Circuit Simulation Project

https://esim.fossee.in/circuit-simulation-project

Name of the participant :

SAI AUCHITYA BUSSA

Title of the circuit :

Linear-feedback shift register (LFSR)

Theory/Description :

LFSR stands for linear feedback shift register, it is comprised of a series of D-flip flops, depending on the size of the LFSR. Some of the states and especially the last one is feed back to the system by going through logical XOR.

The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits that appears random and has a very long cycle.

The repeating sequence of states of an LFSR allows it to be used as a clock divider or as a counter when a non-binary sequence is acceptable, as is often the case where computer index or framing locations need to be machine-readable. LFSR counters have simpler feedback logic than natural binary counters or Gray-code counters, and therefore can operate at higher clock rates. However, it is necessary to ensure that the LFSR never enters an all-zeros state, for example by presetting it at start-up to any other state in the sequence.

So a linear feed-back shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. This is a rotating register, in which one of the Flip-Flops has

a XOR as its input, an XOR among two or more outputs of the remaining Flip-Flops. The outputs connected to the XOR Gate are called TAP.

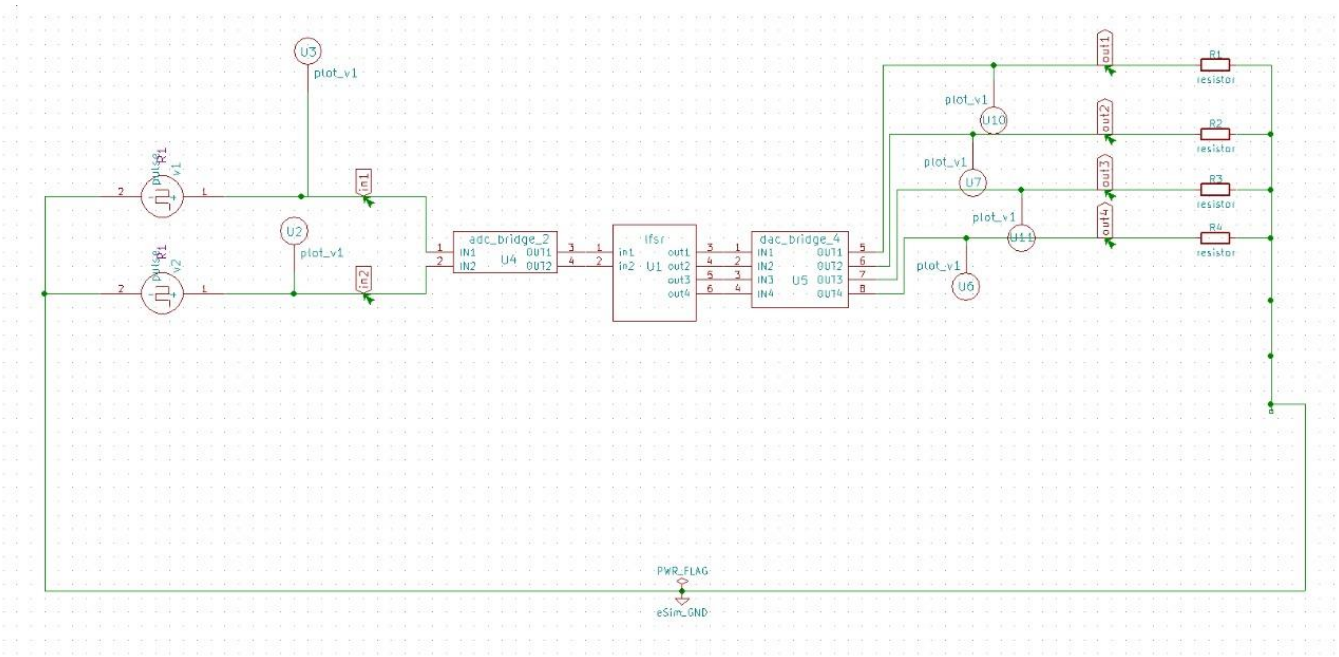
Lets build an 8-bit lfsr in vhdl. I will design the lfsr using FSM(finite state machine).The top level entity is comprised of 2 input and 1 output ports. Input ports are clock and reset. Output port is an 4-bit number.

