# bit ALU :

## VHDL Code :

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity alu is

Port ( a : in STD\_LOGIC\_VECTOR (3 downto 0); b : in STD\_LOGIC\_VECTOR (3 downto 0);

s : in STD\_LOGIC\_VECTOR (2 downto 0);

y : out STD\_LOGIC\_VECTOR (3 downto 0)); end alu;

architecture Behavioral of alu is

begin process(a,b,s) begin

case s is

when "000"=>y<=a+b; when "001"=>y<=a-b;

when "010"=>y<=a and b; when "011"=>y<=a or b; when "100"=>y<=not b; when "101"=>y<=a xor b;

when others=>y<="0000"; end case;

end process;

end Behavioral;

## TestBench code :

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.std\_logic\_unsigned.all; USE ieee.numeric\_std.ALL;

ENTITY alu1247\_vhd IS END alu1247\_vhd;

ARCHITECTURE behavior OF alu1247\_vhd IS

-- Component Declaration for the Unit Under Test (UUT) COMPONENT alu1447

PORT(

a : IN std\_logic\_vector(3 downto 0); b : IN std\_logic\_vector(3 downto 0); s : IN std\_logic\_vector(2 downto 0);

y : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

SIGNAL a : std\_logic\_vector(3 downto 0) := (others=>'0'); SIGNAL b : std\_logic\_vector(3 downto 0) := (others=>'0'); SIGNAL s : std\_logic\_vector(2 downto 0) := (others=>'0');

--Outputs

SIGNAL y : std\_logic\_vector(3 downto 0);

BEGIN

-- Instantiate the Unit Under Test (UUT) uut: alu1447 PORT MAP(

a => a, b => b, s => s, y => y

);

tb : PROCESS BEGIN

a<="1100"; b<="0000";

s<="000";

wait for 100 ns; s<="001";

waitfor 100 ns; s<="010";

wait for 100 ns;

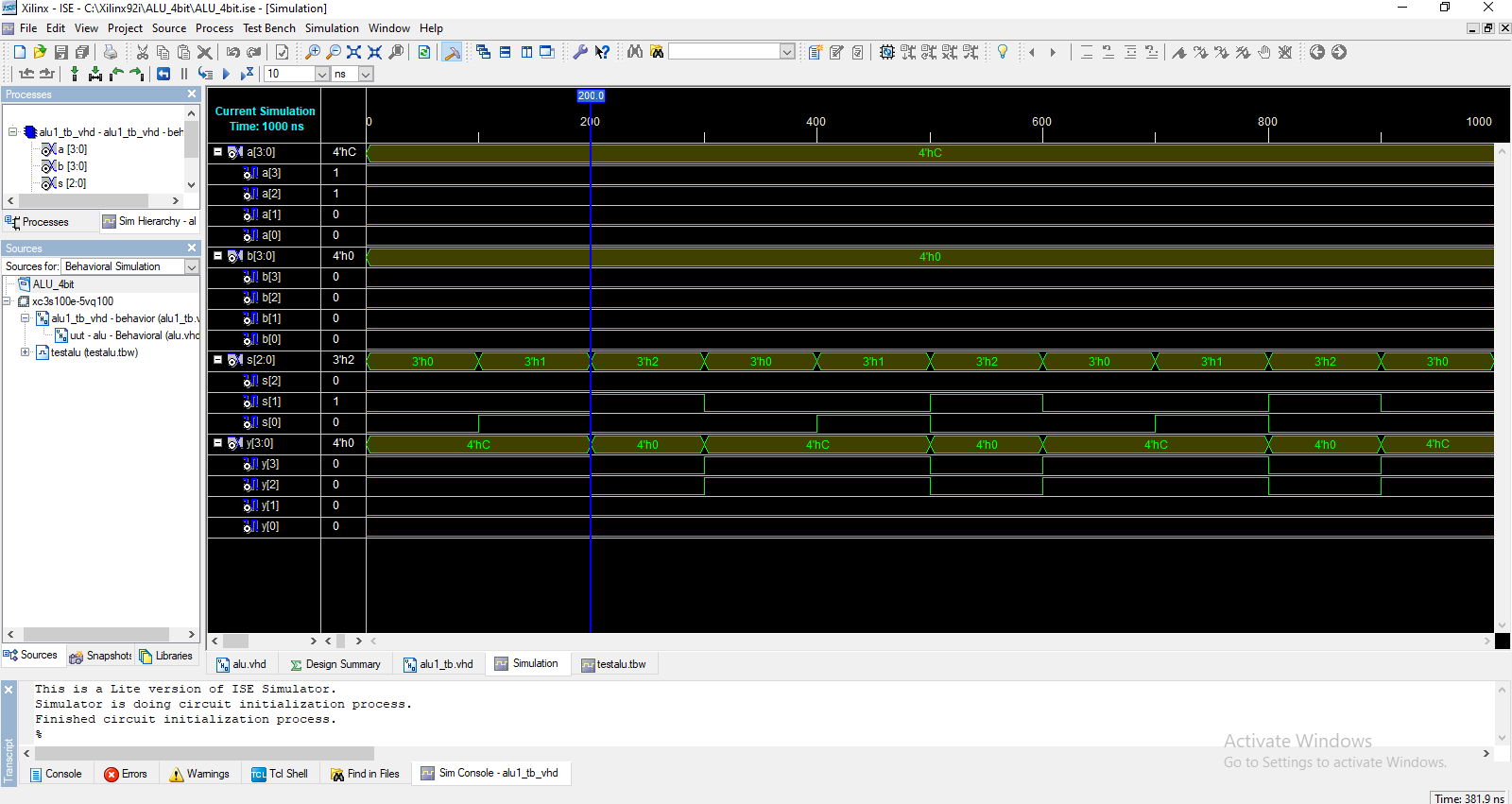
-- Wait 100 ns for global reset to finish

--wait for 100 ns;

-- Place stimulus here

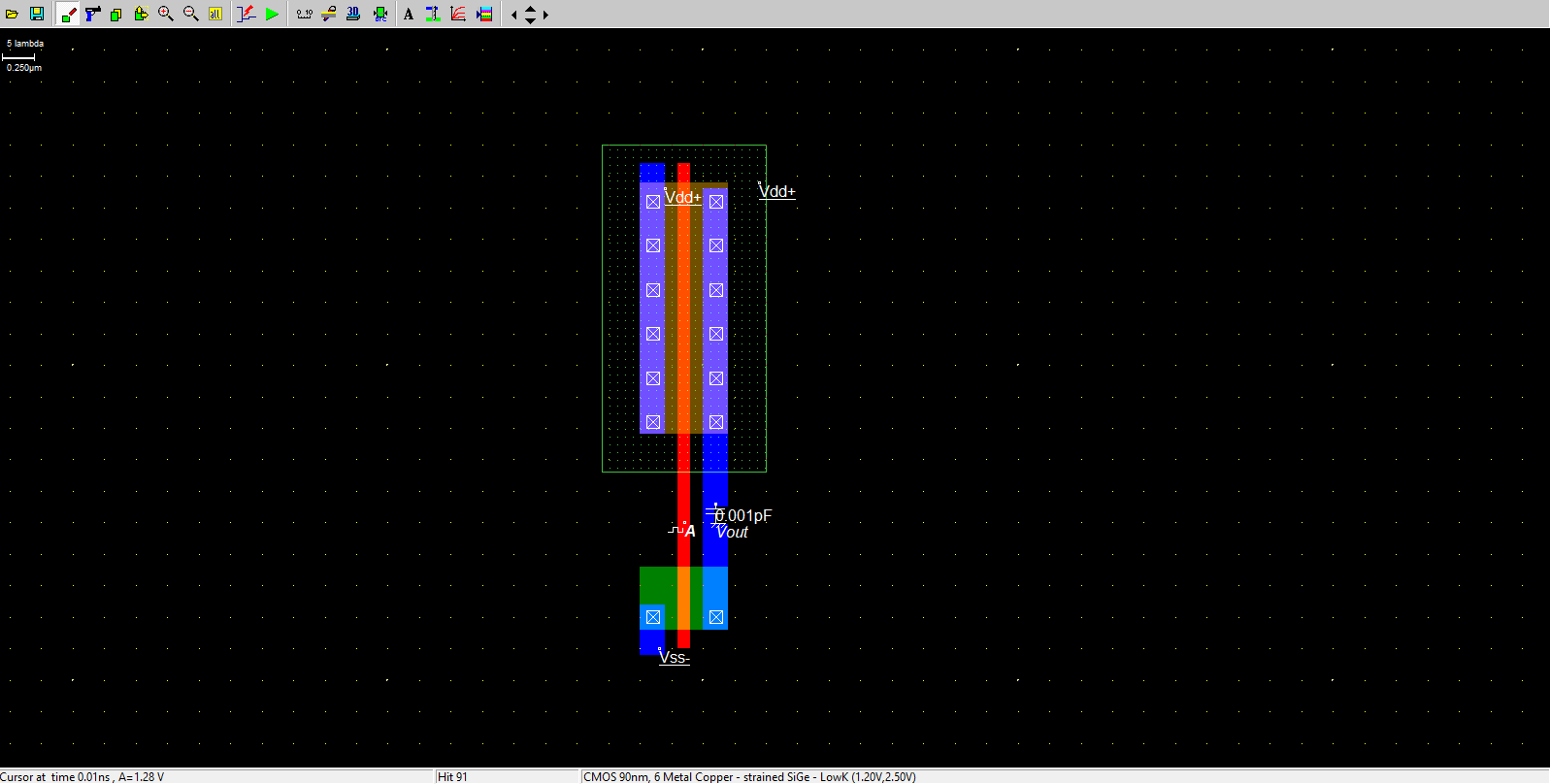
wait; -- will wait forever END PROCESS;

END;

