Circuit Simulation Project

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

Name of the participant : S Sai Venkata Kishan Kumar

Title of the circuit : Ring-Counter

Theory/Description : A ring counter is a Shift Register (a cascade connection of flip-flops) with the output of the last flip flop connected to the input of the first. It is initialised such that only one of the flip flop output is 1 while the remainder is 0. The 1 bit is circulated so the state repeats every n clock cycles if n flip-flops are used. The "MOD" or "MODULUS" of a counter is the number of unique states. The MOD of the n flip flop ring counter is n.

Ring counters are often used in hardware design (e.g. ASCI and FPGA design) to create finite-state machines . A binary counter would require an adder circuit which is substantially more complex than a ring counter and has higher propagation delay as the number of bits increases, whereas the propagation delay of a ring counter will be nearly constant regardless of the number of bits in the code.

A general disadvantage of ring counters is that they are lower density codes than normal binary encoding of state numbers.

Straight ring counters generate fully decoded one-hot codes to that are often used to enable a specific action in each state of a cyclic control cycle.

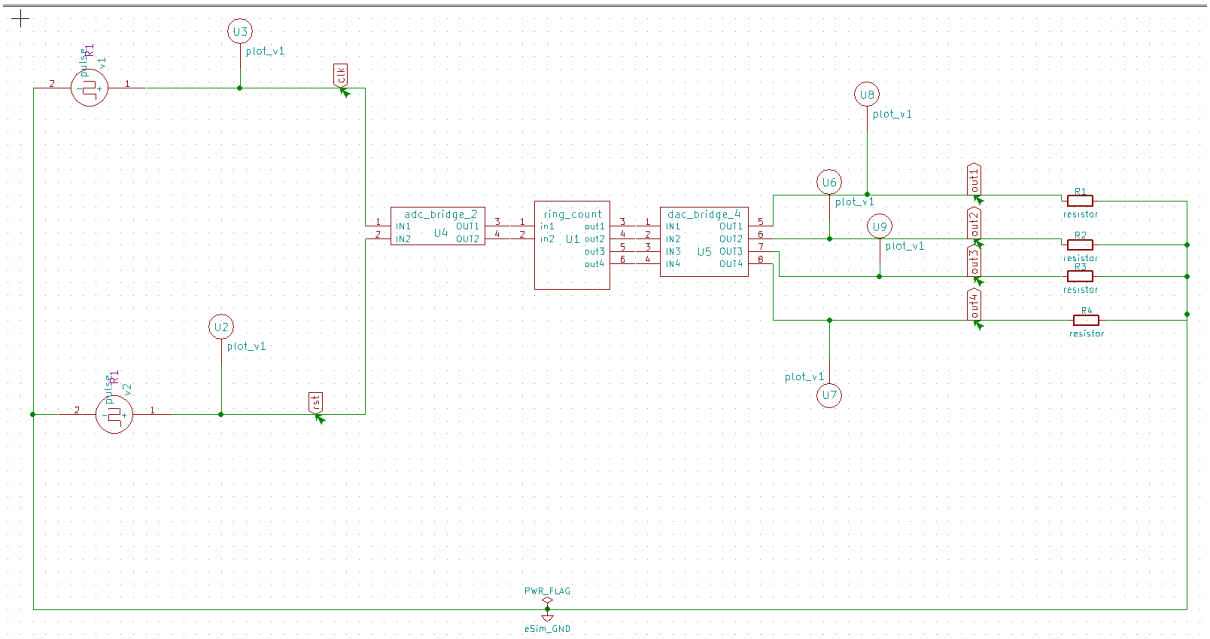
VHDL Code:

```
library ieee;
use ieee.std_logic_1164.all;

entity ring_count is
port(clk : in std_logic;
  rst : in std_logic;
  op : out std_logic_vector(3 downto 0));
end entity;

architecture beh of ring_count is
signal opt : std_logic_vector(3 downto 0);
begin
 process (clk,rst)
  begin
   if (rst = '1') then
    opt <= "1000";
   elsif (clk='1') then
    opt <= opt(0) & opt(3 downto 1);
   end if;
  end process;
  op <= opt;
end beh;
```

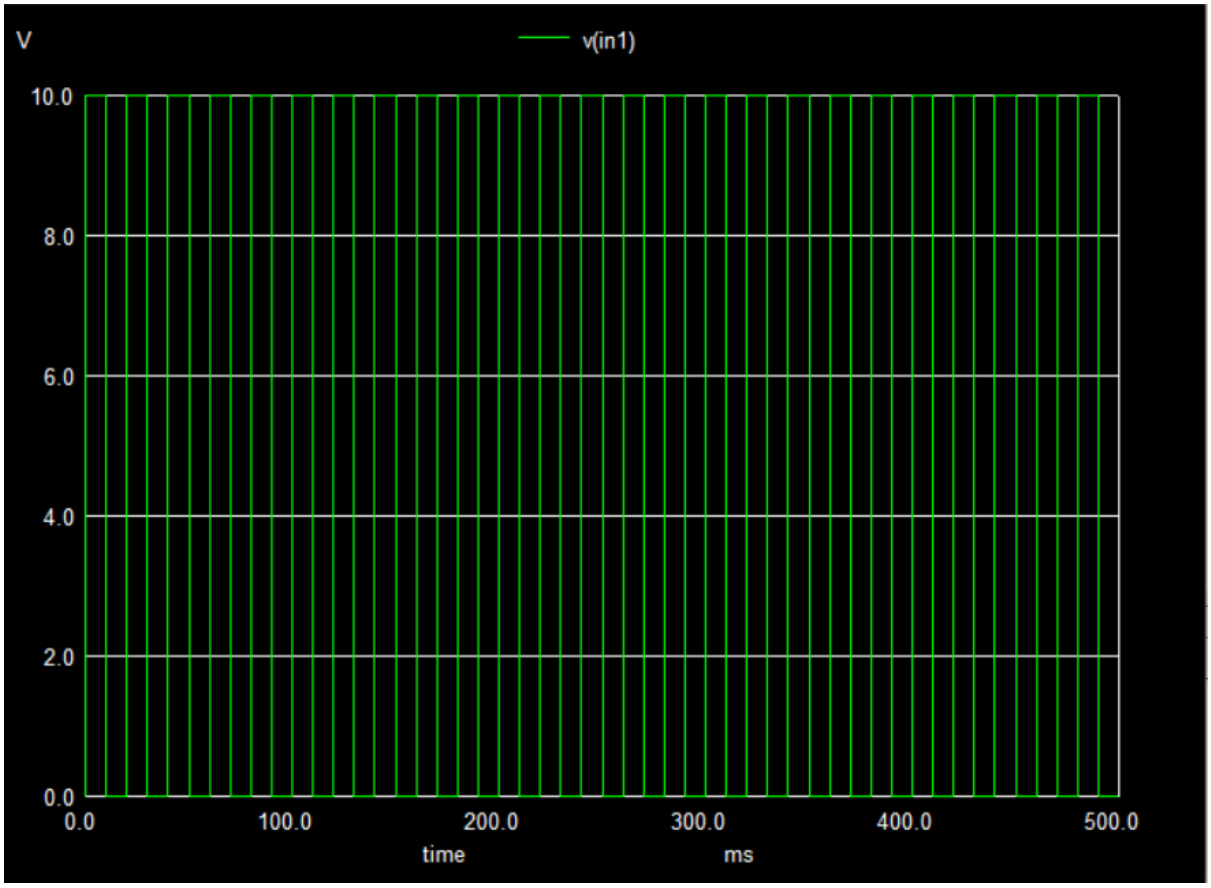
Circuit Diagram(s) :