VHDL CODE:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity lfsr is
    Port ( clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           outp : out STD_LOGIC_VECTOR (3 downto 0));
end lfsr;

architecture Behavioral of lfsr is
signal feedback : std_logic;
signal out_reg : std_logic_vector(3 downto 0):="0000";
begin
feedback <= not (out_reg(3) xor out_reg(2));
process (clk,rst)
begin
    if (rst='1') then
        out_reg <= "0000";
    elsif (rising_edge(clk)) then
        out_reg <= out_reg(2 downto 0) & feedback;
    end if;
end process;
```

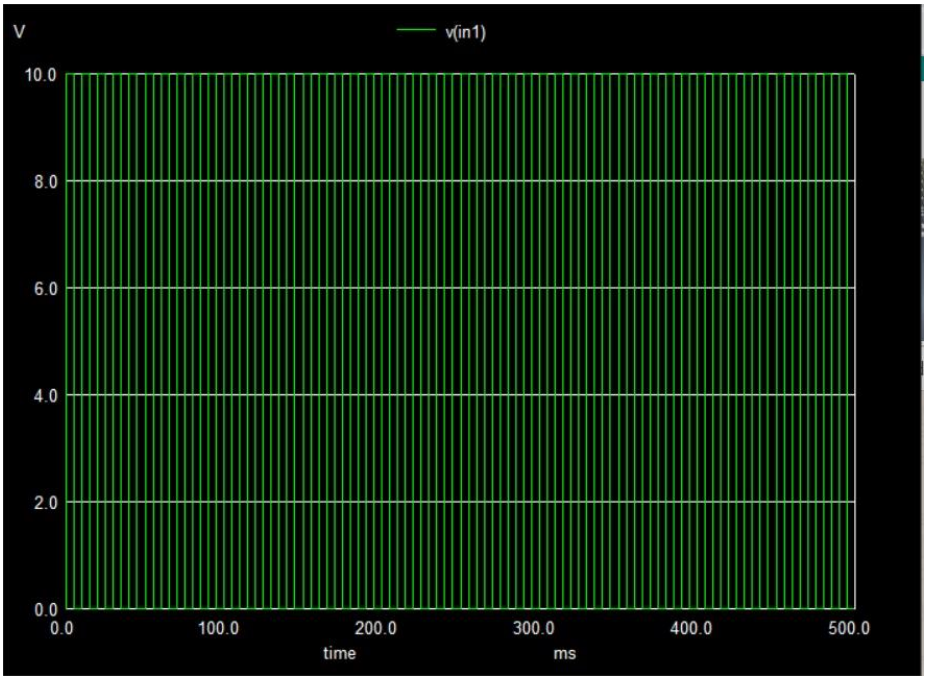
```
outp <= out_reg;