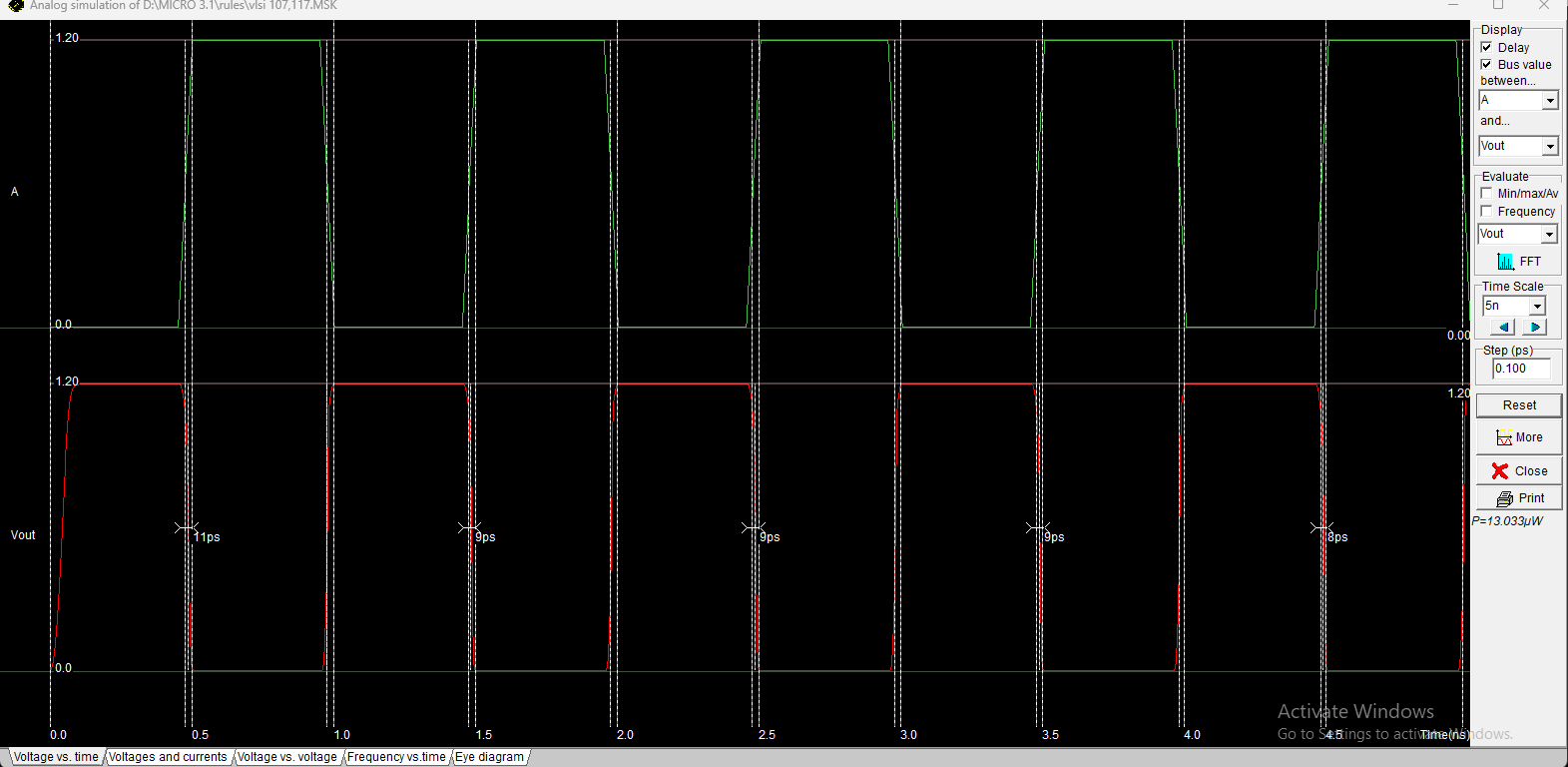


# CMOS Inverter Gate :

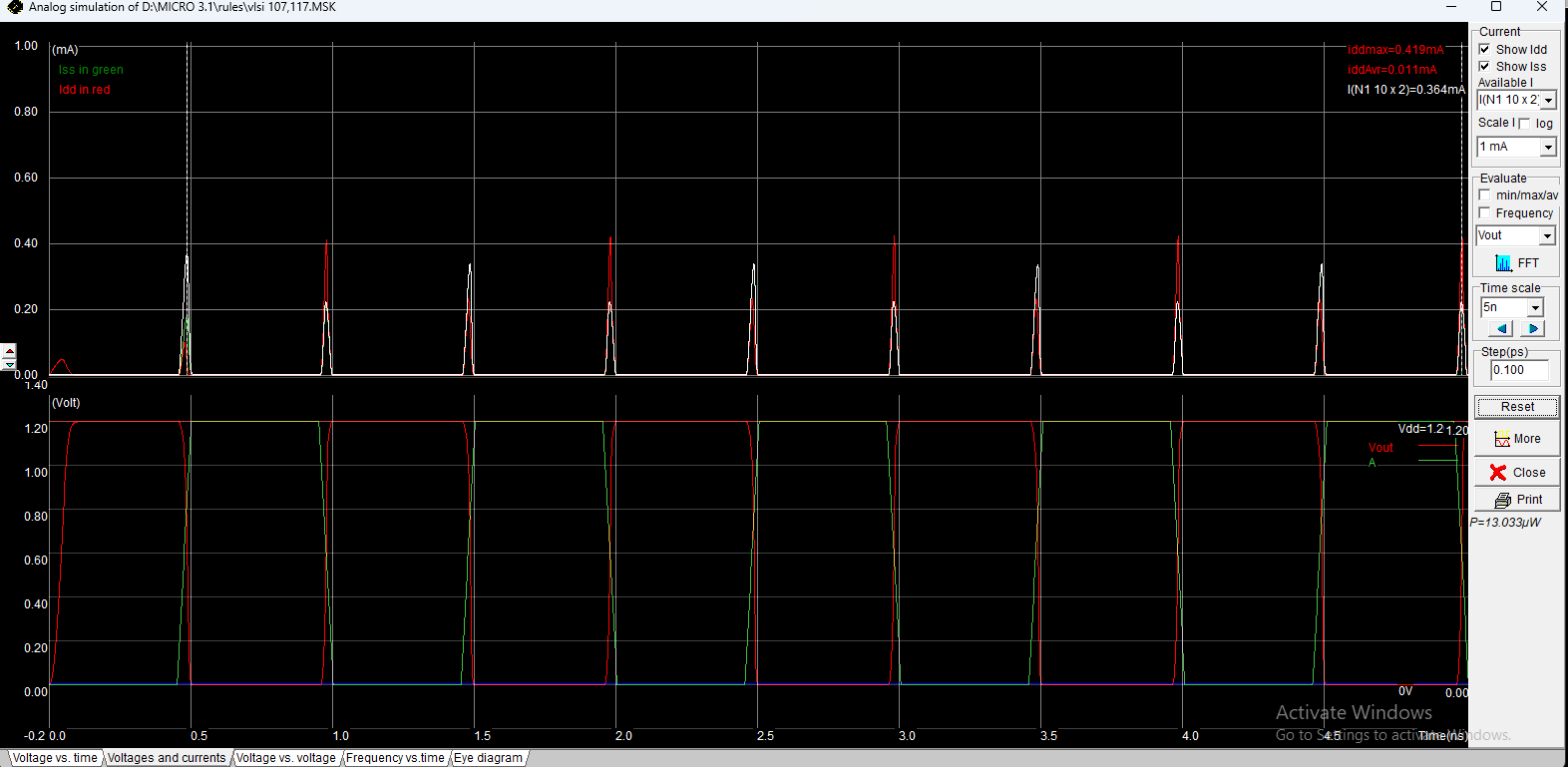
## CMOS Inverter :



