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 |

1. 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S | ~X~Y | ~XY | XY | X~Y |
| Z | 1 |  | 1 |  |
| ~Z |  | 1 |  | 1 |

**S = XY~Z + X~YZ + ~XY~Z + ~X~YZ**

1. 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S | ~X~Y | ~XY | XY | X~Y |
| Z |  | 1 | 1 | 1 |
| ~Z |  |  | 1 |  |

**C = XY + ZX + YZ**

1. 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.

S = X ⊕ Z ⊕ Y

= X(Z ⊕ Y) + ~X(Z ⊕ Y)

= X(Y~Z+~YZ) + ~X(Y~Z + ~YZ)

= XY~Z + X~YZ + ~XY~Z + ~X~YZ

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

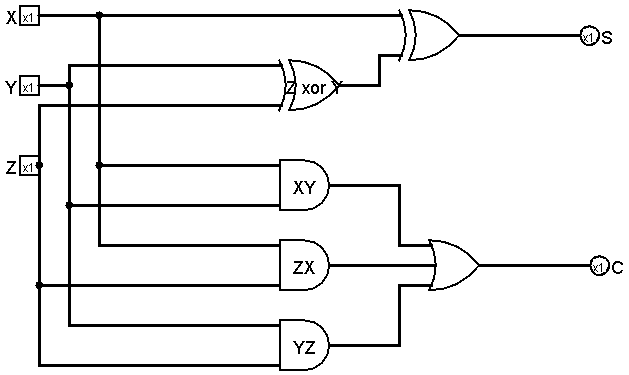
C = xy + (x XOR y)z

= XY + (X~Z + ~XZ)Z

= XY(X + ~X)(Y + ~Y)Z

= XY + (XZ + ~XZ) + (YZ + ~YZ)

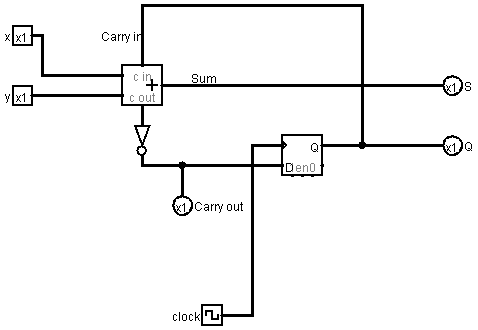
= XY + ZX + YZ

1. 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 the 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 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

# Q3. (4 points)

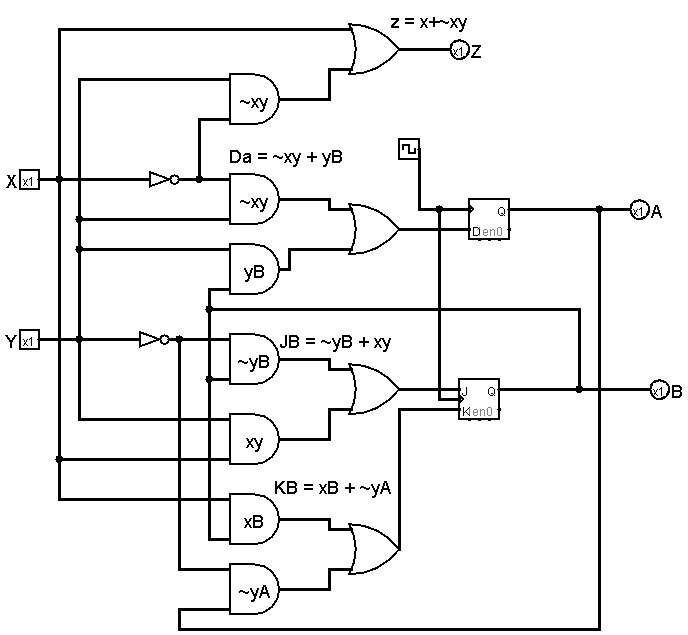
A sequential circuit has one D flip-flop and one JK flip-flop, two inputs x and y, and one output z. A and B are the outputs of each D flip-flop, and JK-flip-flop, respectively. The flip-flop *input* equations and the circuit output are as follows. Here DA is the D input of the D-flip flop of A, and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

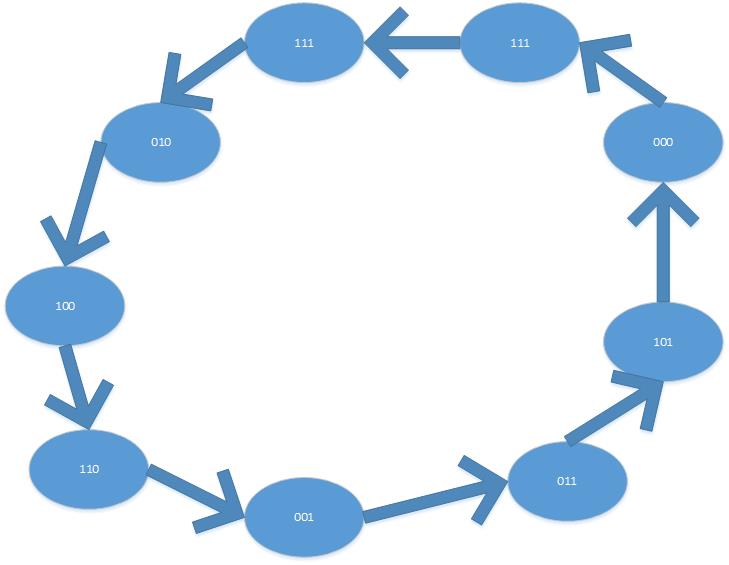
1. Draw the logic diagram of the circuit and test it with Logisim. **Attach the circuit image and attach the generated table; submit the circuit file.**