end Behavioral;
```

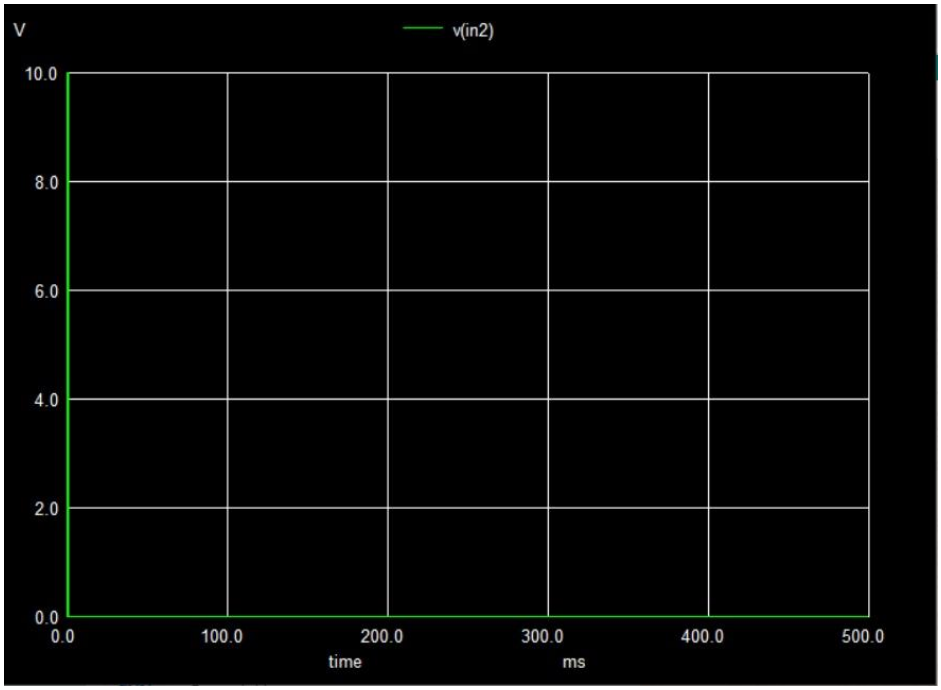
Circuit Diagram(s) :



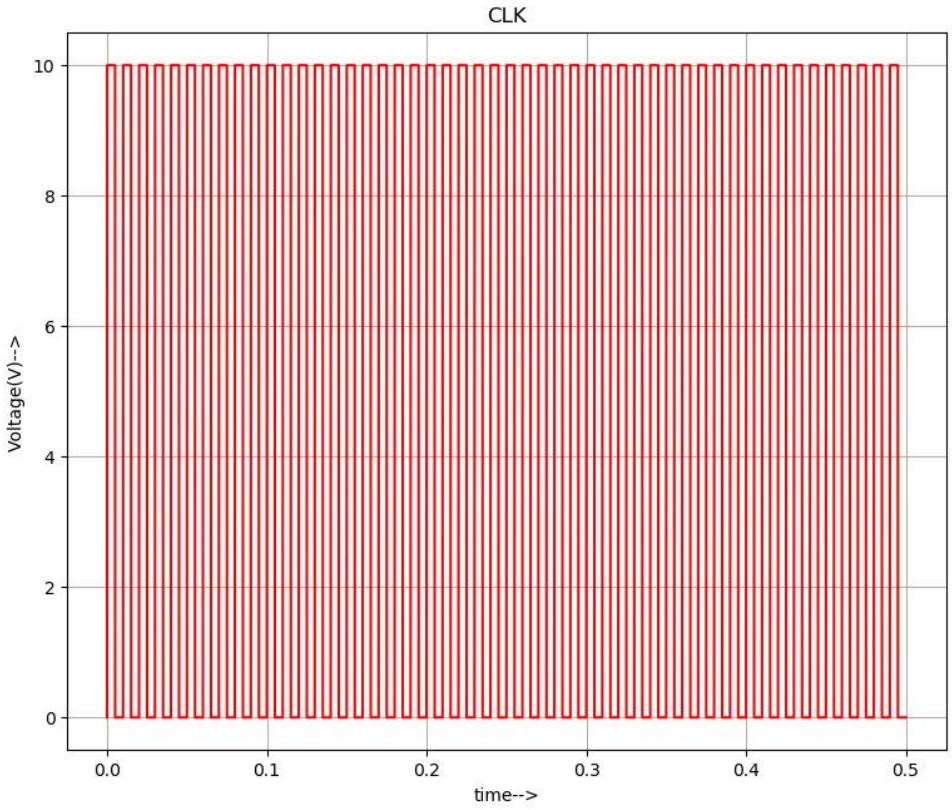
Results (Input, Output waveforms and/or Multimeter readings) :

Input :

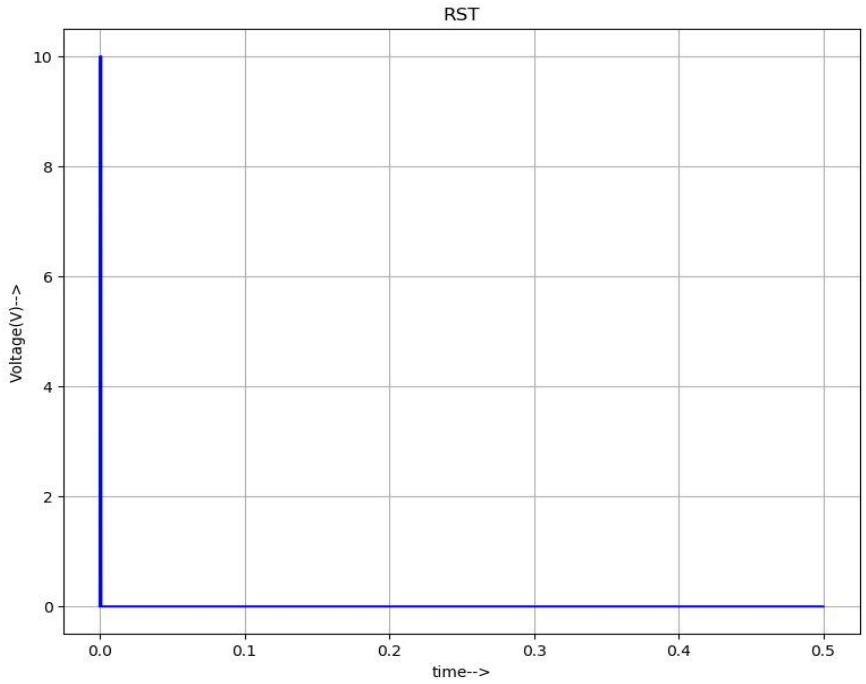