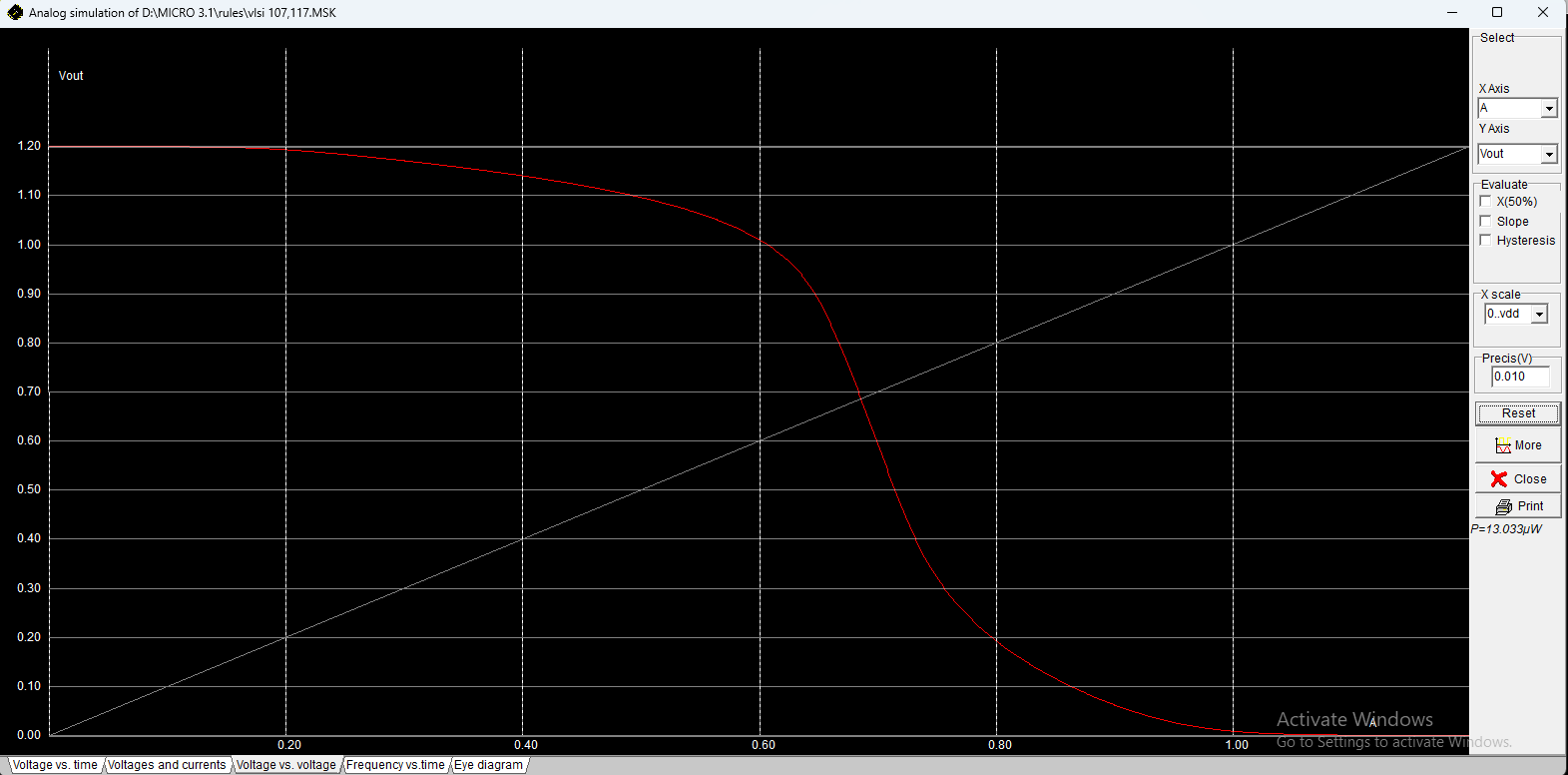
Voltage vs Time -



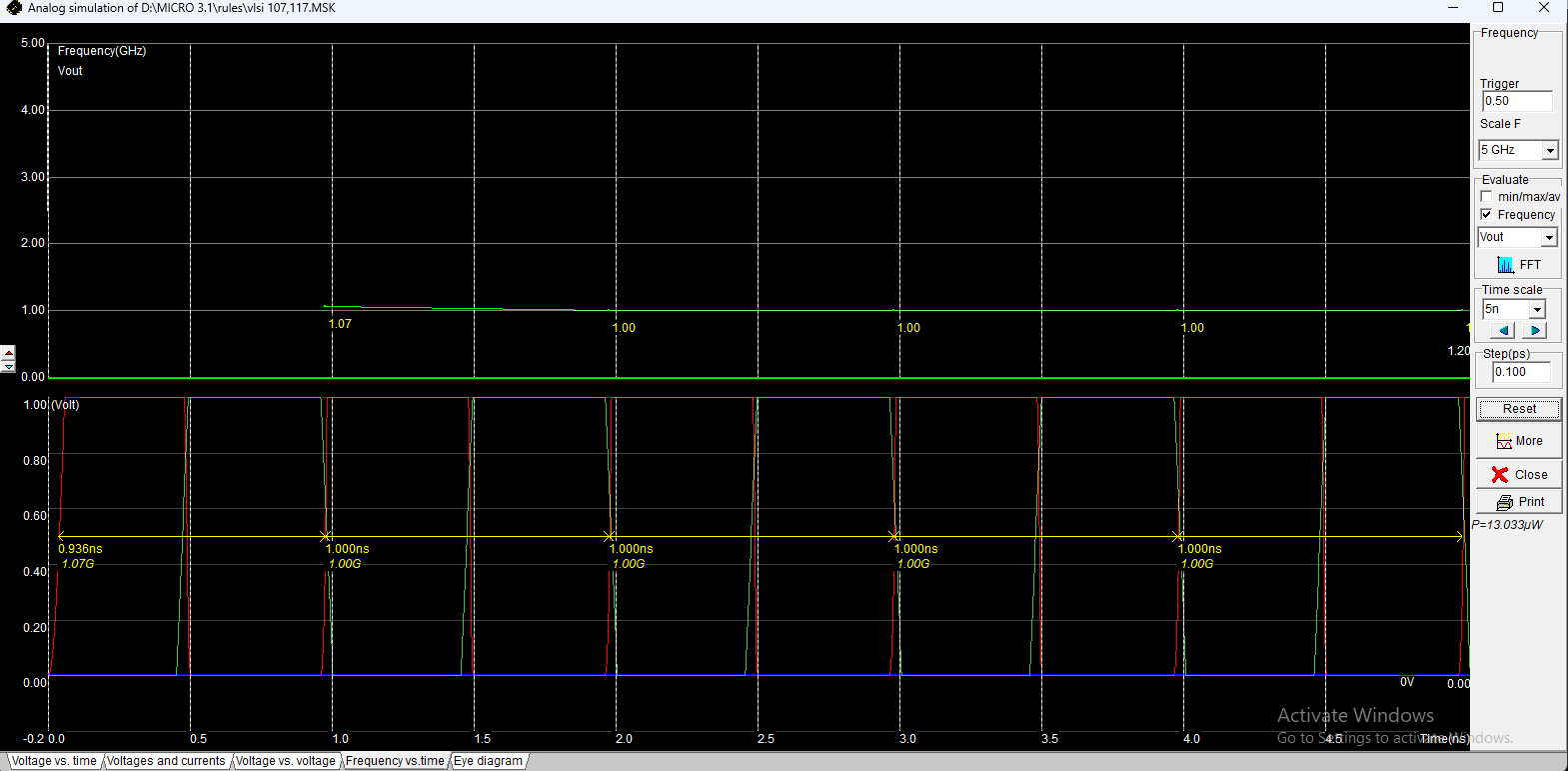
Voltage and Current -



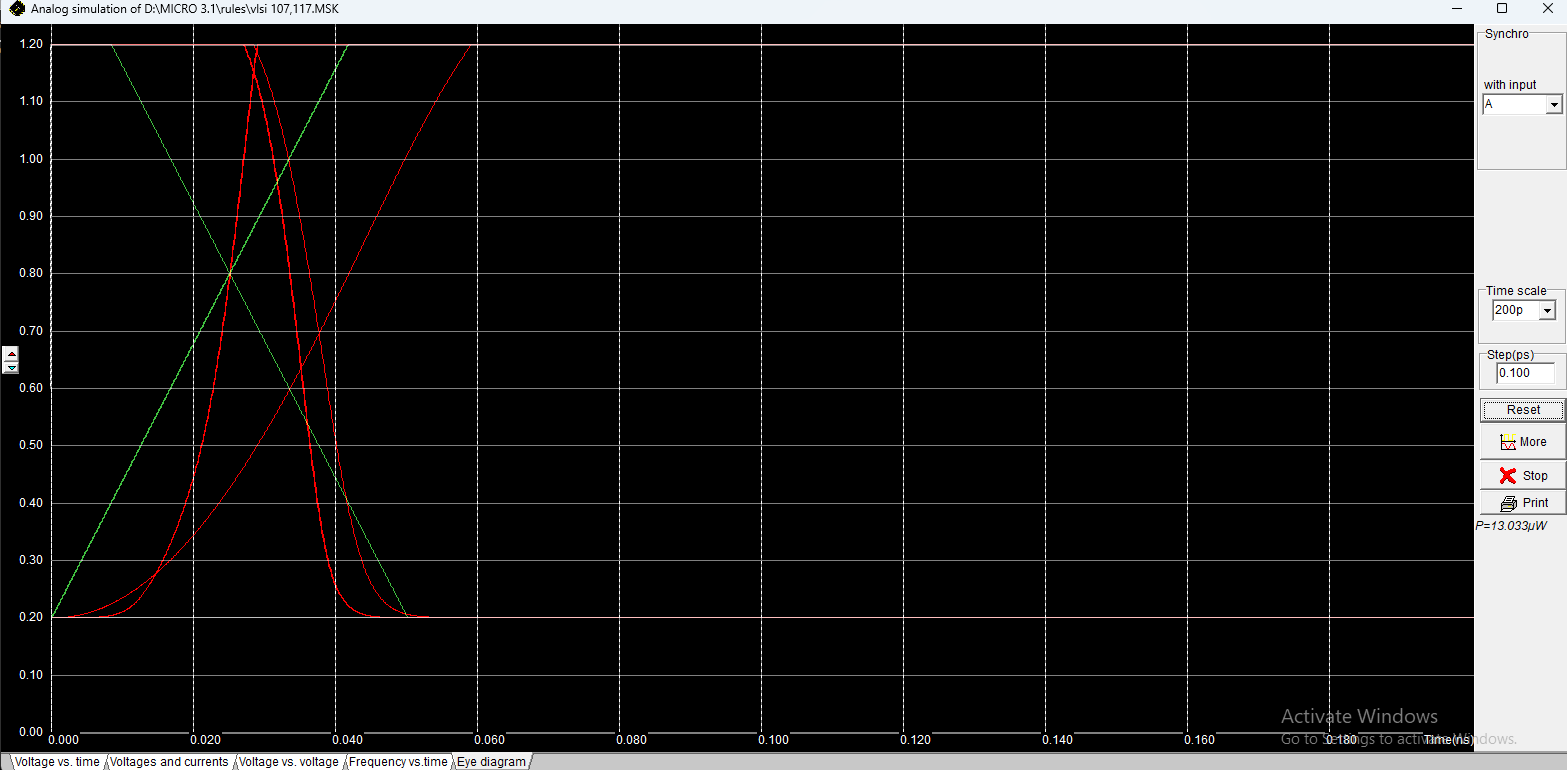
Voltage vs Voltage -



