In this lab you will work on:

1. *D Flip Flop*
2. *Linear Feedback Shift Register*

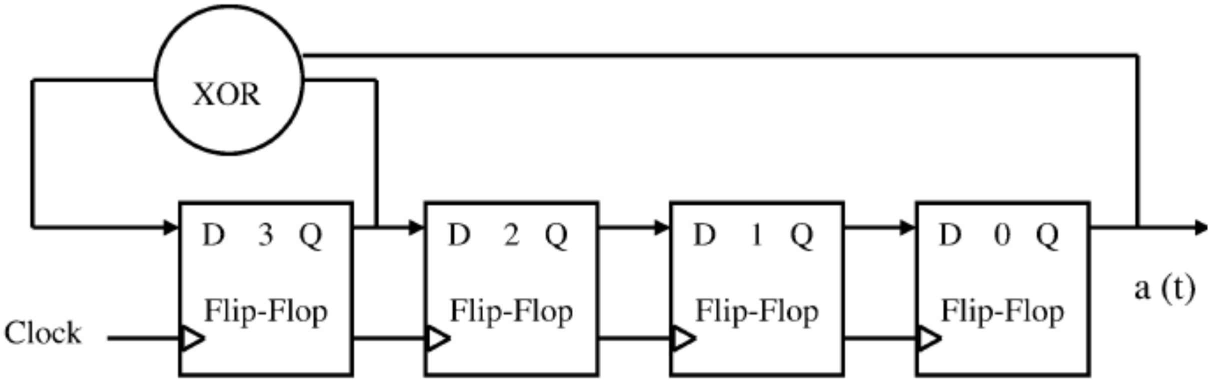
Problem Description:

EDA Playground starter files: https://edaplayground.com/x/Z2L4

In this week's lab you are tasked with creating a pseudo random number generator using a Linear Feedback Shift Register (LFSR). A LFSR is similar to a regular shift register that uses cascading flip flops all on the same clock. The output of each flip flop is fed as the input to the next one. The way it differs is that there is a linear function of some outputs (called taps) that is fed back into the first input of the FF. In our case, it is a xor function of the first and the last FF. It generates pseudo random numbers because the numbers are not truly random in the sense that depending on the initial seed value (the values the FF's the initialized to), it will generate the same repeating pattern of random numbers. As you will notice in your output waveform, the numbers will be random, but after a point, that sequence will repeat.

A customer wants you to design the following system to create a Pseudo Random Number Generator:

Create a 4-bit pseudo random number generator using four cascading positive edge D Flip Flops. In order to randomize the output, we request you use a xor gate to feed the input of your first D Flip Flop. Following the diagram below the input D_3 will be equal to Q_3 xor Q_0.



Project Requirements:

DFF.sv

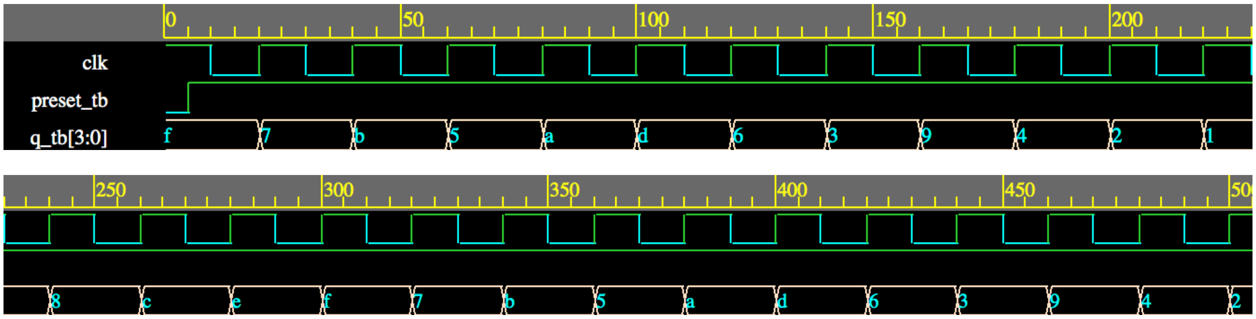
- You will have a module DFF.sv that will function as a D Flip Flop
- DFF.sv will have three inputs clock, d, and preset and one output q
- The preset is asynchronous (think about if you have to include it in the always block sensitivity list or not) and active low - if preset is 0 then your output q should be set to 1
- on the positive clock edge your output q should be set to equal d just like a D-FF would function
- your q and d should both be one bit

Design.sv

- You will have a module Design.sv that will be your top level module and act as your shift register
- Design.sv will have two inputs clock and preset and one output q which should be 4-bits
- This module should follow the design of the circuit shown above, it should call four D-FF's with the correct inputs fed into it
- q[3] should be assigned to the output of D-FF$_3$, q[2] should be assigned to the output of D-FF$_2$, q[1] should be assigned to the output of D-FF$_1$, and q[0] should be assigned to the output of D-FF$_0$

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 you should see a wave form similar to the following


0
50
100
150
200
clk
preset_tb
q_tb[3:0]
f
7
b
5
a
d
6
3
9
4
2
1
250
300
350
400
450
50
8
c
e
f
7
b
5
a
d
6
3
9
4
2