In this lab you will work on:

1. *Finite State Machines*
2. *Moore Machines*

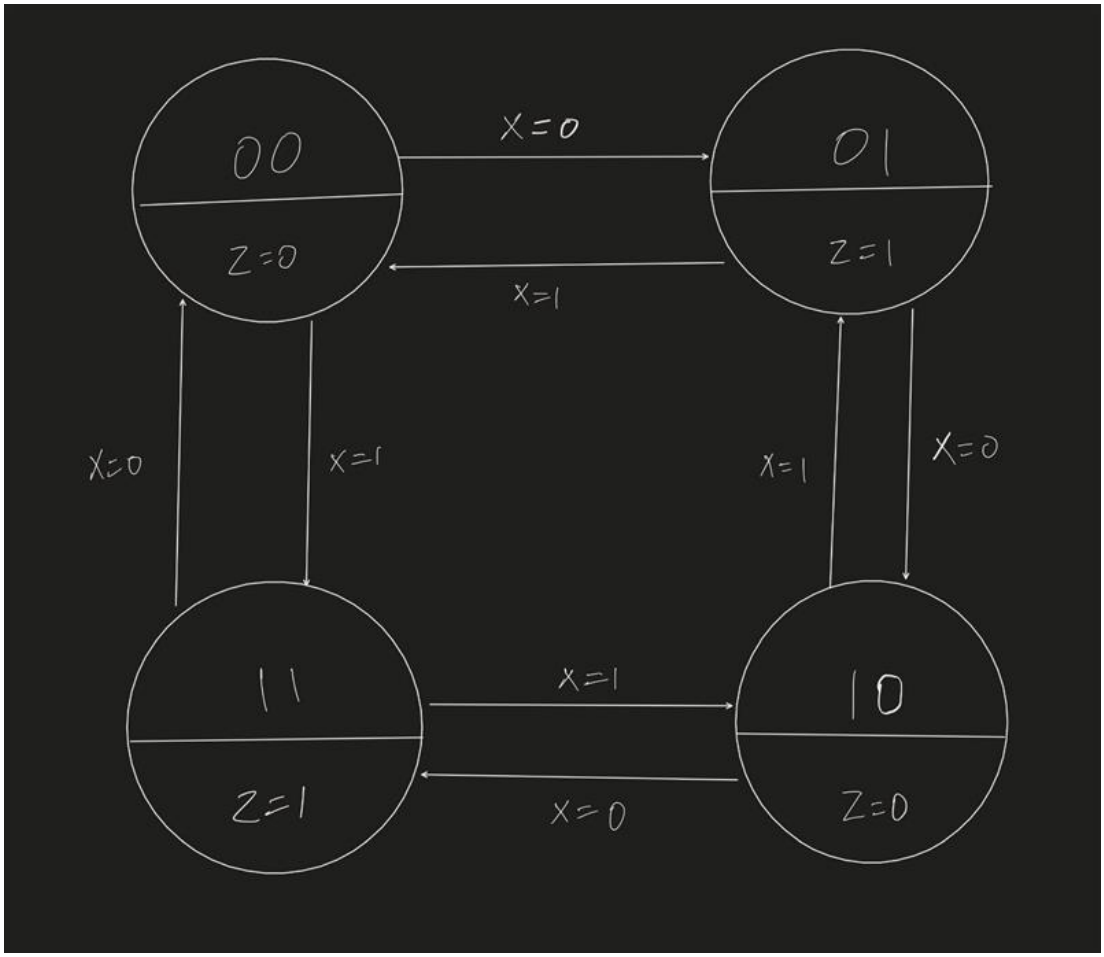
Problem Description:

EDA Playground starter files: https://edaplayground.com/x/7Jy5

In this week's lab you are tasked with creating a counter using a Finite State Machine (FSM) more specifically a Moore Machine. As you have learned in class a Moore machine means that the output is only dependent on the current state. Please watch the video in this week's module on Canvas that goes over an FSM example and how it should be implemented in Verilog (it will make this lab much easier if you watch the video!).

A customer wants you to design the following system to create a Counter:

Create a 2-bit counter that can count in increasing order or decreasing order depending on the input. You are given the State Diagram below which your program should match. Since the client request a 2-bit counter this means you should have 4 states 0,1,2,3. When the input is 0, your counter should count in increasing order 0->1->2->3->0... When the input is 1, it should count in decreasing order 0->3->2->1->0... The client would like the machine to output 1 if the counter is in an odd state or 0 if it is in an even state.



Project Requirements:

Design.sv

- You will have a module Design.sv that will be your top level module and act as your Moore Machine
- Design.sv will have three inputs clock, x, and reset and one output z all of which are 1-bit
- This module should follow the design of the circuit shown above, it should contain two always blocks: one that determines the next state and another that determines the output
- when x=0 the counter should count upward (increasing) and when x=1 the counter should count downward (decreasing)
- z should be set to 0 if the current state is even and 1 if it is odd
- when reset is 0 the current state should be set to the 0 state

Input output format:

1. Once again, the testbench is completed for you, you do not need to change anything on it.
2. If you have completed the project successfully the following output

```
DOWN COUNTING

STATE = 0 INPUT = 1, OUTPUT = 0
STATE = 3 INPUT = 1, OUTPUT = 1
STATE = 2 INPUT = 1, OUTPUT = 0
STATE = 1 INPUT = 1, OUTPUT = 1

UP COUNTING

STATE = 2 INPUT = 0, OUTPUT = 0
STATE = 3 INPUT = 0, OUTPUT = 1
STATE = 0 INPUT = 0, OUTPUT = 0
STATE = 1 INPUT = 0, OUTPUT = 1
```