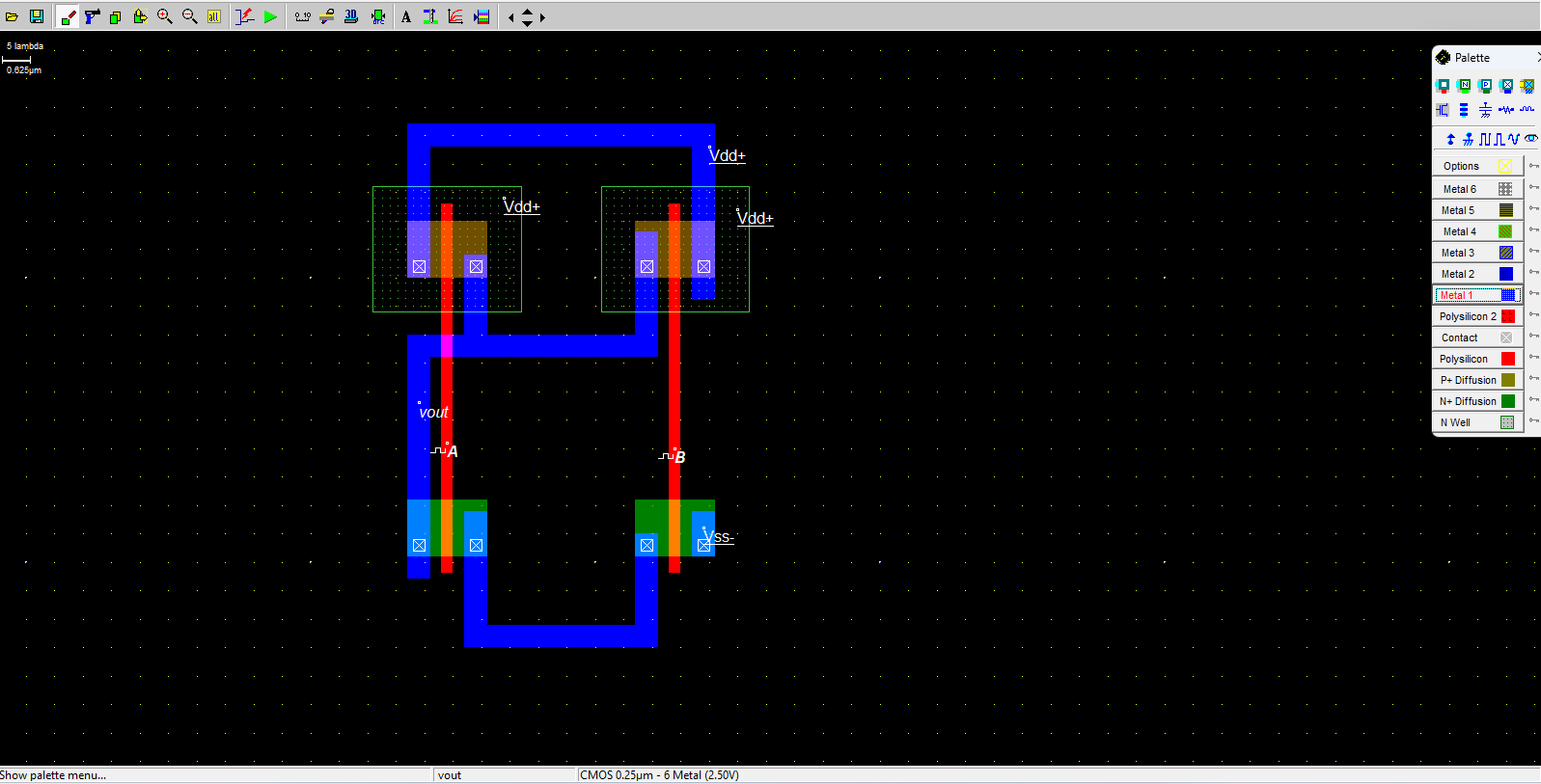
Freq vs time –



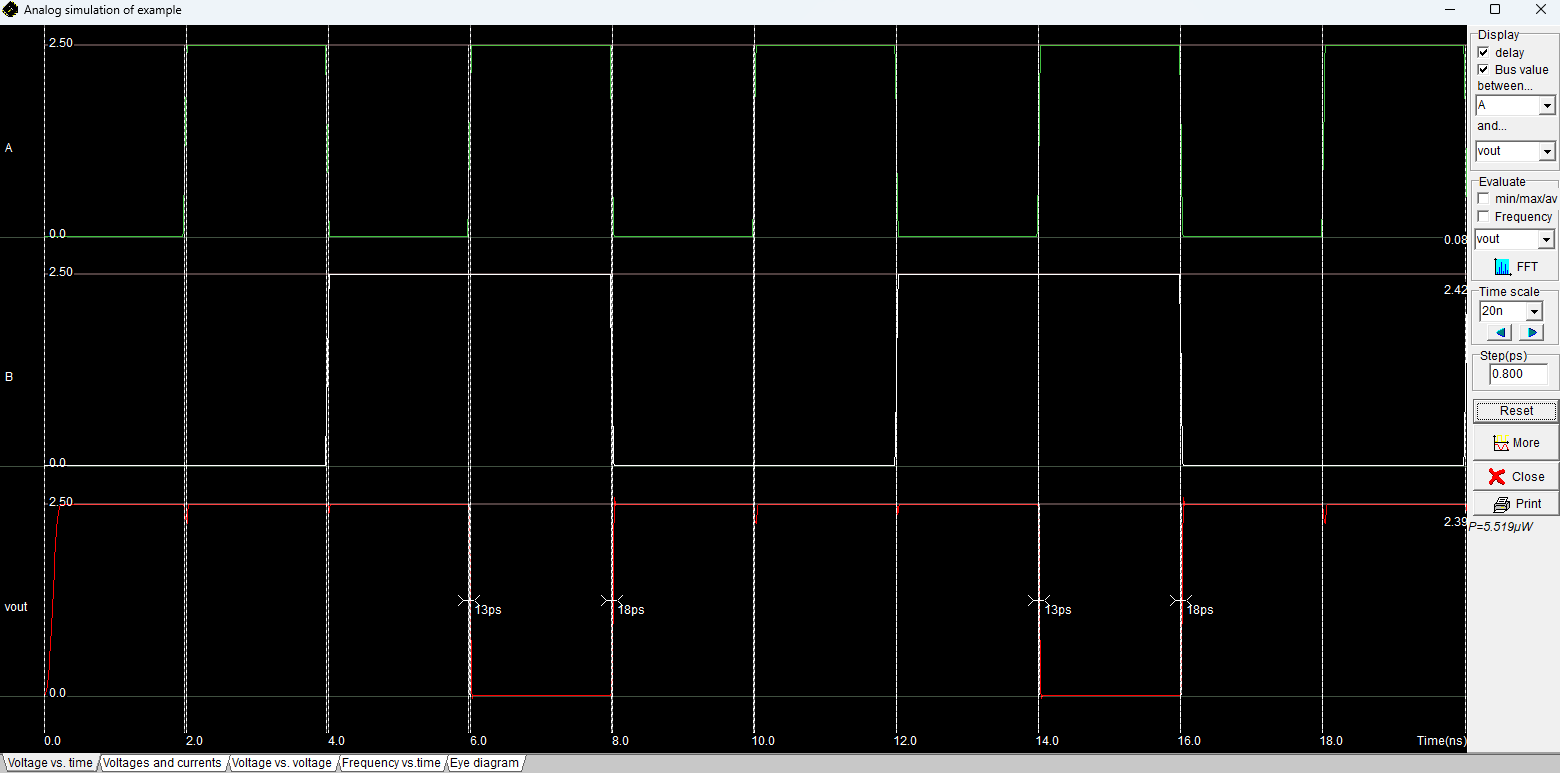
Eye diagram –



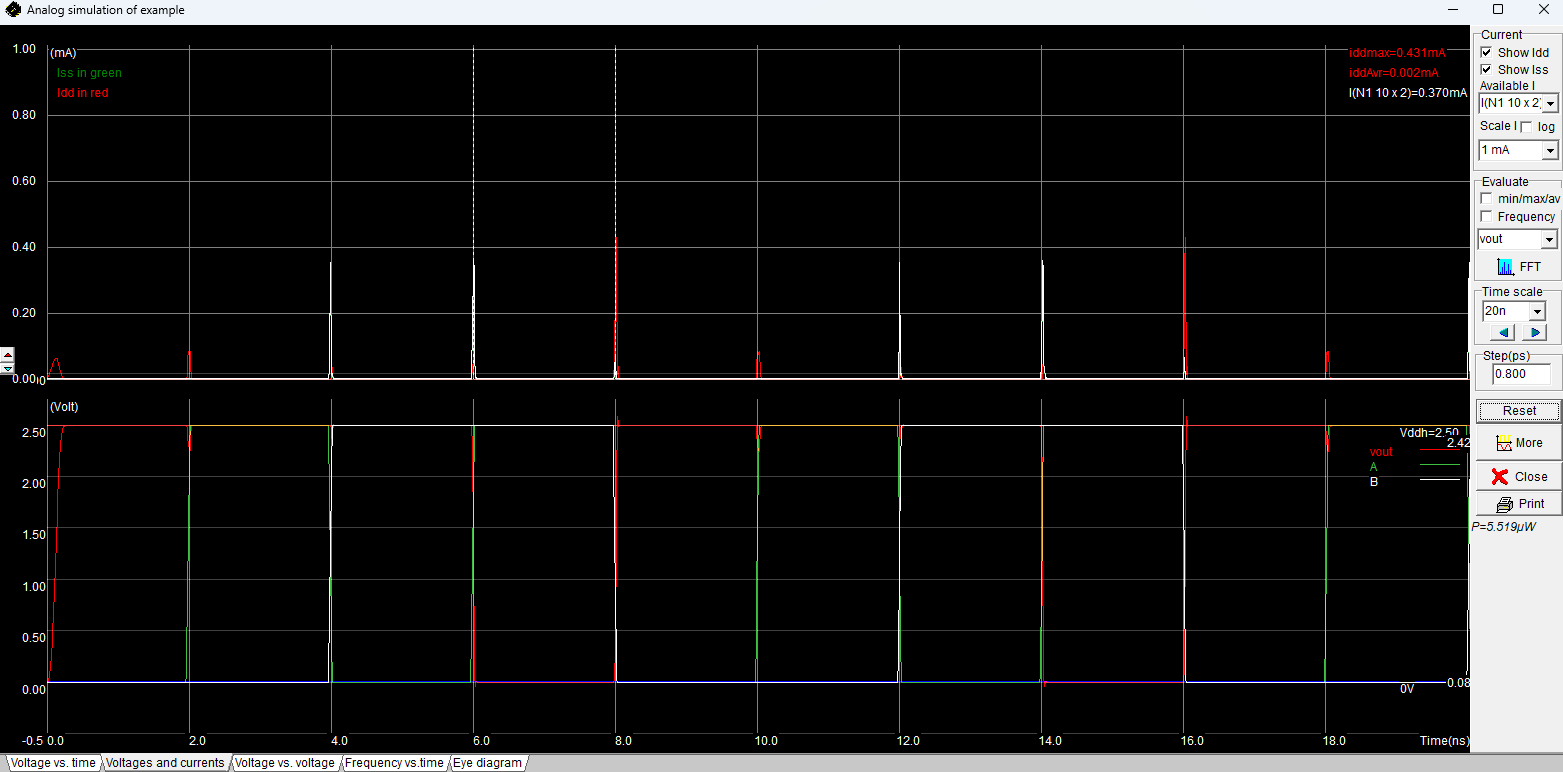
## CMOS NAND Gate :