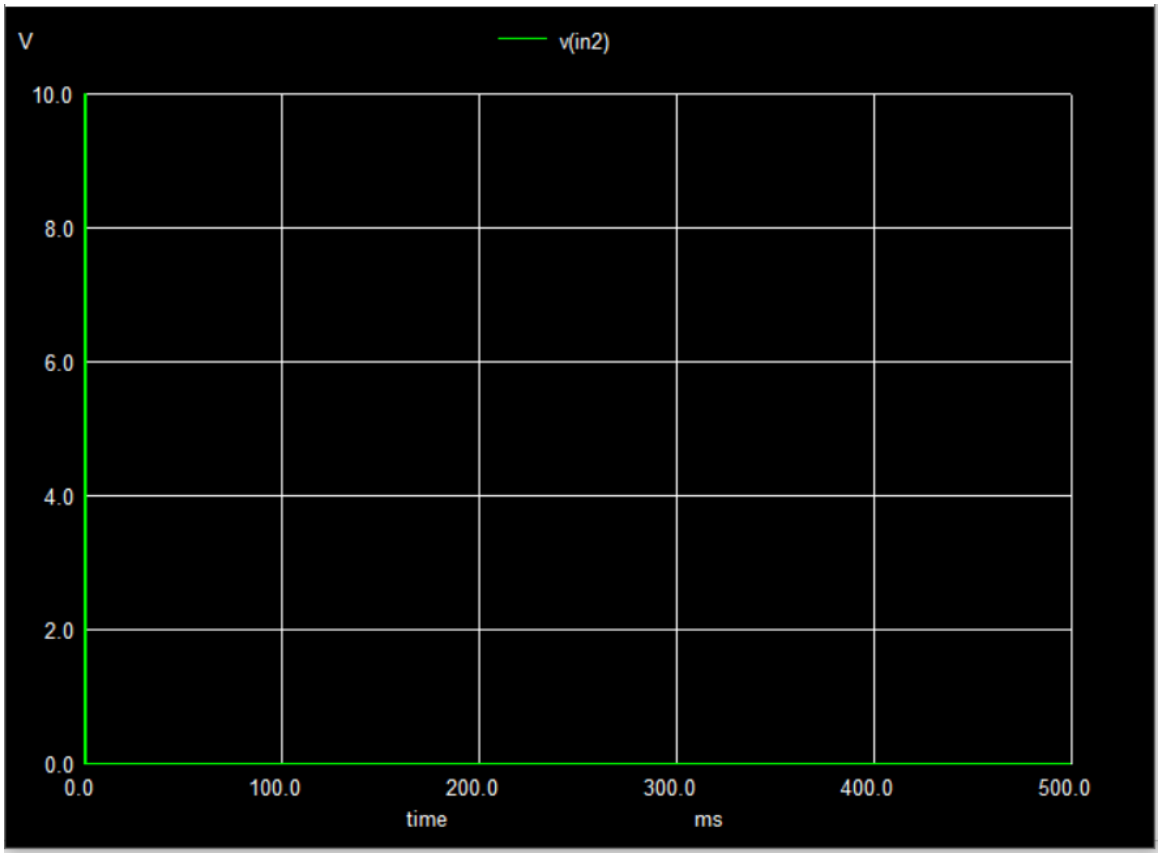
Results (Input, Output waveforms and/or Multimeter readings) :