V
v(in2)
10.0
8.0
6.0
4.0
2.0
0.0
0.0
100.0
200.0
300.0
400.0
500.0
time
ms

Clock:

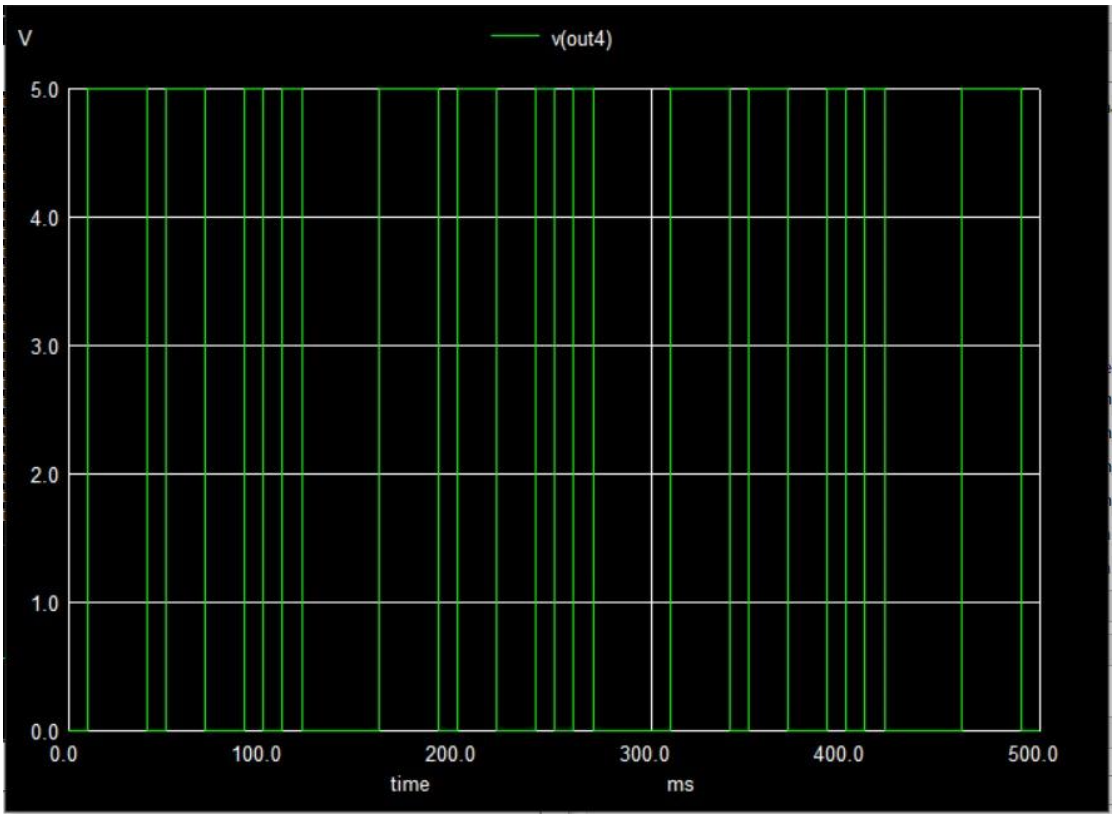

CLK
10
8
6
4
2
0
Voltage(V)-->
0.0
0.1
0.2
0.3
0.4
0.5
time-->

Reset:

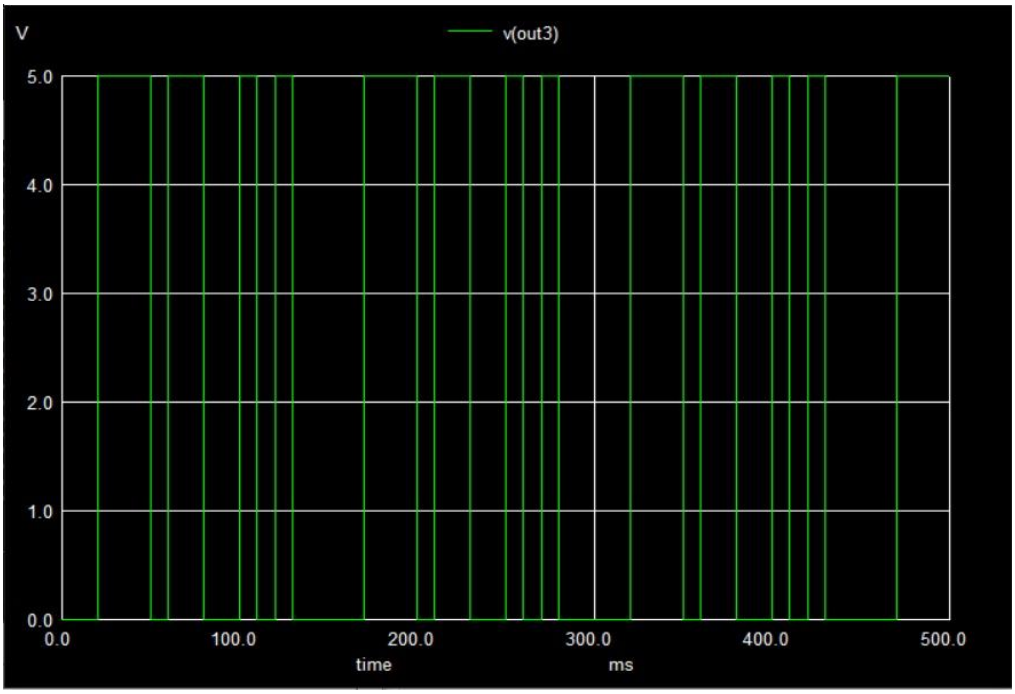


Vout:

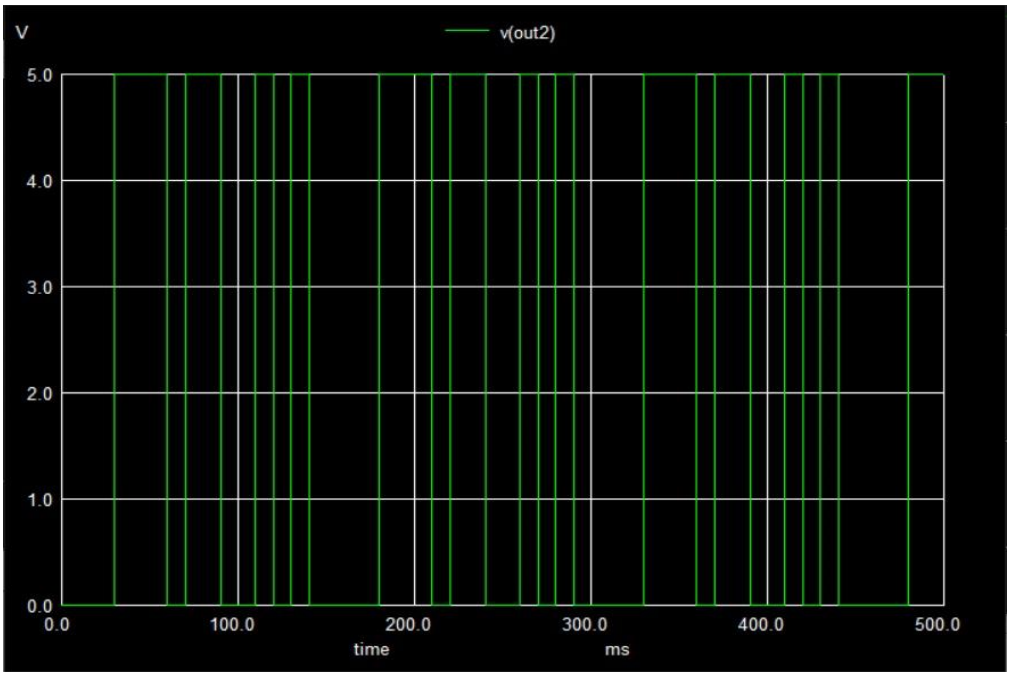
V4 :



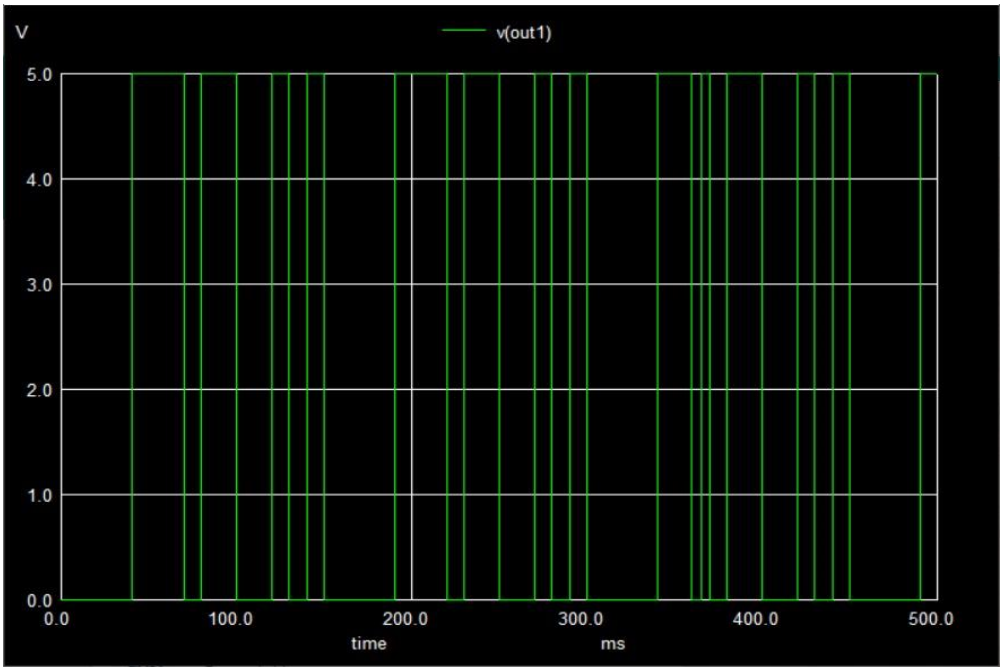
V3:


V
v(out3)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
100.0
200.0
300.0
400.0
500.0