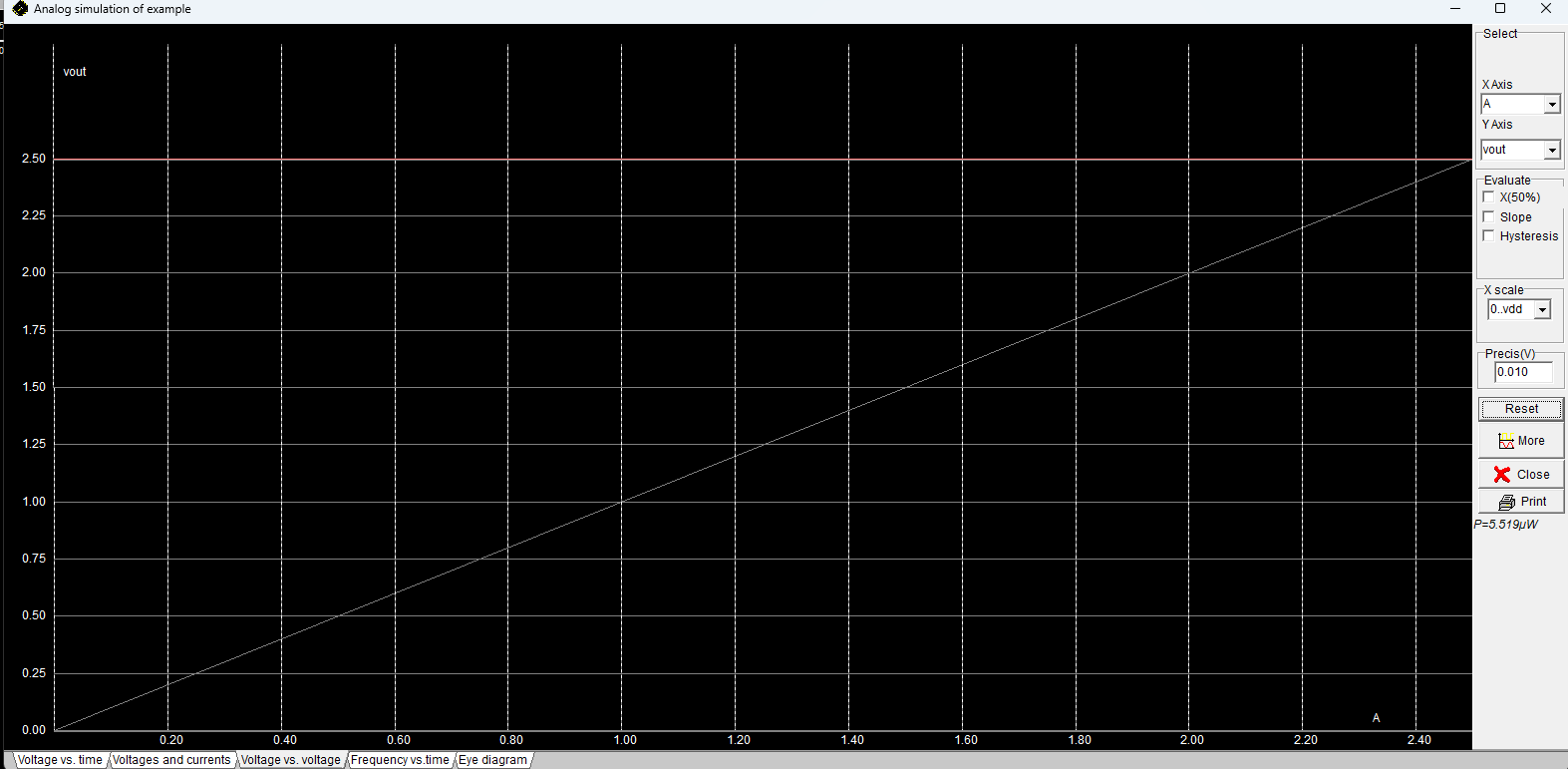
Voltage vs time -



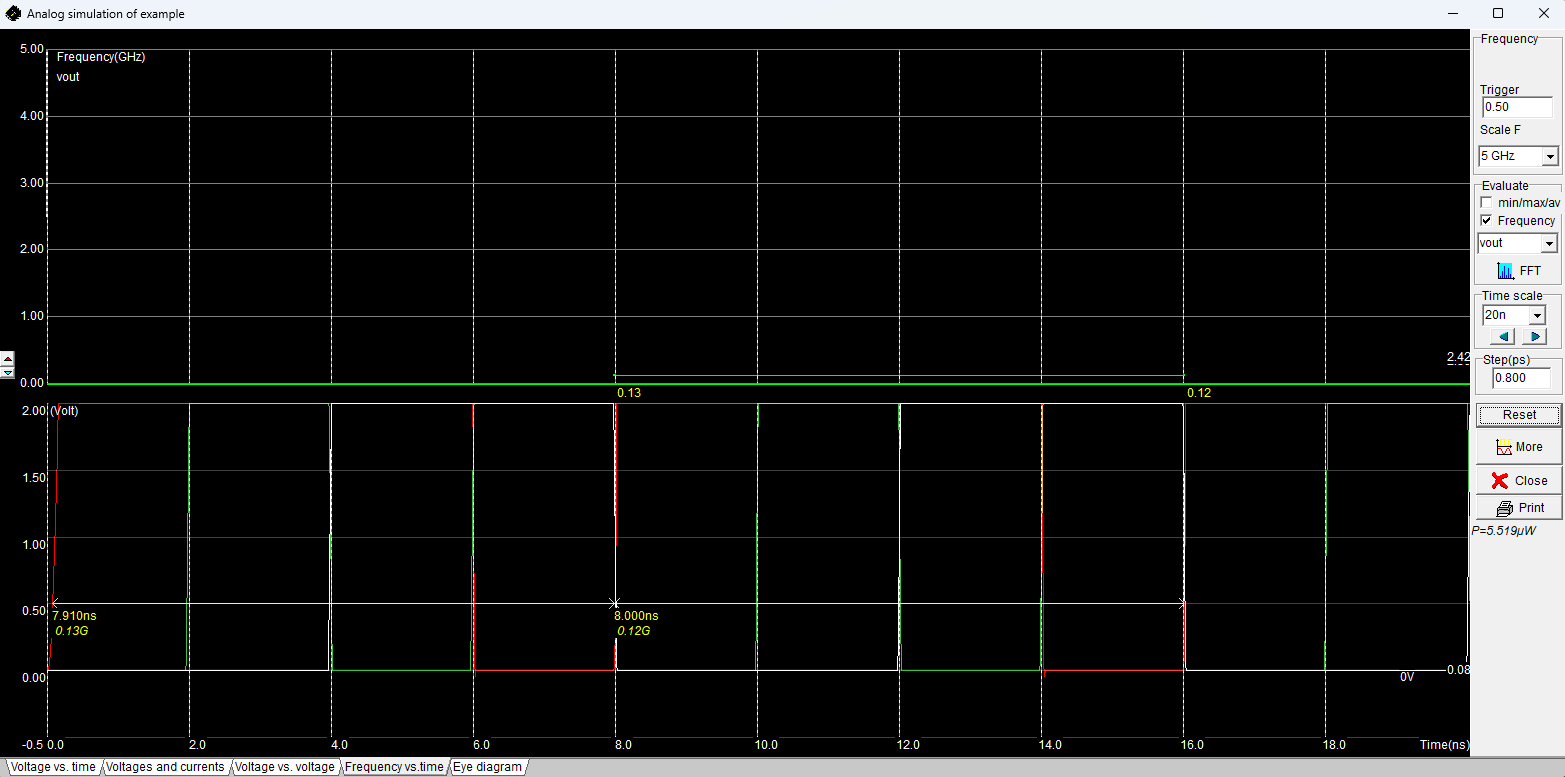
Voltages and current -



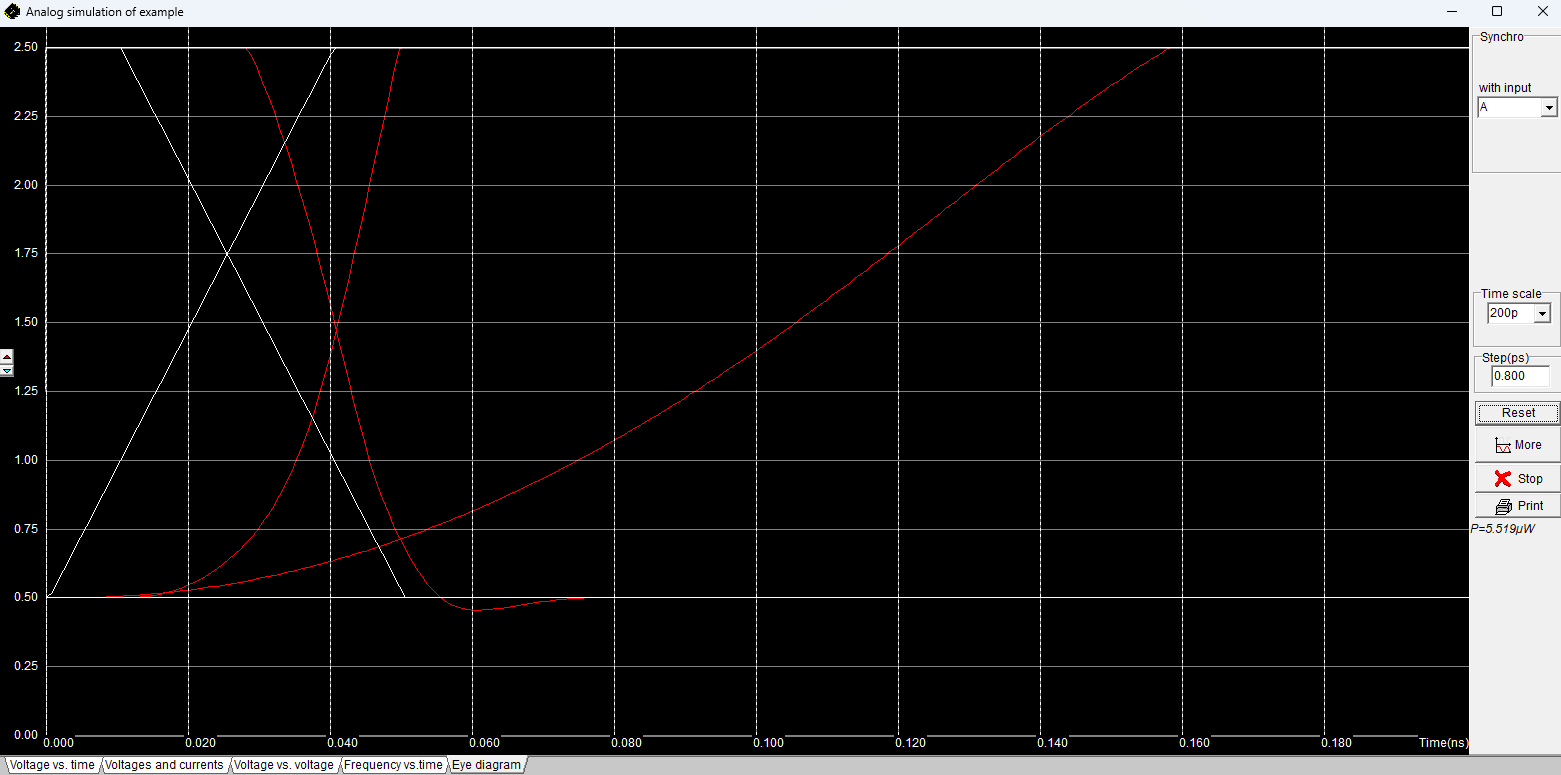
Voltage vs voltage -



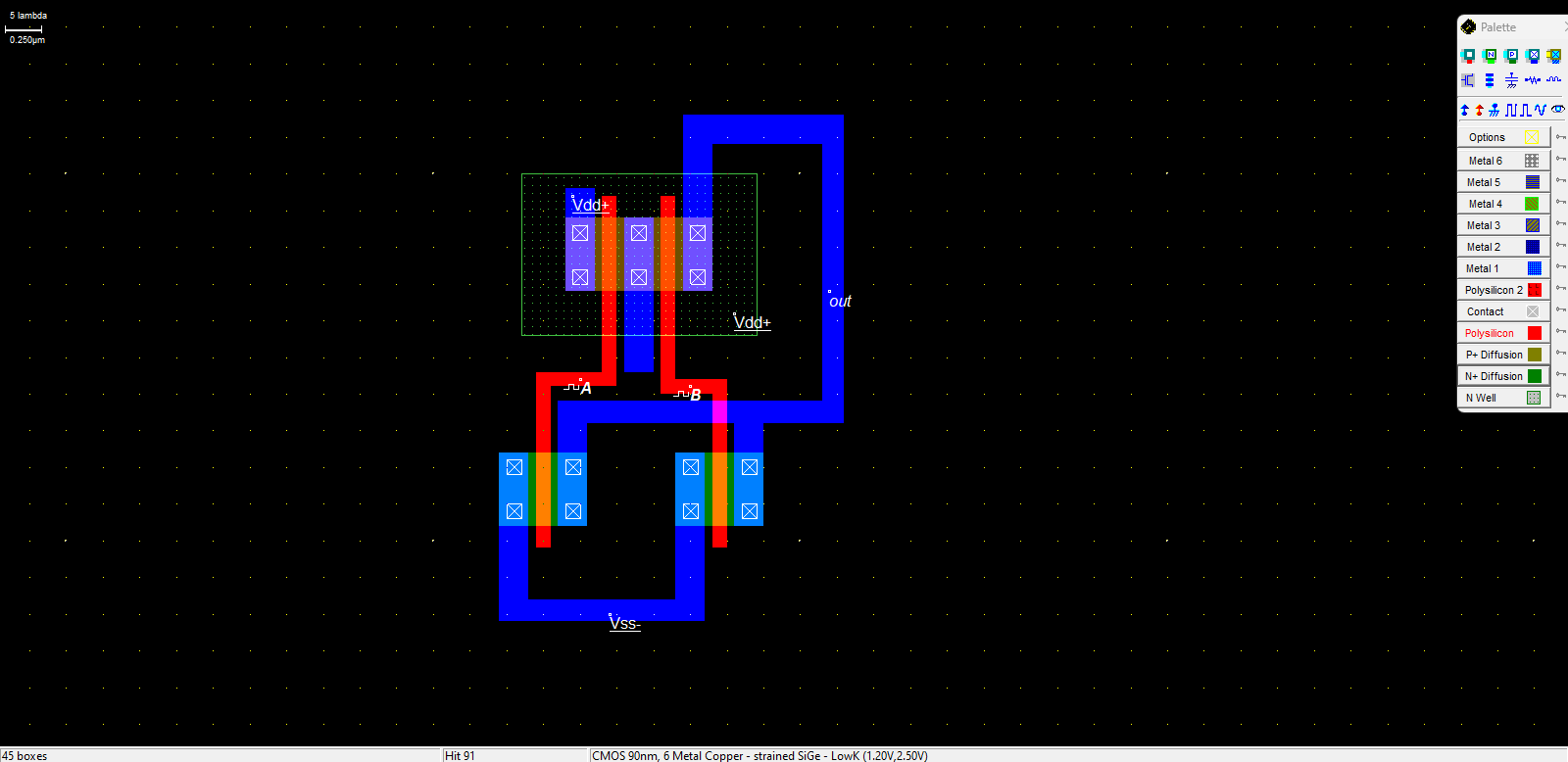
Frequency vs time -



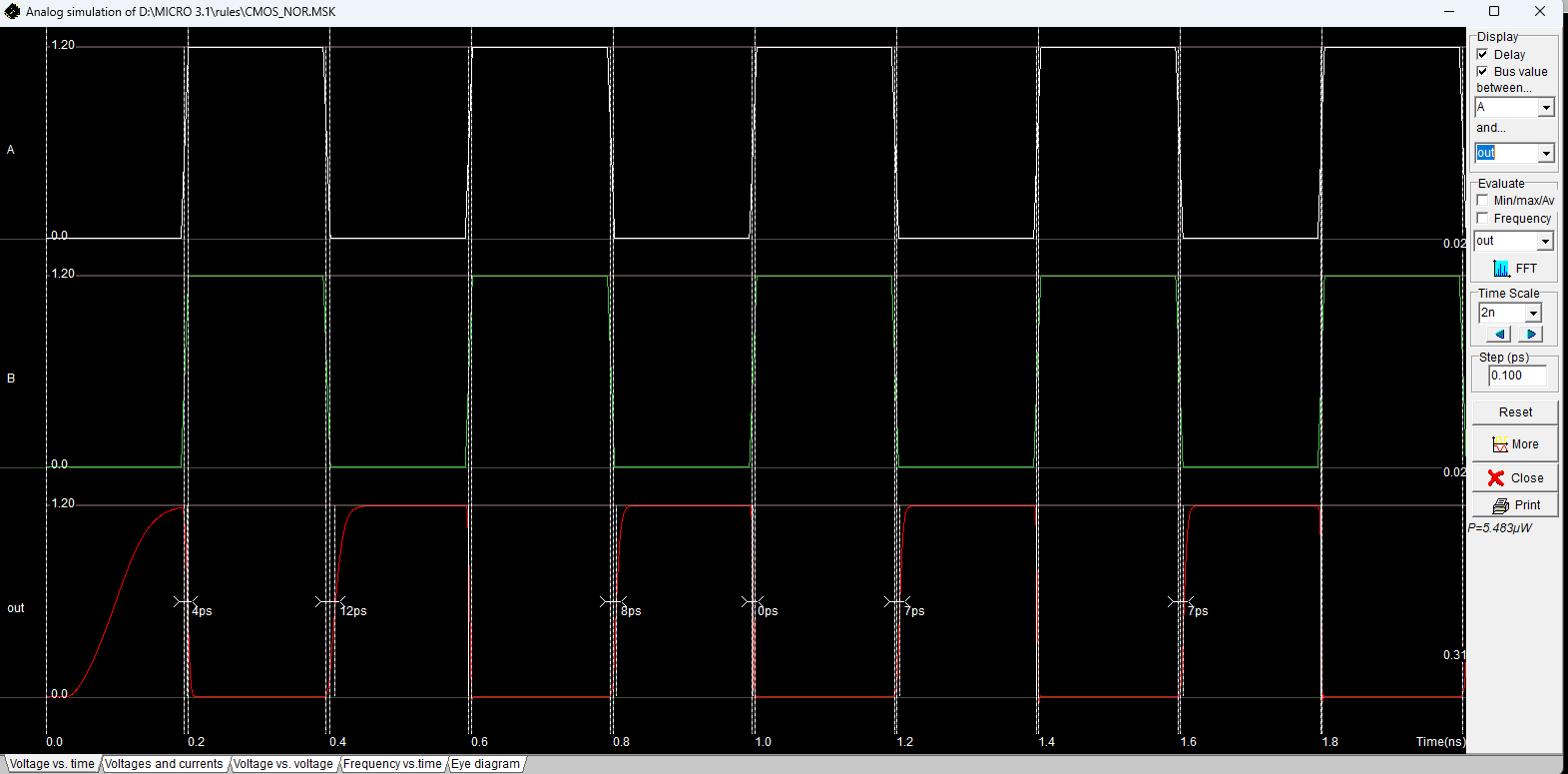
Eye diagram -



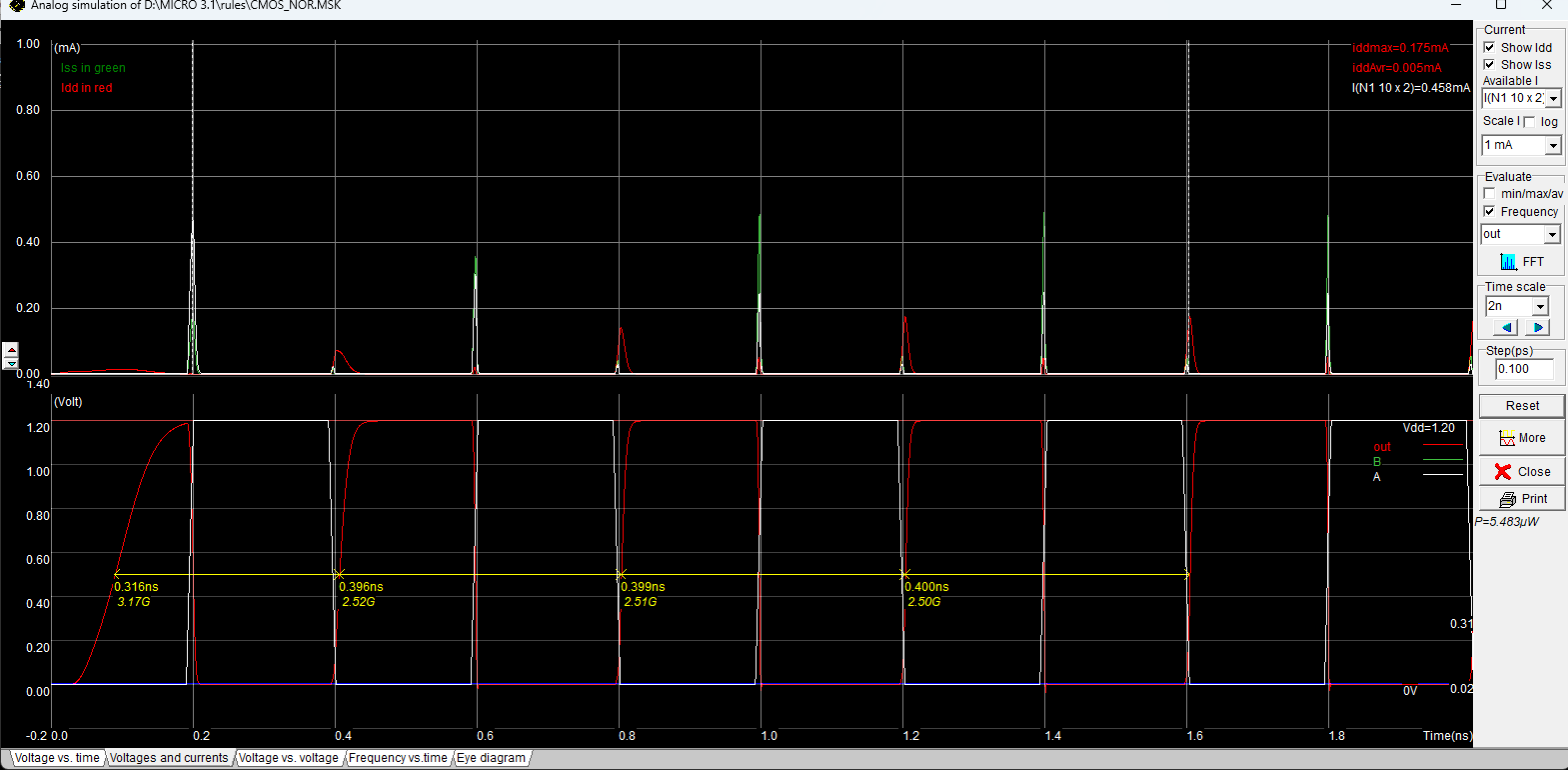
## CMOS NOR Gate :