NGSPICE PLOTS:-

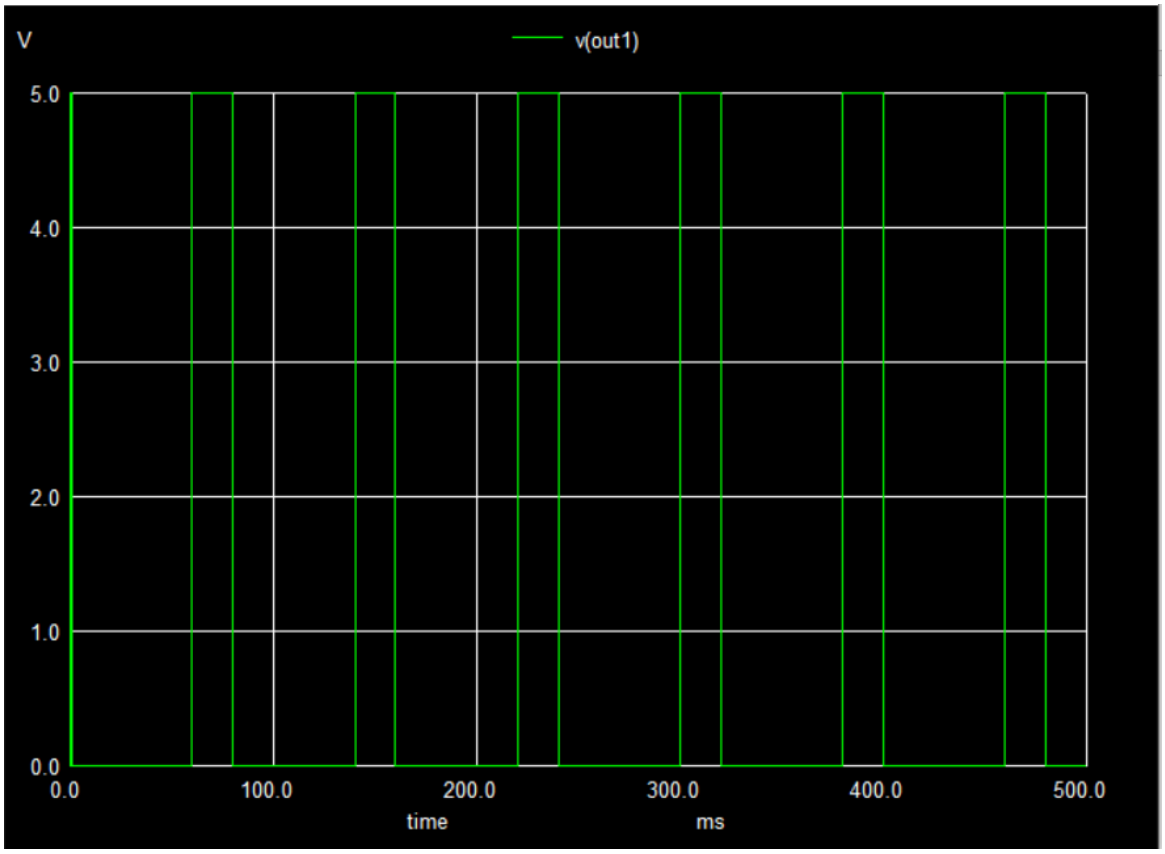
1) Plot(CLK):-


V
v(in1)
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

