is the D input of the D-flip flop of A, and JB, KB is the J and K input of the JK-flip flop of B.

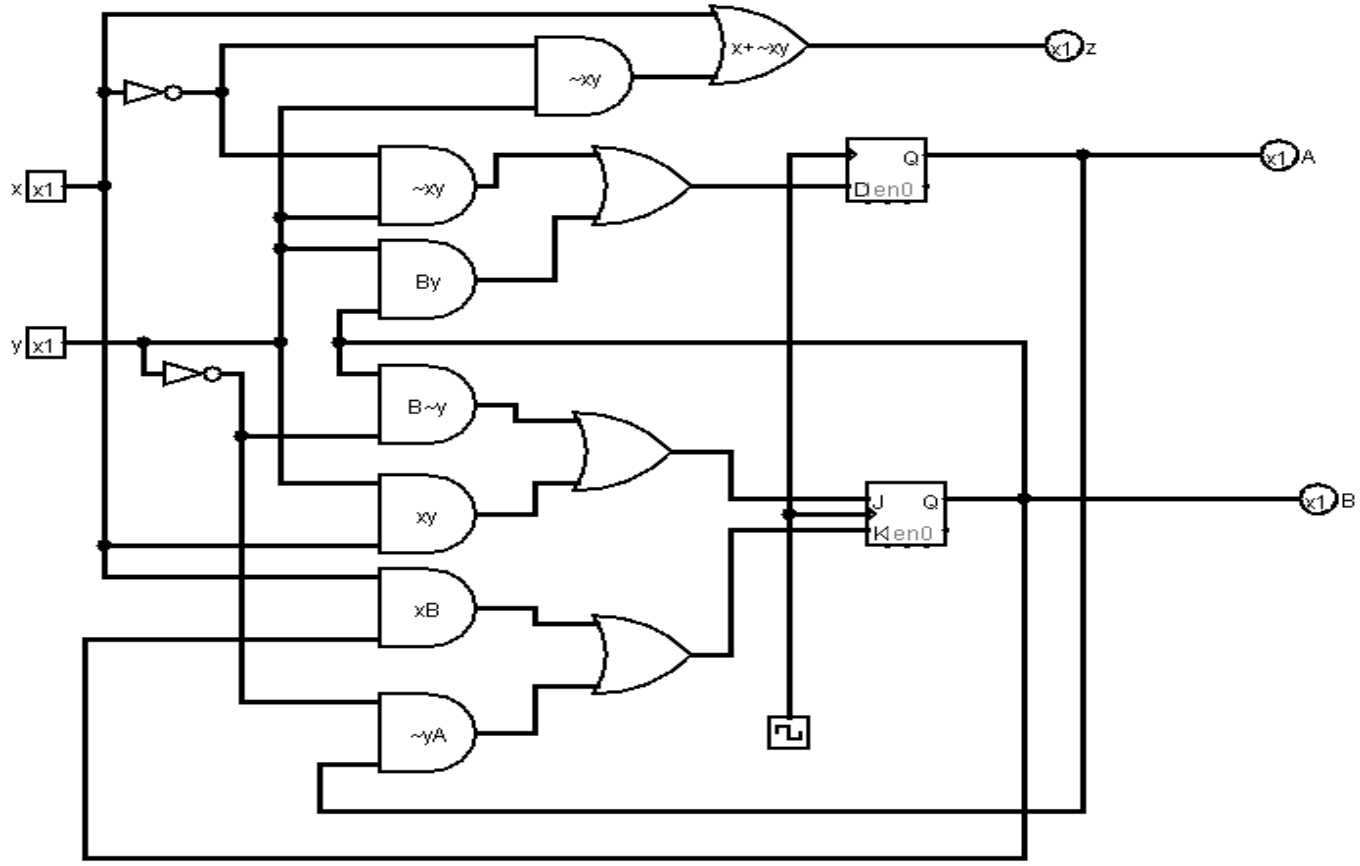
DA = ~xy + yB

JB = ~yB + xy

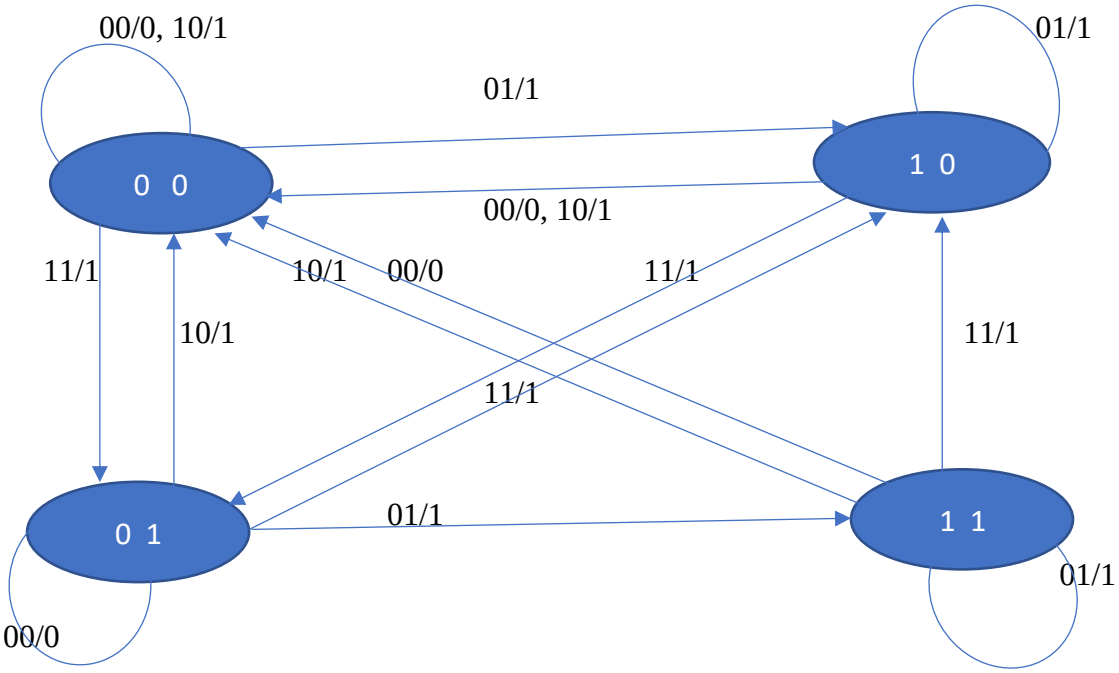
KB = xB + ~yA

z = x+~xy

1) Draw the logic diagram of the circuit and test it with Logisim.