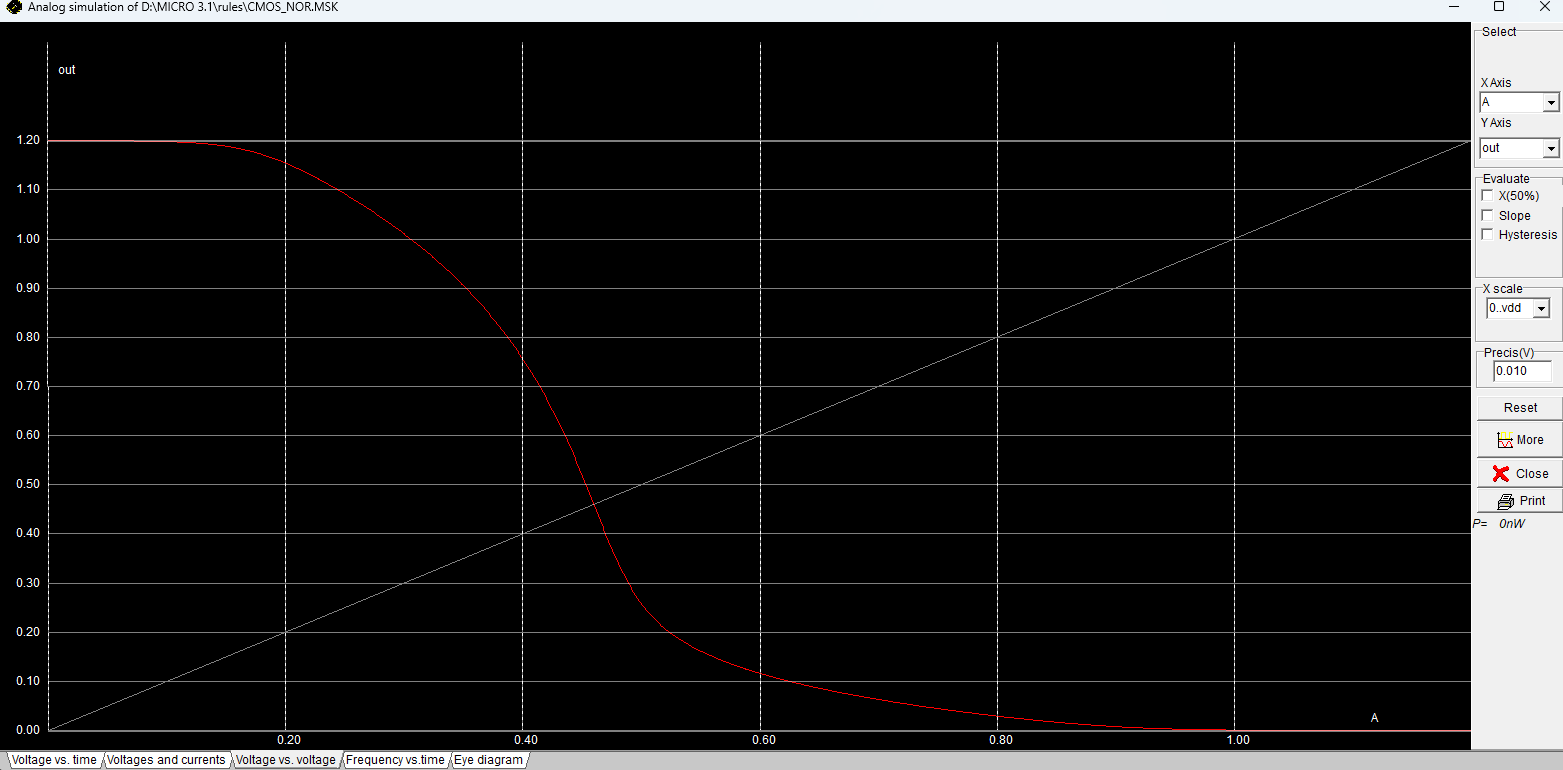
Voltage vs time –



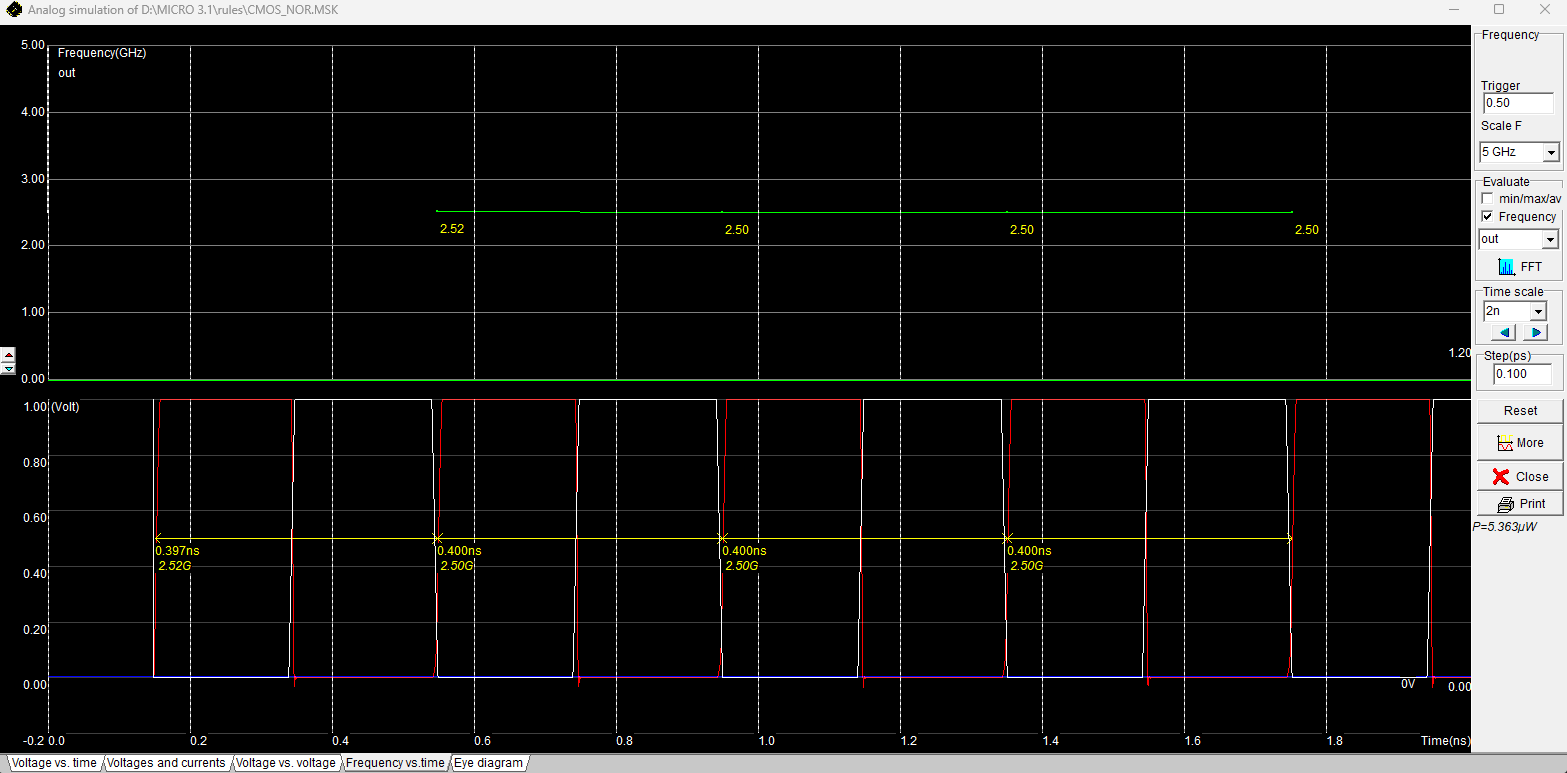
Voltage vs current –



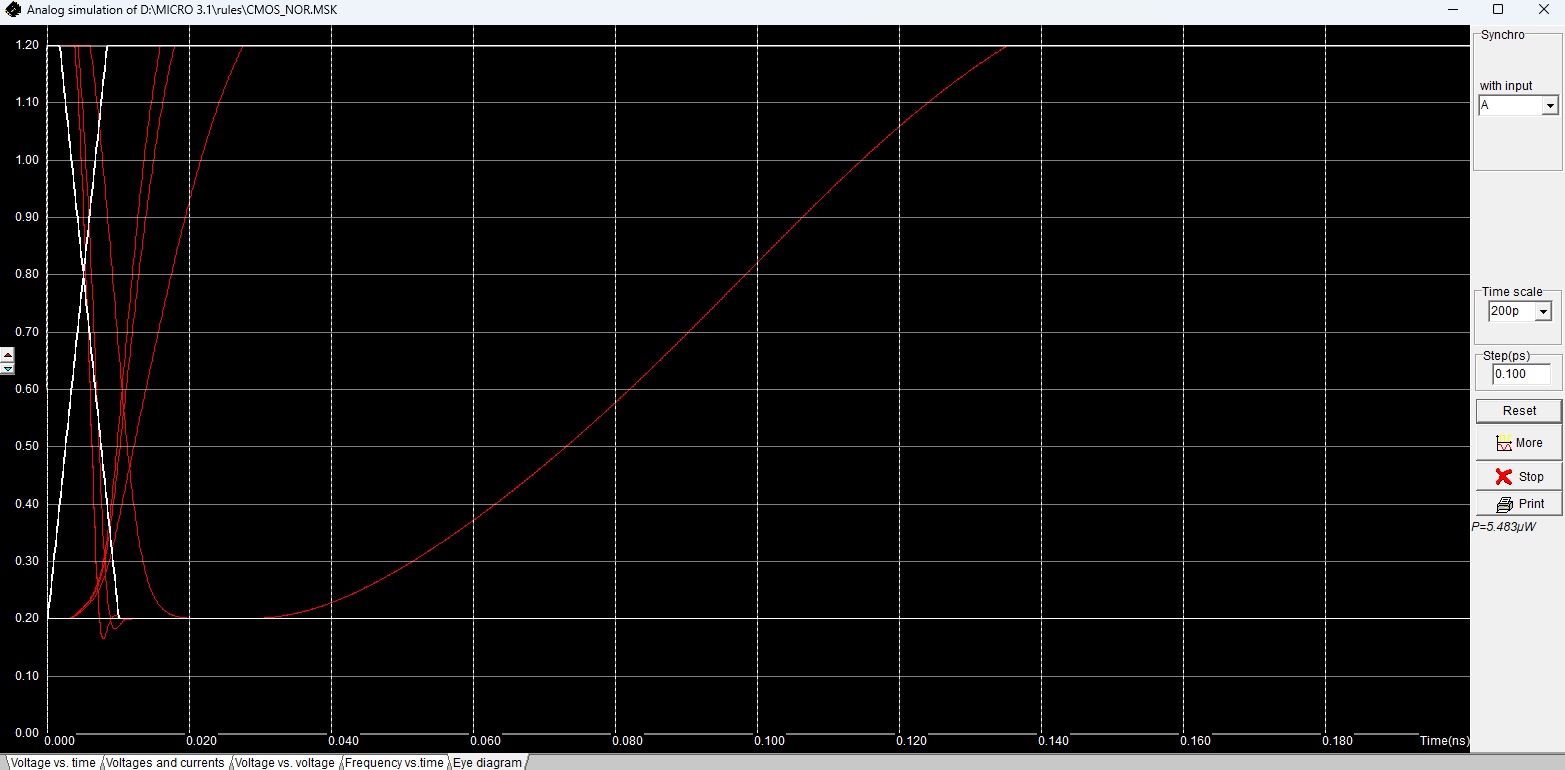
Voltage vs voltage –



Freq vs time –