time
ms

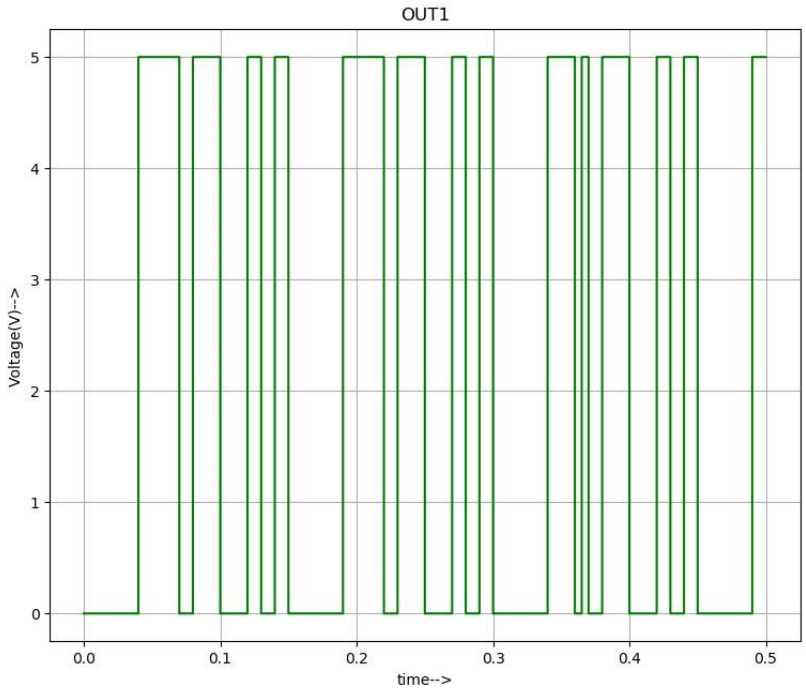
V2:

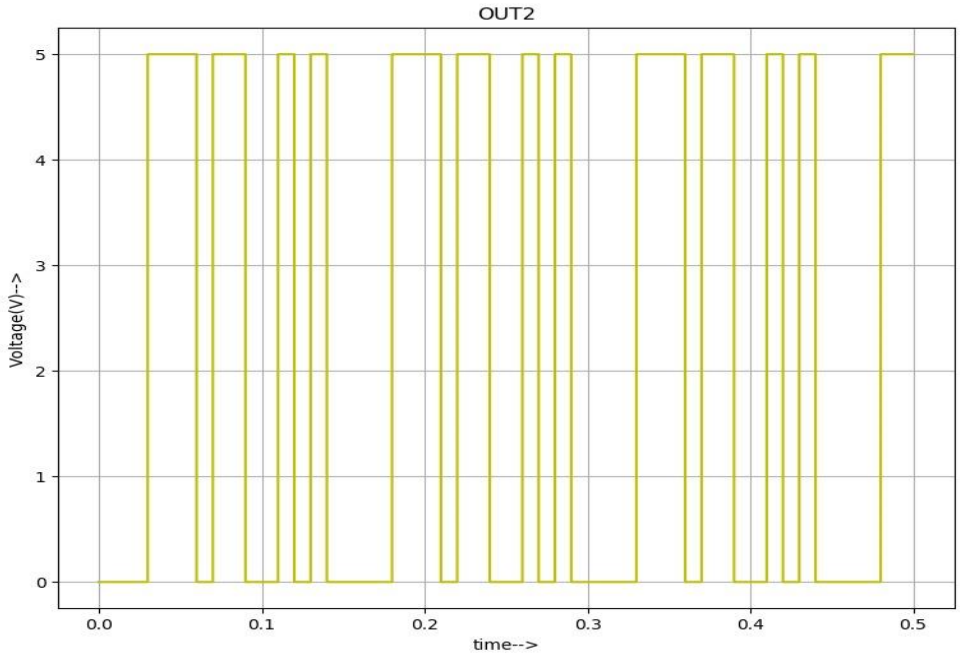

V
v(out2)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
100.0
200.0
300.0
400.0
500.0
time
ms

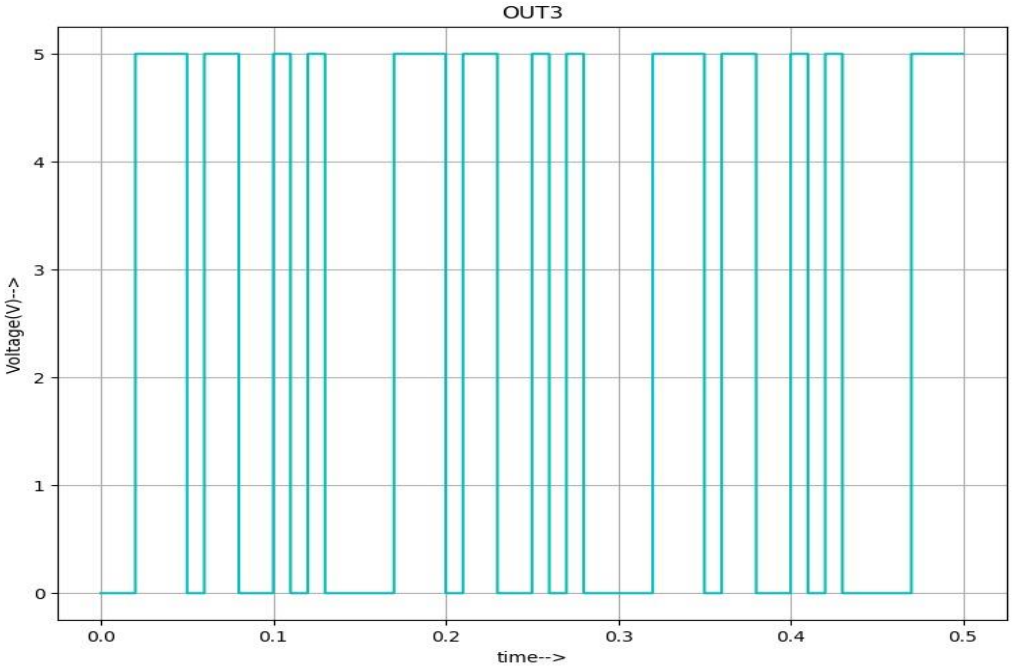