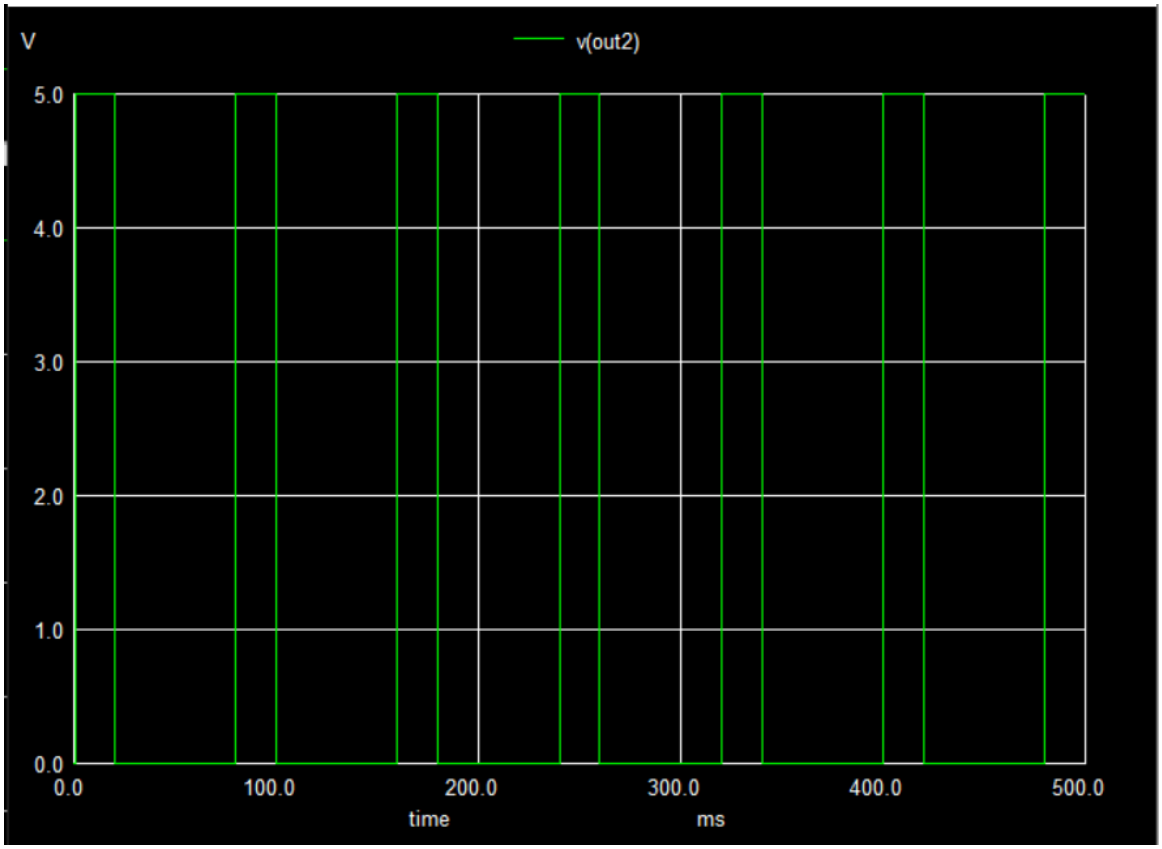
2) Plot(RST):-


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