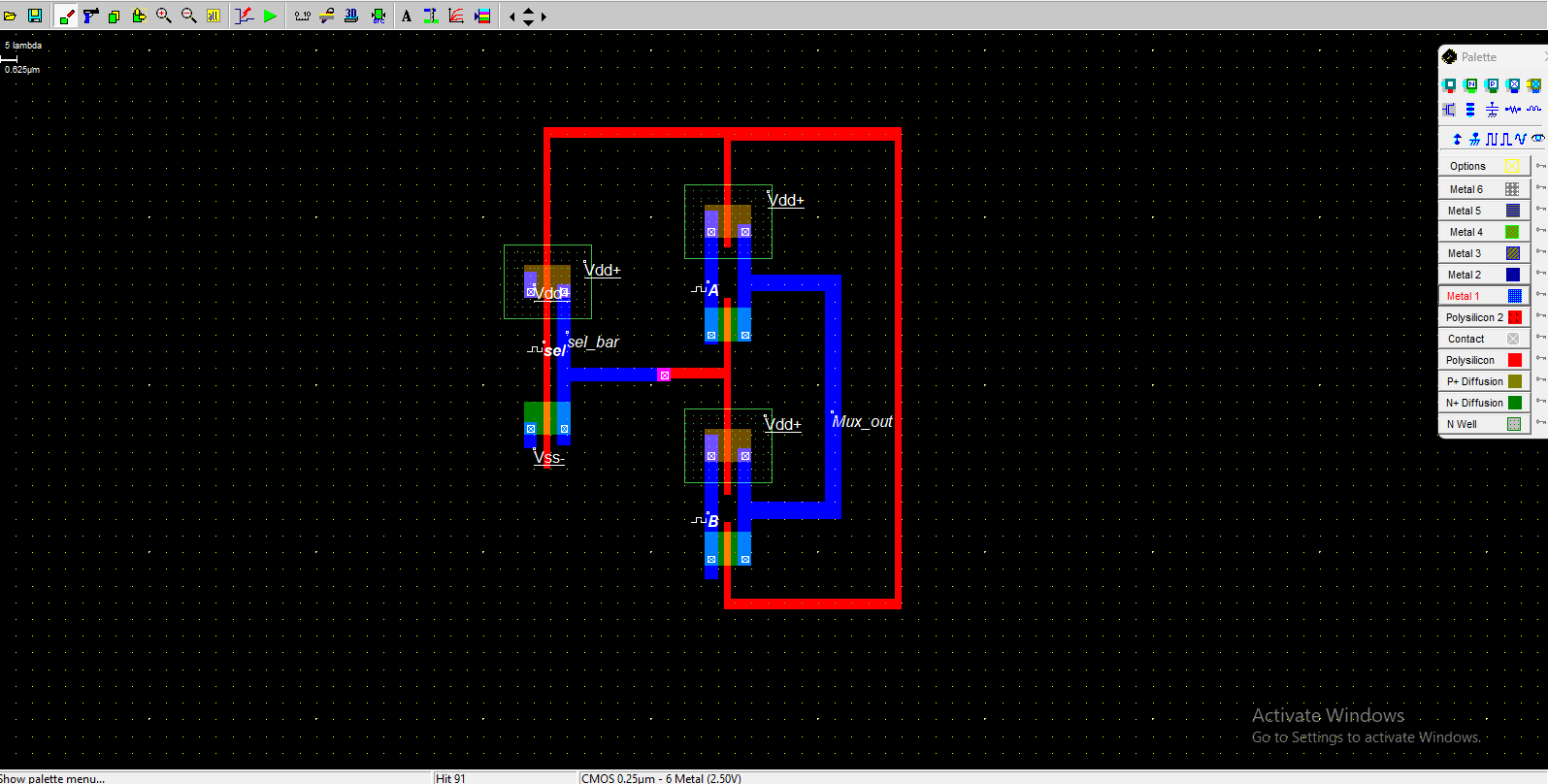


Eye diagram –

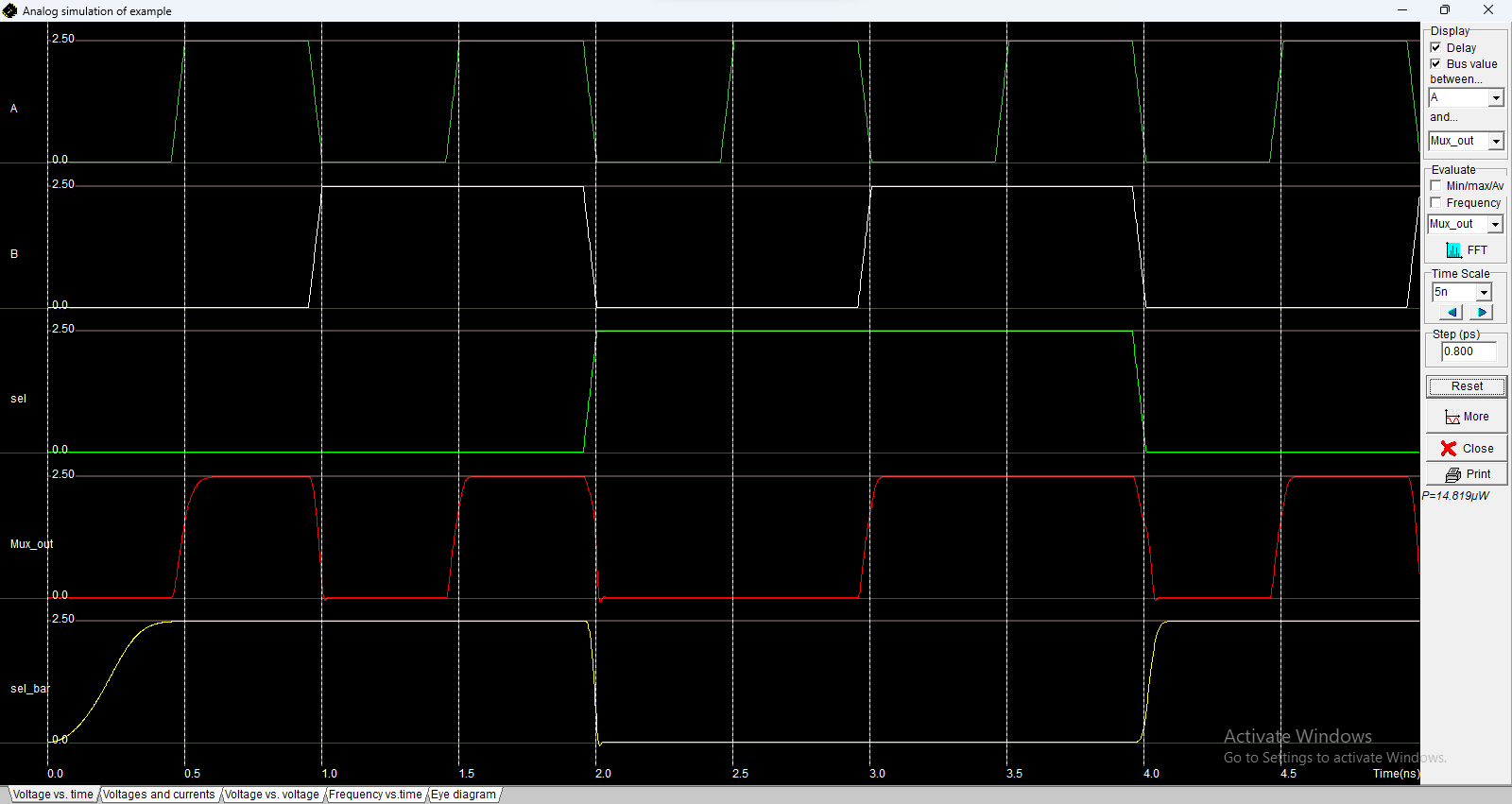


# 2:1 MUX using Transmission Gate and Logic gate :

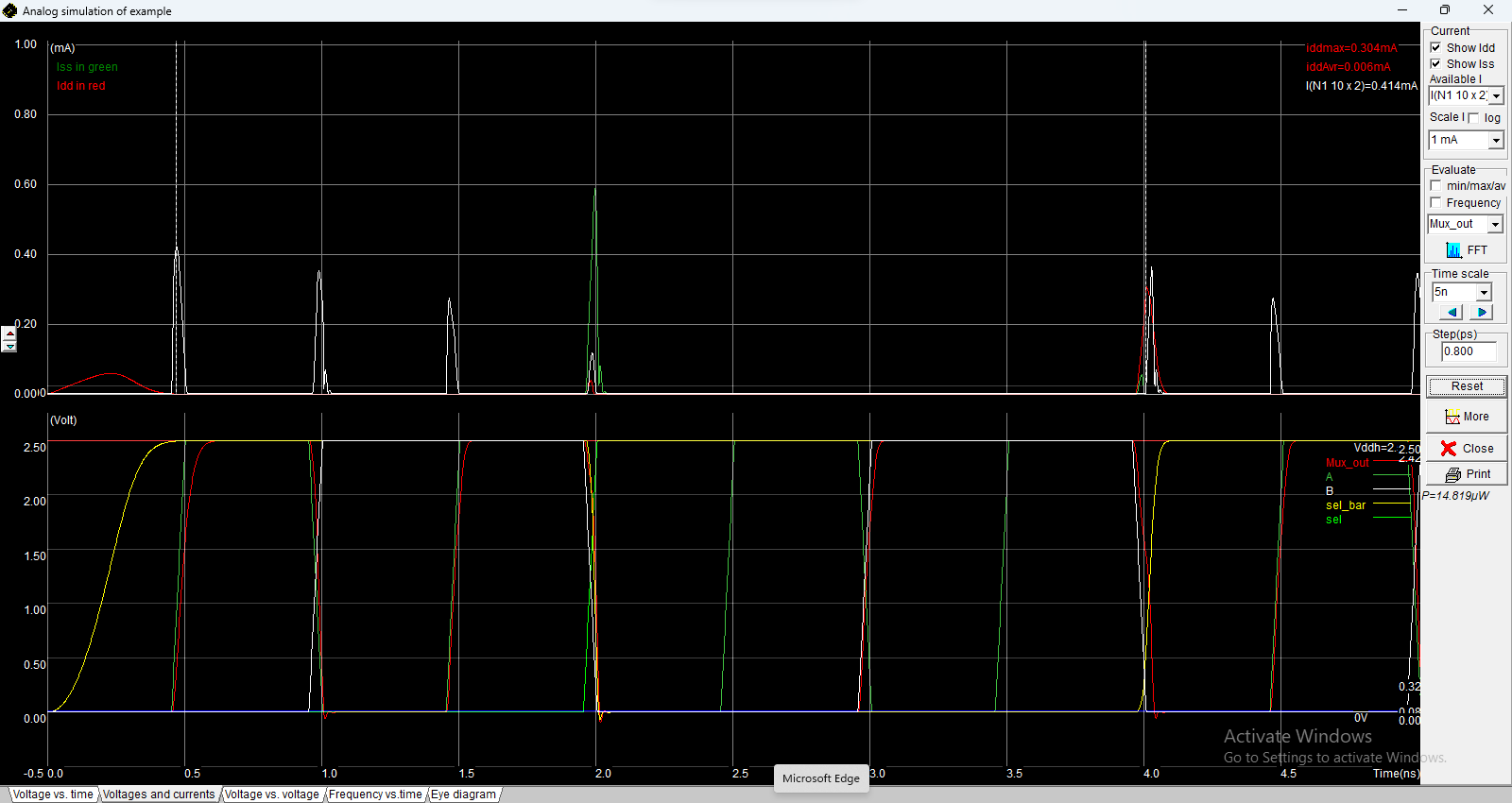
## 2:1 MUX using Transmission Gate :