V1:

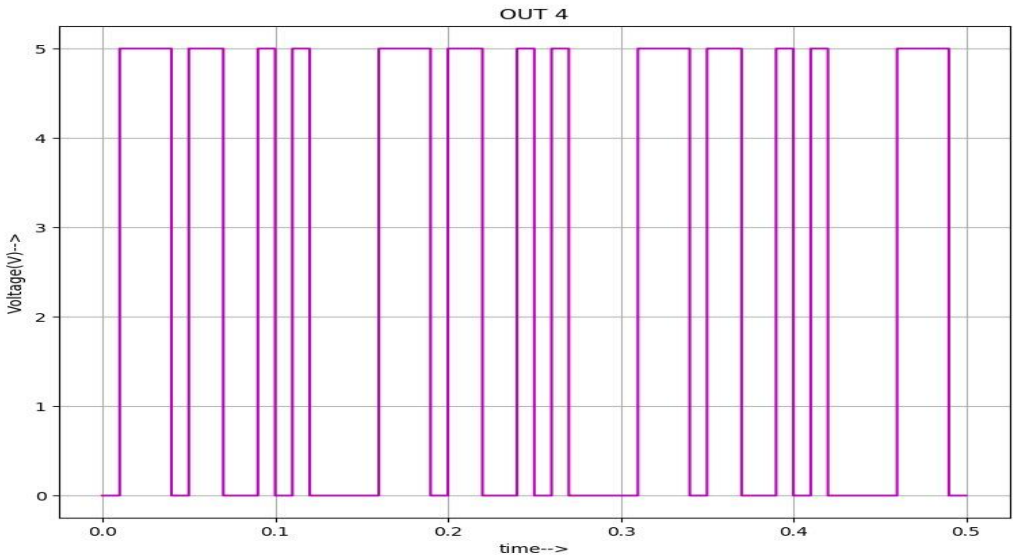

V
v(out1)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
100.0
200.0
300.0
400.0
500.0
time
ms

OUTPUT:


OUT1
Voltage(V)-->
5
4
3
2
1
0
0.0
0.1
0.2
0.3
0.4
0.5
time-->


OUT2
Voltage(V)-->
time-->
0
1
2
3
4
5
0.0
0.1
0.2
0.3
0.4
0.5


OUT3
Voltage(V)-->
time-->
0
1
2
3
4
5
0.0
0.1
0.2
0.3
0.4
0.5



Source/Reference(s) :

https://www.engineersgarage.com/vhdl/feed-back-register-in-vhdl/