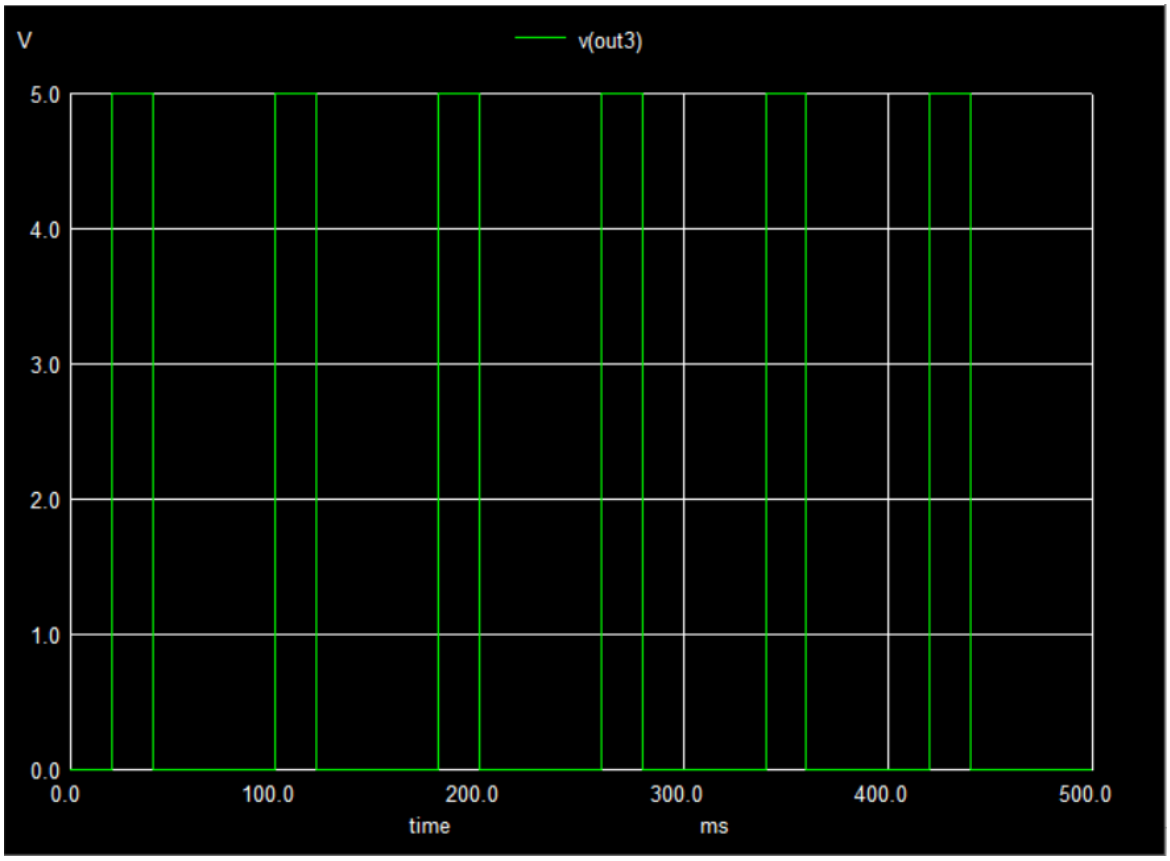
3) Plot(out(3)):-



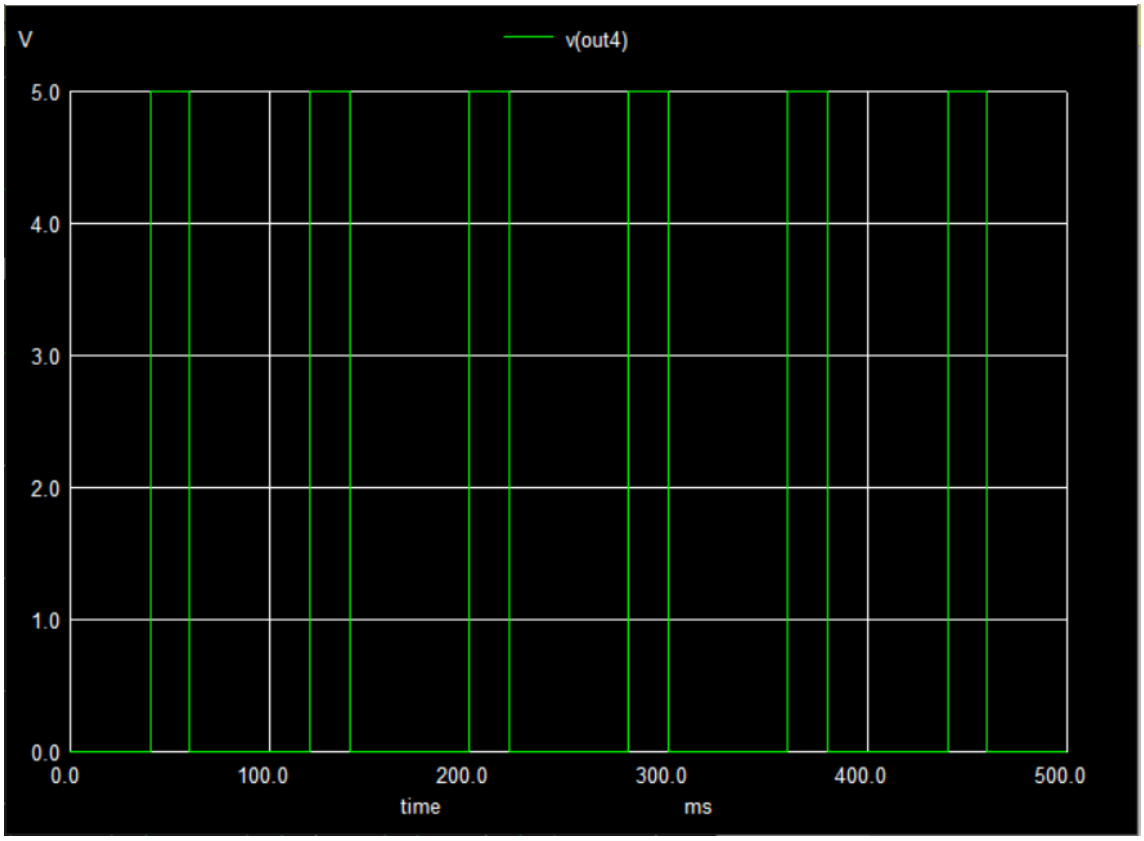
4) Plot(out(2)):-