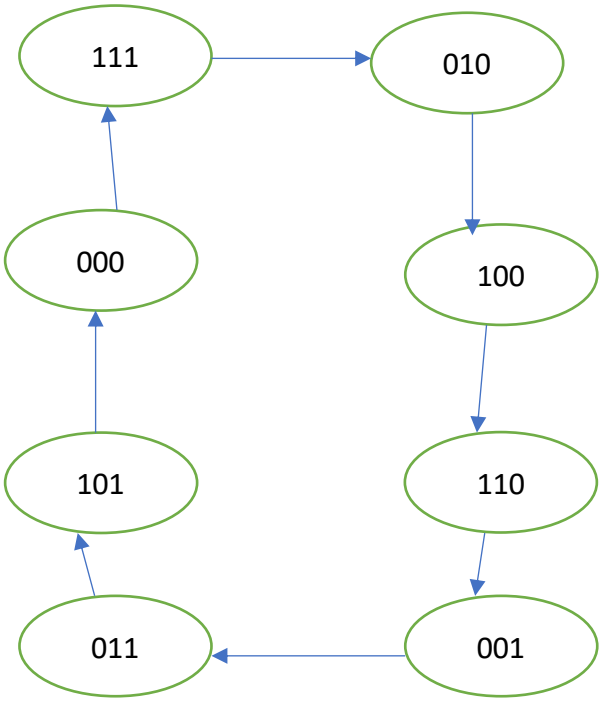


2) Construct a state diagram of this circuit.



Q4. (10 points) 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, 010, 100,110, 001, 011, 101, 000, 111 and repeat. Use JK flip-flops.

1) Draw a state diagram.



2) Construct an excitation table.

