Homework 3
Total Points: 80

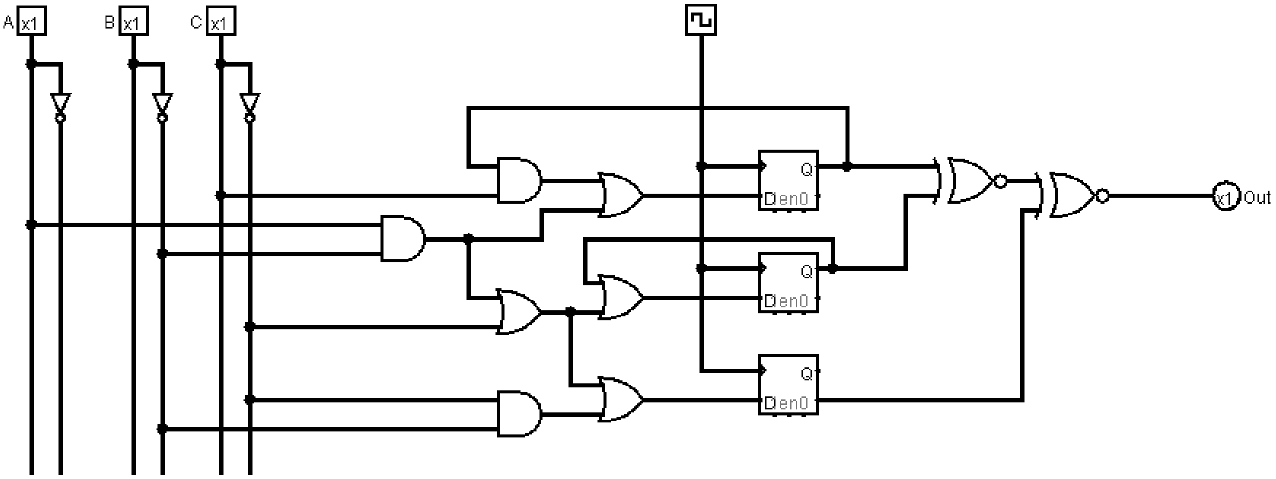
Written

1. (5 points) Create a JK Flip-Flop using only a T Flip-Flop and basic logic gates.
2. (5 points) Create an SR Flip-Flop using only a D Flip-Flop and basic logic gates.
3. (5 points) Implement the **simplest** circuit possible using **JK Flip-Flops** based on the following transition table.

Current State/Encoding	Input	Next State	Output
00	0	11	1
00	1	10	0
01	D	00	0
10	0	10	0
10	1	11	0
11	0	01	1
11	1	00	1

4. (5 points) Derive the **minimal** state table for a single input, single output **Moore** model FSM that outputs 1 whenever it detects either 110 or 101 in the input sequence. Overlapping sequences should be detected. For example if the input is 1101 then the output would be 00011 (Don't forget that the output of a Moore is delayed 1 clock cycle behind the input.
5. (5 points) Repeat number 4 but for a **Mealy** model FSM.
6. (5 points) Given the propagation delays contained in the table below and that the setup time for a D Flip-Flop is 3ns determine the length of the worst case path **and** the maximum clock frequency for the following circuit.

Gate	Propagation Delay
AND	5ns
OR	3ns
NOT	2ns
XNOR	6ns
D Flip-Flop	4ns


A
x1
B
x1
C
x1
Q
D
en0
Q
D
en0
Q
D
en0
x1
Out

Logisim

You may use all components under wiring and all components under gates **except** for the parity checkers. From the memory tab you may use all Flip-Flops and the register. You may use no other components besides these unless specified in the problem.

1. (15 points) Implement a 3 bit synchronous up/down counter that stops counting when the minimum/maximum count is reached. For example if the count is at 111 and you are told to count up you should stay at 111. Count_Up and Count_Down will never both be 1 at the same time. If both Count_Up and Count_Down are 0 the count should not change.
    1. Inputs: Count_Up, Count_Down.
        1. If Count_Up is high the value increases by 1 unless the count is at 111.
        2. If Count_Down is high the value decreases by unless the count is 000.
    2. Output: Count
2. (15 points)Implement a **Moore** model circuit that calculates the even parity over a **group** of 4 bits. The circuit should output a 1 if after the last bit in the group is received, the number of 1's in the group of 4 bits is even and 0 at all other times.
    1. Example inputs/output
        1. Input: 0111 1000 1010
        2. Output: 0000 0000 0000 1
3. (Credit for this problem goes to Sean Davis 20 points) Implement a **Mealy** model circuit to control a coin operated vending machine. This machine only accepts quarters, dimes, and nickels. Coins are inserted until a total of 30 cents or more is deposited. X1 = quarter, x2 = dime, X3 = nickel. Only 1 coin is deposited at a time. The output signal, z1 indicates whether merchandise should be provided or not; z1 = 0 indicates no merchandise and z1 = 1 means merchandise should be given. Coincident with the last coin input, the change outputs are to be set. Assume that the machine can give a dime, z2 = 1, and/or a nickel, z3 = 1. If the customer does something silly like entering a quarter followed by a quarter, correct change does not have to be provided but the maximum amount of change should be given. Note that only 1 input can be high at a time.