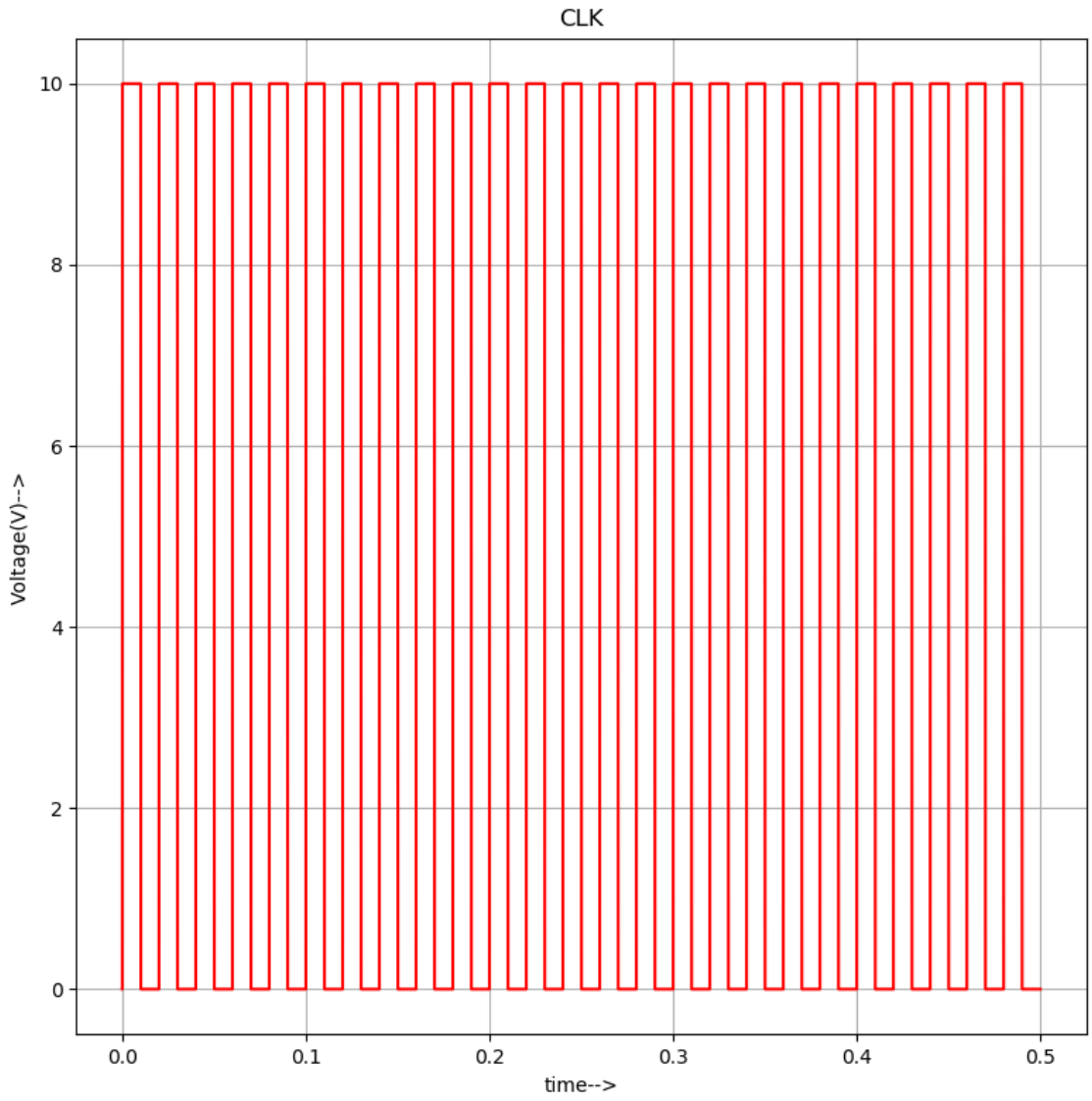
5) Plot(out(1)):-

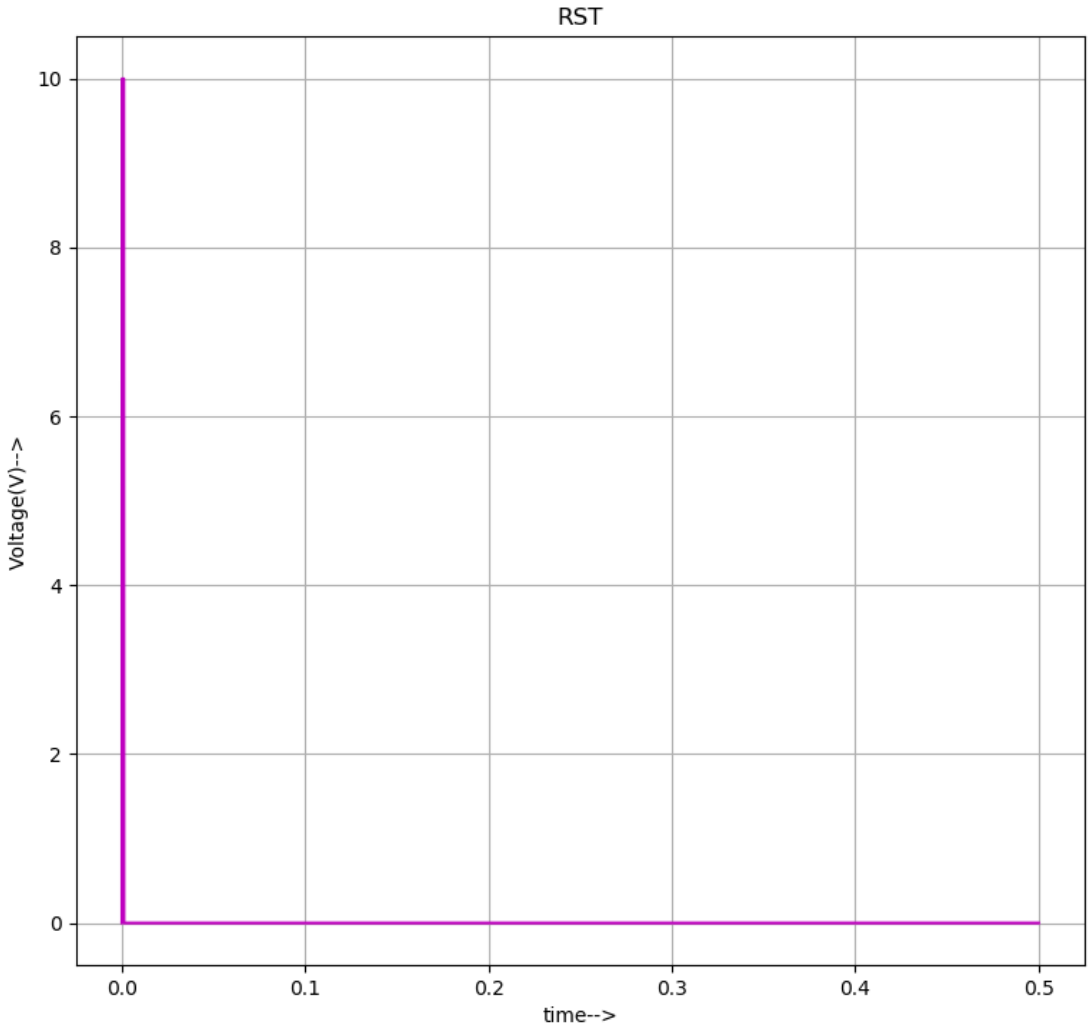


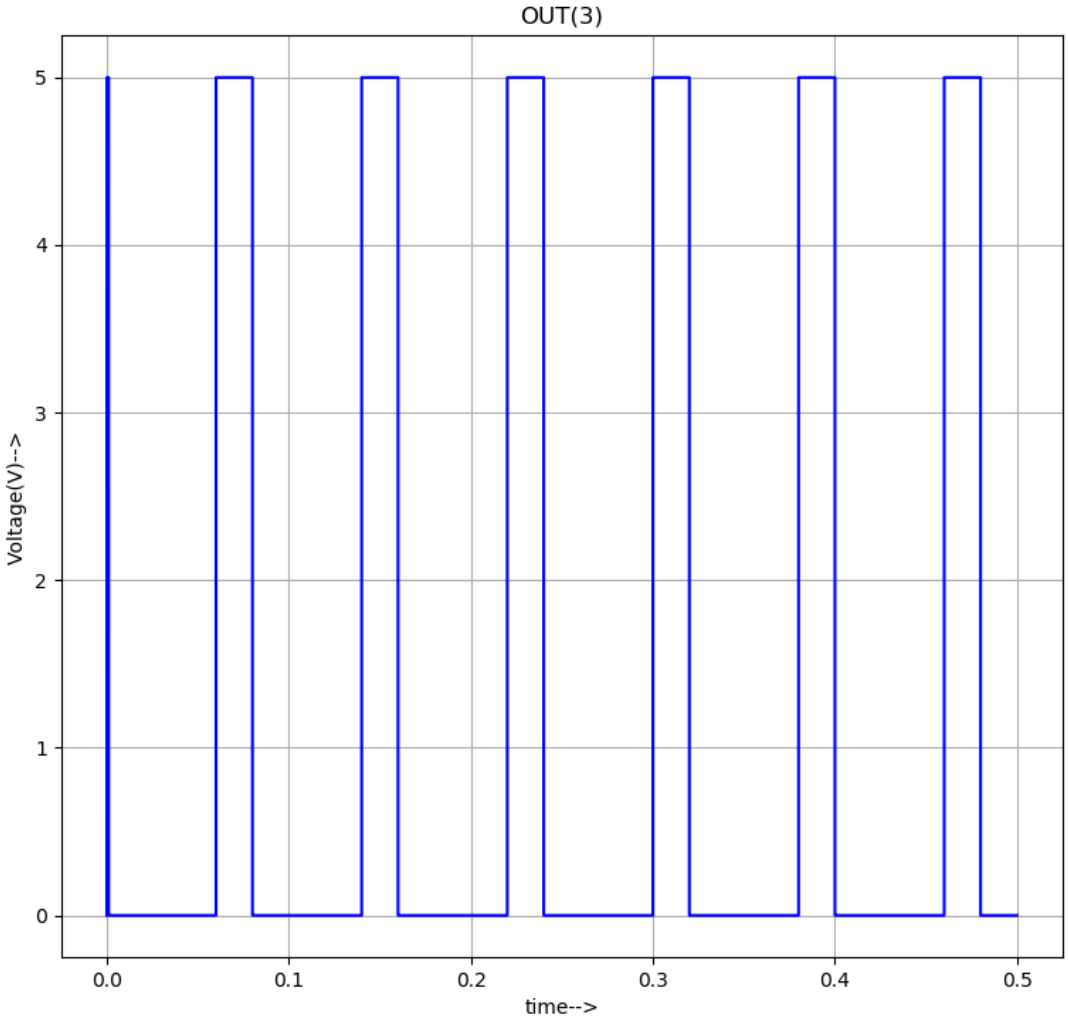