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

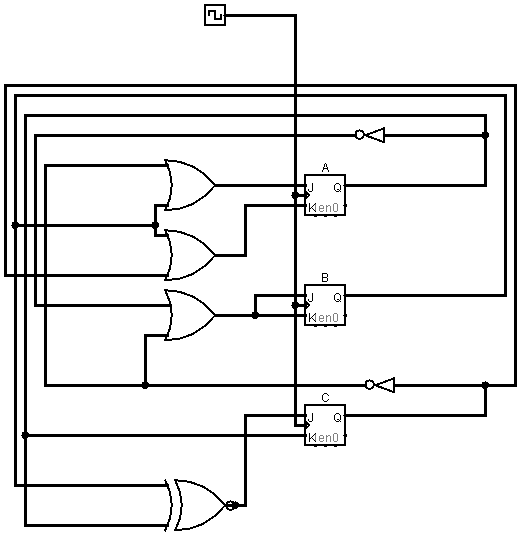


1. Construct an excitation table

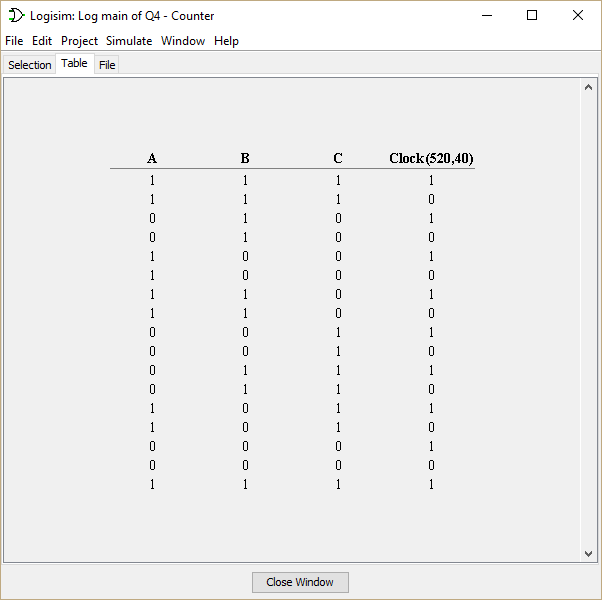
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qt | | | Q(t+1) | | | A | | B | | C | |
| A | B | C | A | B | C | JA | KA | JB | KB | JC | KC |
| 1 | 1 | 1 | 0 | 1 | 0 | X | 1 | X | 0 | X | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | X | X | 1 | 0 | X |
| 1 | 0 | 0 | 1 | 1 | 0 | X | 0 | 1 | X | 0 | X |
| 1 | 1 | 0 | 0 | 0 | 1 | X | 1 | X | 1 | 1 | X |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | X | 1 | X | X | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | X | X | 1 | X | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | X | 1 | 0 | X | X | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | X | 1 | X | 1 | X |
| 1 | 1 | 1 | 1 | 1 | 1 | X | 0 | X | 0 | X | 0 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| JA | ~A~B | ~AB | AB | A~B | KA | ~A~B | ~AB | AB | A~B |
| C |  | 1 | X | X | C | X | X | 1 | 1 |
| ~C | 1 | 1 | X | X | ~C | X | X | 1 |  |
| JA = B + ~C | | | | | KA = B + C | | | | |
| JB | ~A~B | ~AB | AB | A~B | KB | ~A~B | ~AB | AB | A~B |
| C | 1 | X | X |  | C | X | 1 |  | X |
| ~C | 1 | X | X | 1 | ~C | X | 1 | 1 | X |
| JB = ~A+~C | | | | | ~A + ~C | | | | |
| JC | ~A~B | ~AB | AB | A~B | KC | ~A~B | ~AB | AB | A~B |
| C | X | X | X | X | C |  |  | 1 | 1 |
| ~C | 1 |  | 1 |  | ~C |  | X | X | X |
| JC = A ⊕ B | | | | | KC = A | | | | |

1. Draw K-maps and derive Boolean equations using K-maps. Make the equations as simple as possible. **SHOWED GROUPINGS IN HANDWRITTEN NOTES AT BOTTOM. HARD TO DO IN WORD**
2. Draw the system in Logisim simulator, attach the circuit image and submit the circuit file here.



1. Test the system and attach the generated table here.



# 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.

Download the circuit file [hereView in a new window](https://canvas.uw.edu/courses/1053612/files/35777441/download). And open the circuit file in logisim to test the circuit in Logisim. And answer the following questions:

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** |