A_t	B_t	C_t	$A_{(t+1)}$	$B_{(t+1)}$	$C_{(t+1)}$	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	1	1	1	1	x	1	x	1	x
0	0	1	0	1	1	0	x	1	x	x	0
0	1	0	1	0	0	1	x	x	1	0	x
0	1	1	1	0	1	1	x	x	1	x	0
1	0	0	1	1	0	x	0	1	x	0	x
1	0	1	0	0	0	x	1	0	x	x	1
1	1	0	0	0	1	x	1	x	1	1	x
1	1	1	0	1	0	x	1	x	0	x	1

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

J_A	AB	~AB	~A~B	A~B
C	x	1		x
~C	x	1	1	x

$J_A = B + \sim C$

K_A	AB	~AB	~A~B	A~B
C	1	x	x	1
~C	1	x	x	

$K_A = B + C$

J_B	AB	~AB	~A~B	A~B
C	x	x	1	
~C	x	x	1	1

$J_B = \sim A + \sim C$

K_B	AB	~AB	~A~B	A~B
C		1	x	x
~C	1	1	x	x

$K_B = \sim A + \sim C$

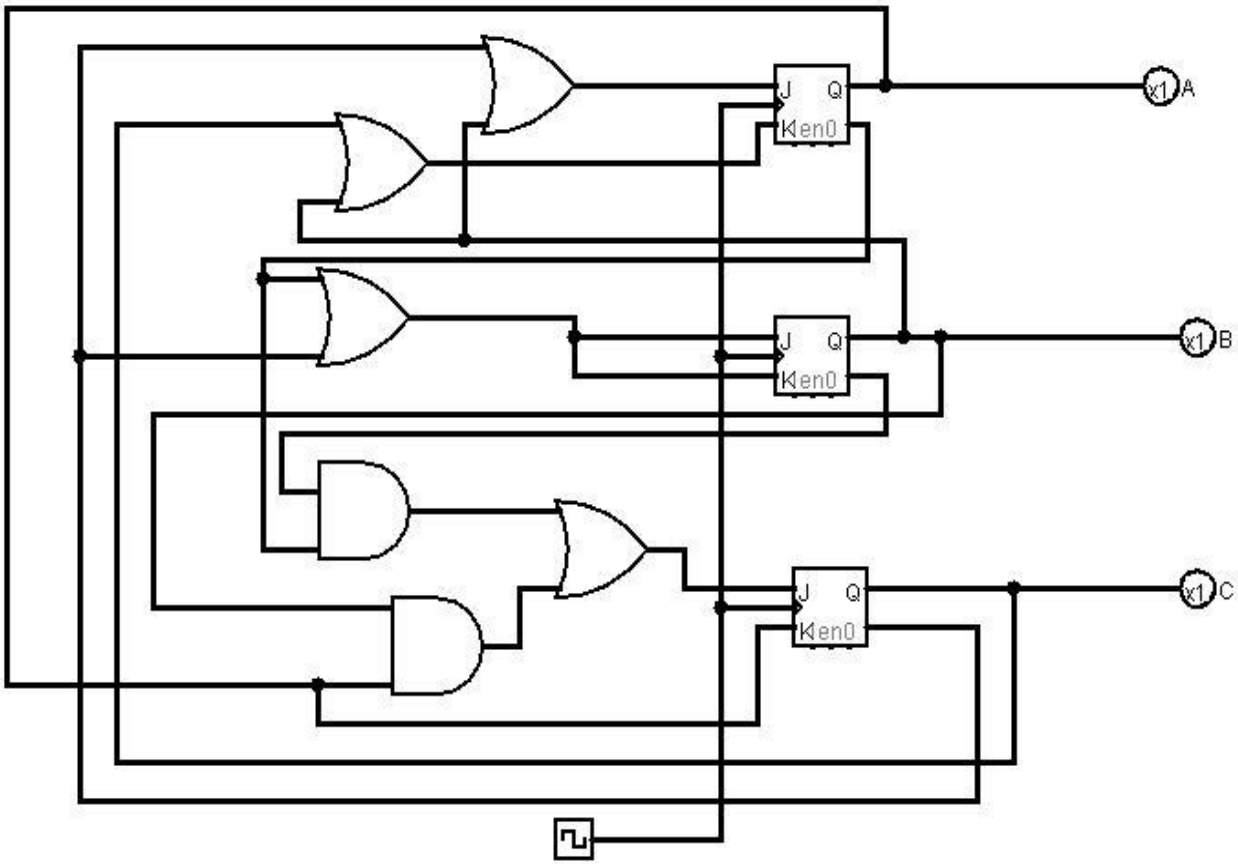
J_C	AB	~AB	~A~B	A~B
C	x	x	x	x
~C	1		1	

$J_C = AB + \sim A \sim B$

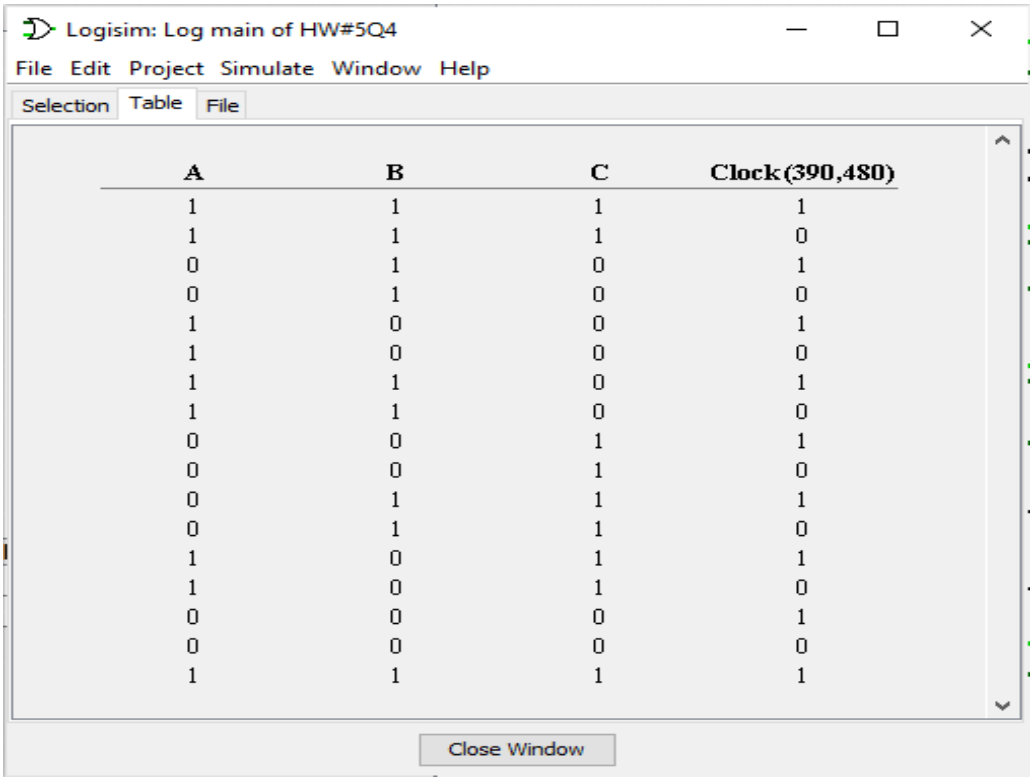
K_C	AB	~AB	~A~B	A~B
C	1			1
~C	x	x	x	x

$K_C = A$

4) Draw the system in Logisim simulator, attach the circuit image and submit the circuit file.



5) Test the system and **attach the generated table**.



A	B	C	Clock(390,480)
1	1	1	1
1	1	1	0
0	1	0	1
0	1	0	0
1	0	0	1
1	0	0	0
1	1	0	1
1	1	0	0
0	0	1	1
0	0	1	0
0	1	1	1
0	1	1	0
1	0	1	1
1	0	1	0
0	0	0	1
0	0	0	0
1	1	1	1

Q5. (2 points) The following circuit is a simple implementation for 4X3 memory chip. In this configuration, your memory chip has four addressable space and each data in the address is 3-bit long. In order to write/read a data into/from a specified address, you have to give a right signal to the address lines, S1 and S0.

1) **(1 point)** Suppose you want to write a data 1 0 1 to the word 3 (address 3). How you will set the values in each case?

RESET	S1	S0	Bit2	Bit1	Bit0	~WE
0	**1**	**1**	**1**	**0**	**1**	**1**

2) (**1 point**) Suppose you want to read a data from the address 1. Give the correct values in each case. If some bits do not affect, then mark as X (don't care).

RESET	S1	S0	Bit2	Bit1	Bit0	~WE
0	**0**	**1**	**X**	**X**	**X**	**0**