6) Plot(out(0)):-

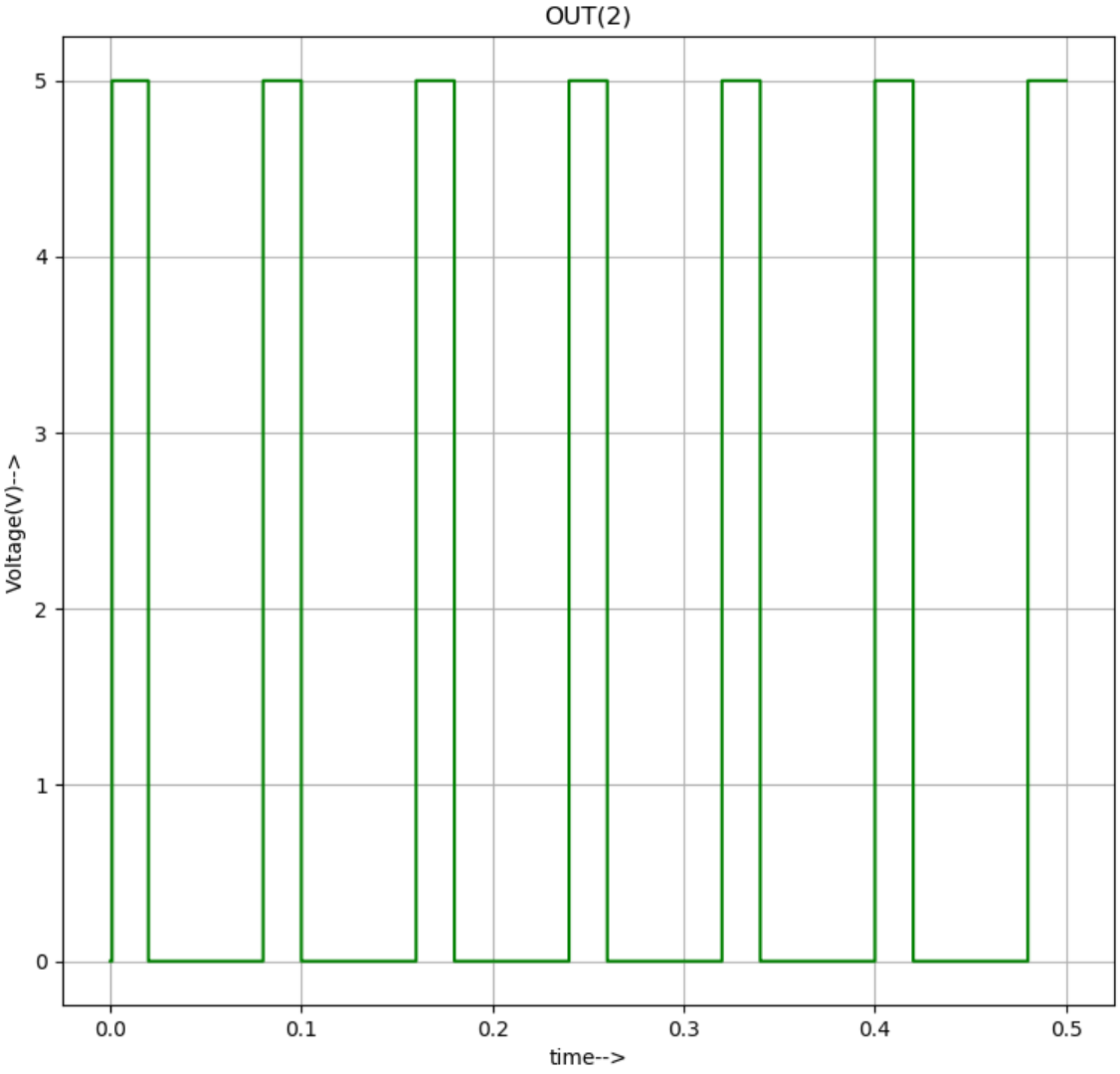


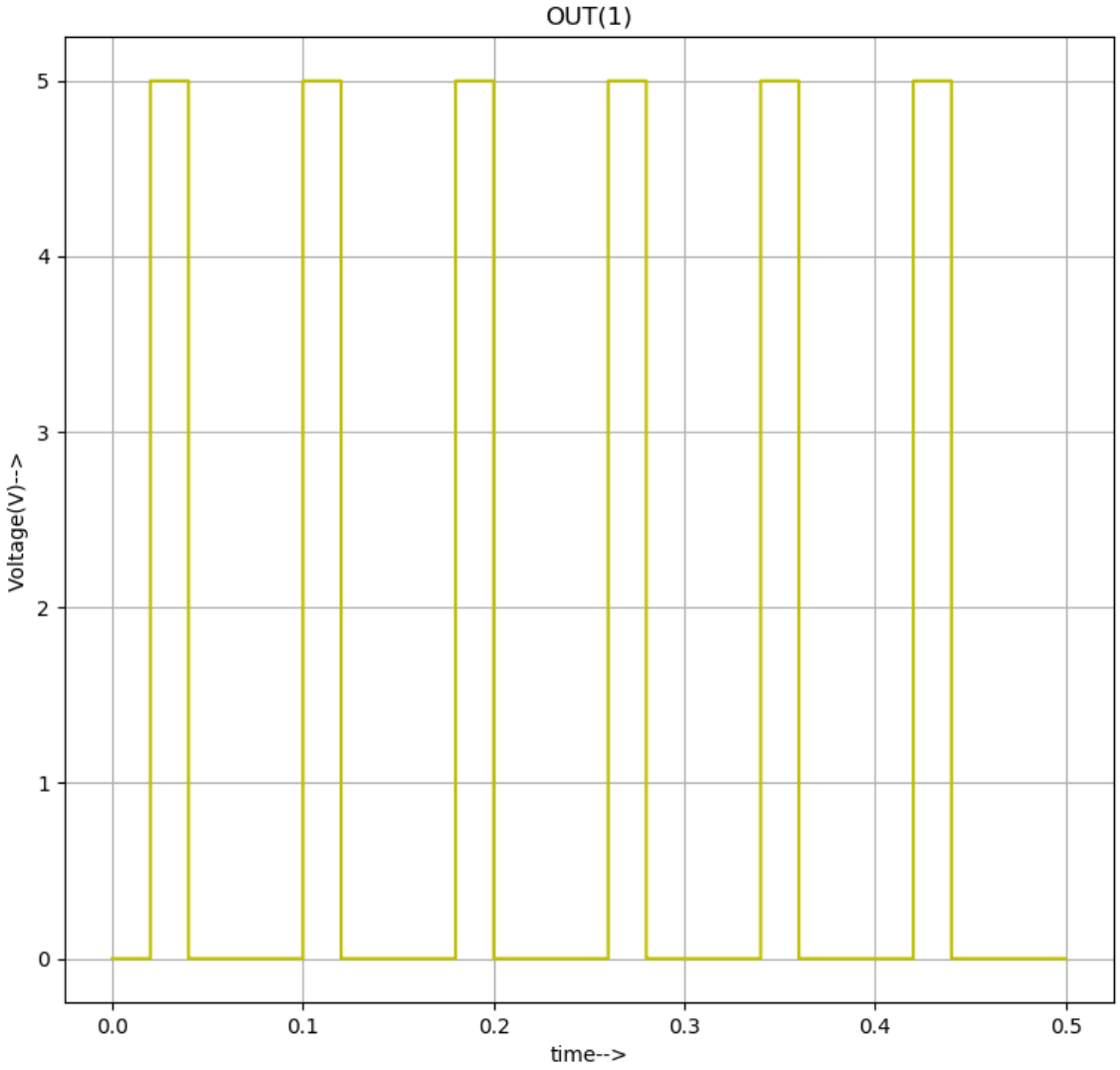
Python Plots:




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


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


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


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

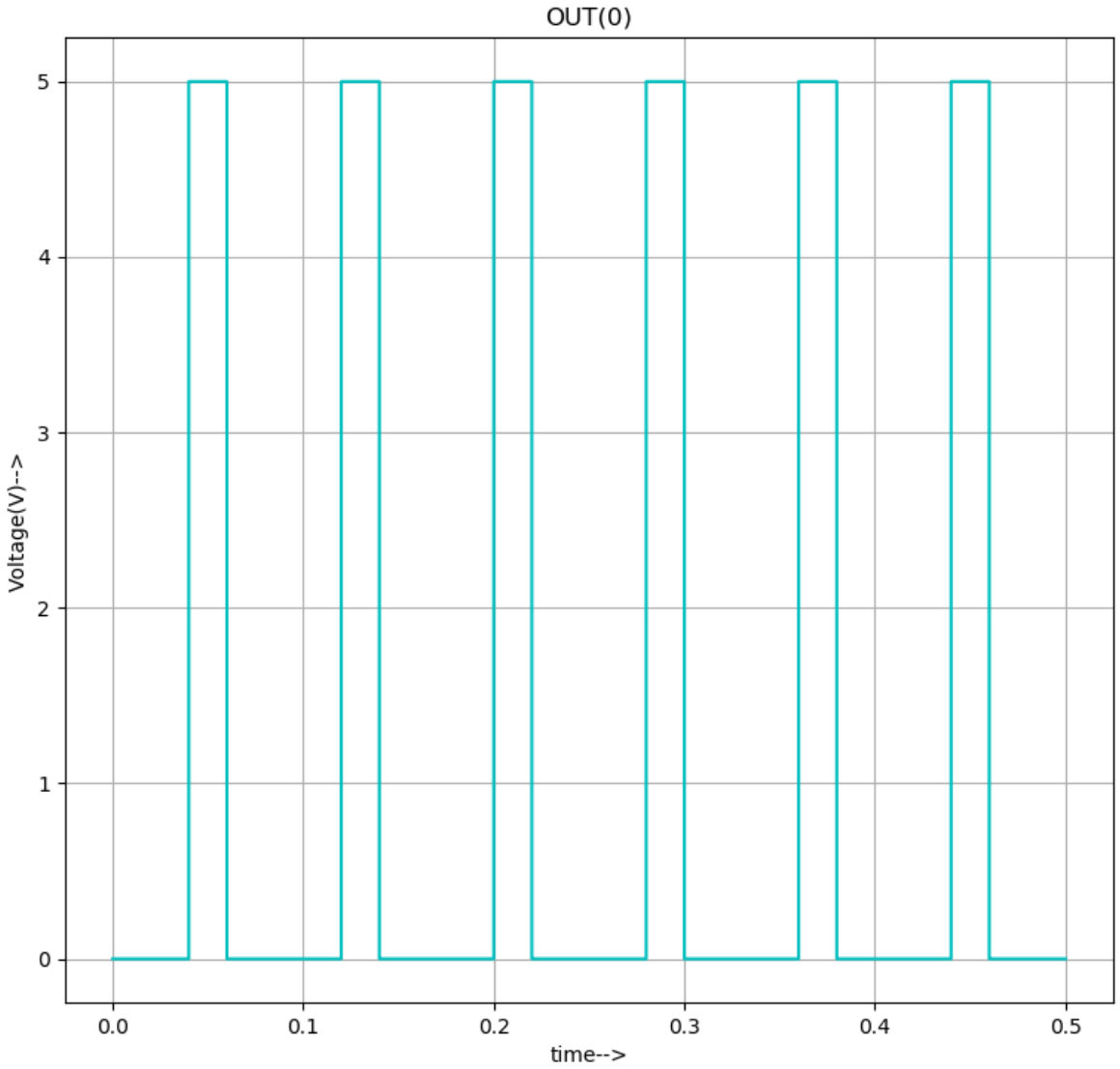


Table:

CLK	RST	OUT(3)	OUT(2)	OUT(1)	OUT(0)
1	1	1	0	0	0
0	0	1	0	0	0
1	0	0	1	0	0
0	0	0	1	0	0
1	0	0	0	1	0
0	0	0	0	1	0
1	0	0	0	0	1

Source/Reference(s) :