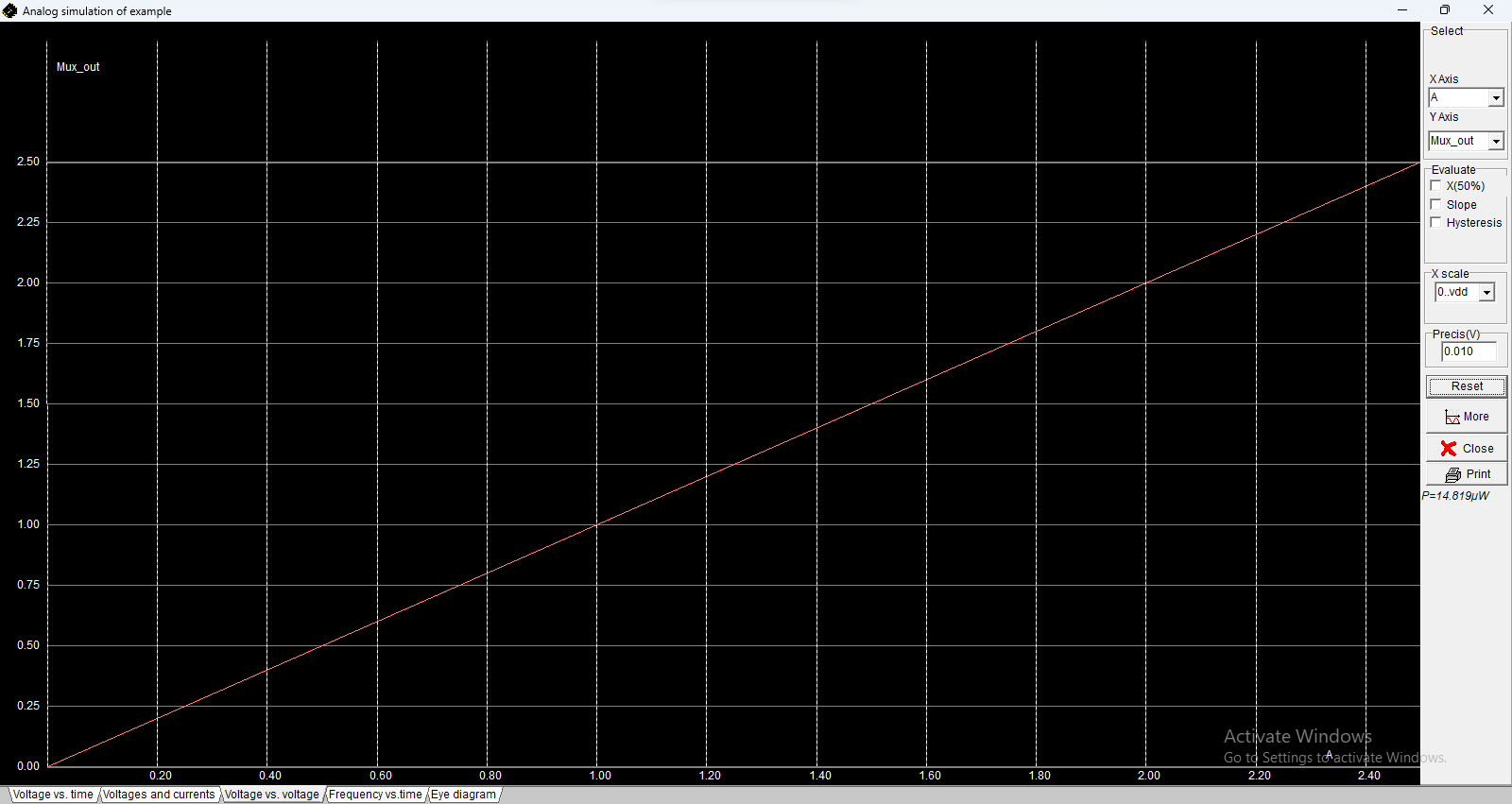
Voltage vs time –



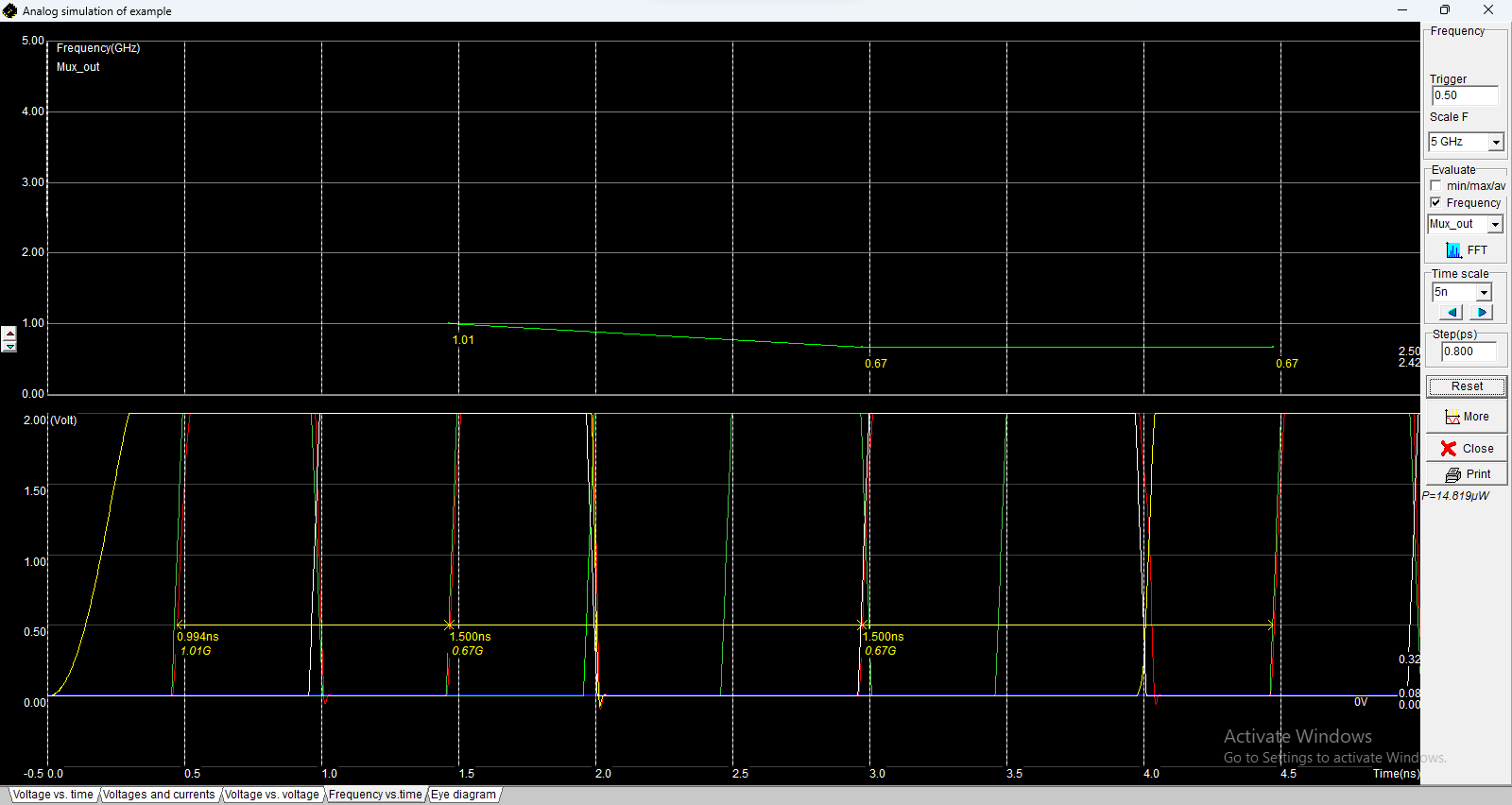
Voltage and current –



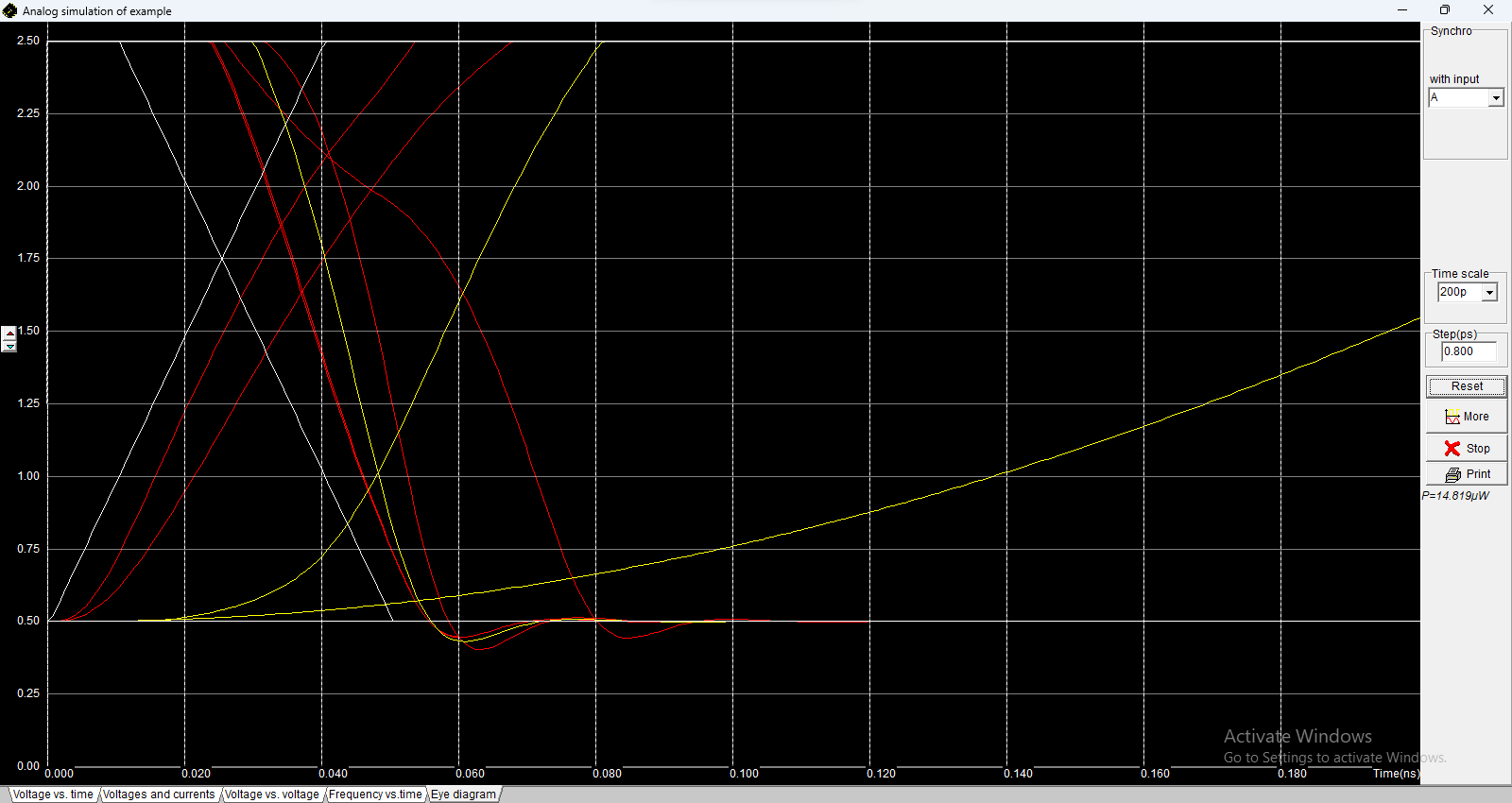
Voltage vs voltage –



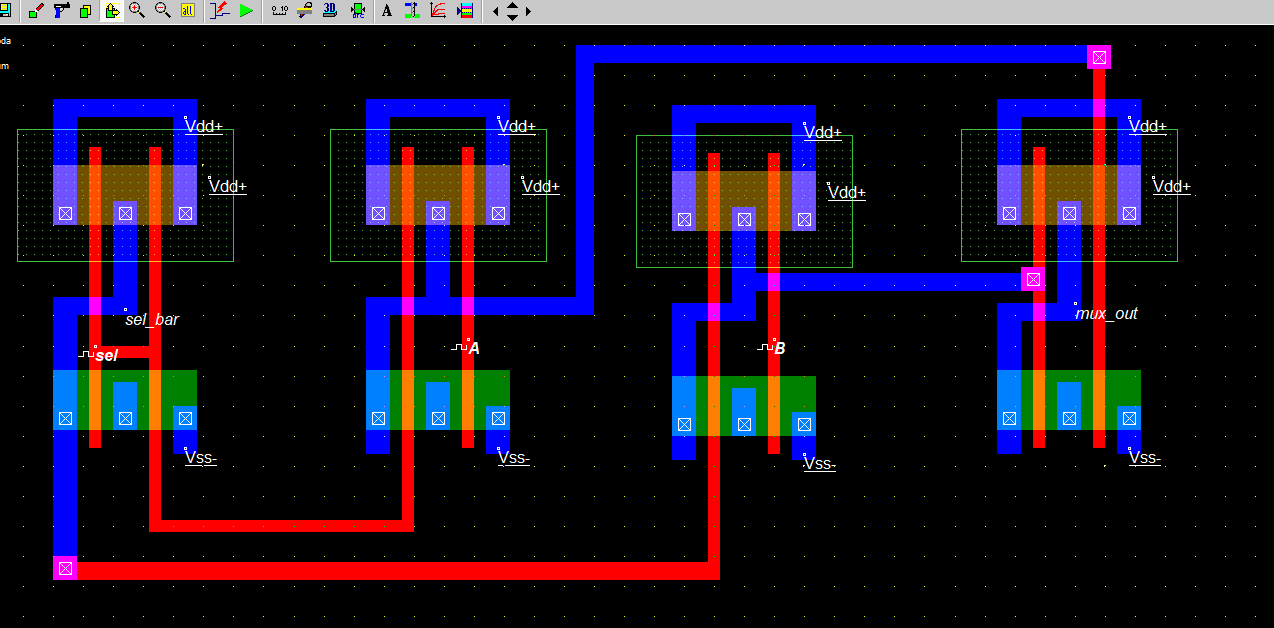
Freq vs time -



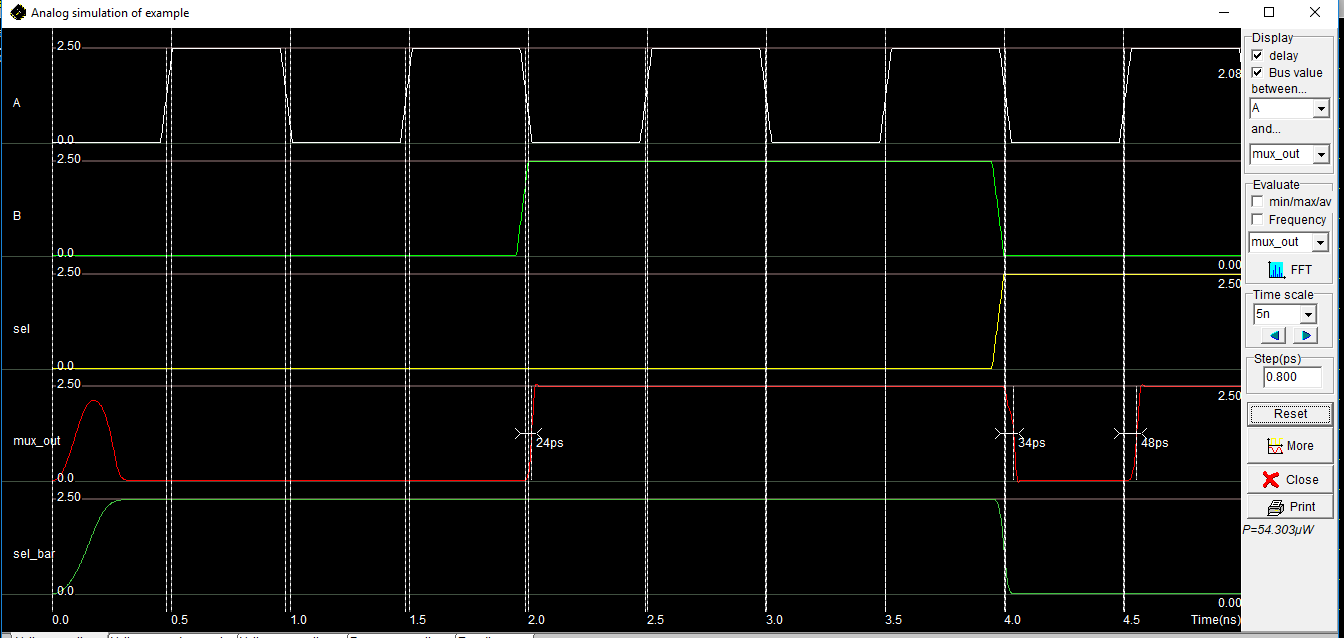
Eye diagram –



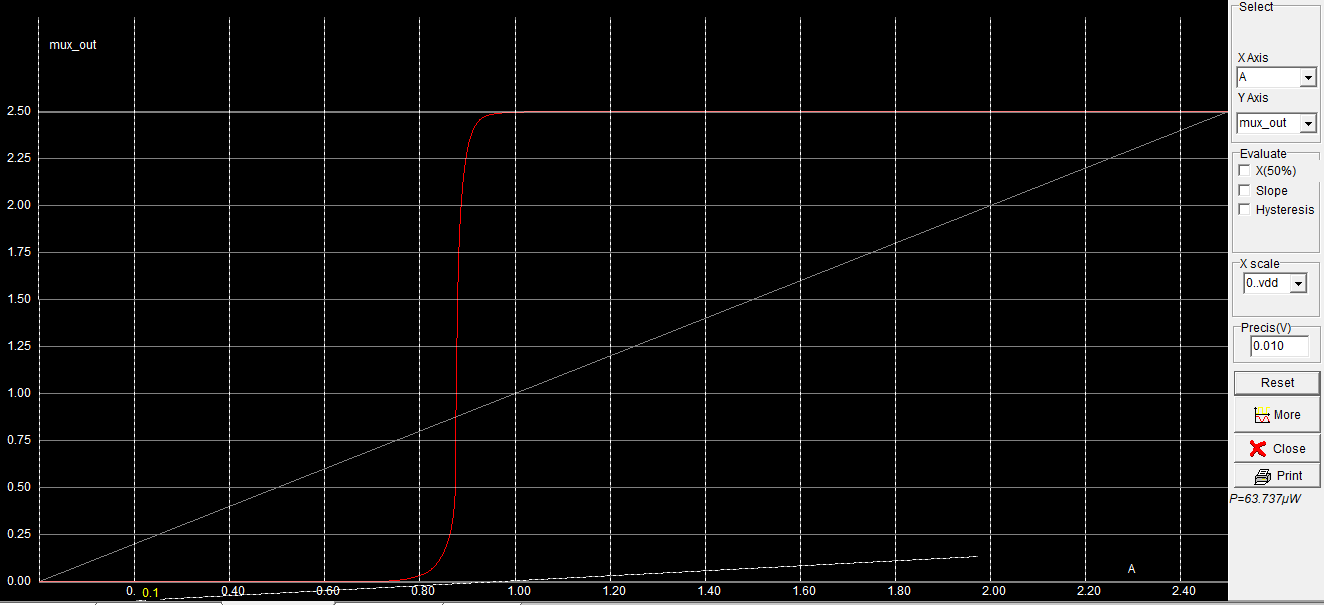
## 2:1 MUX using Logical gate (NAND) :



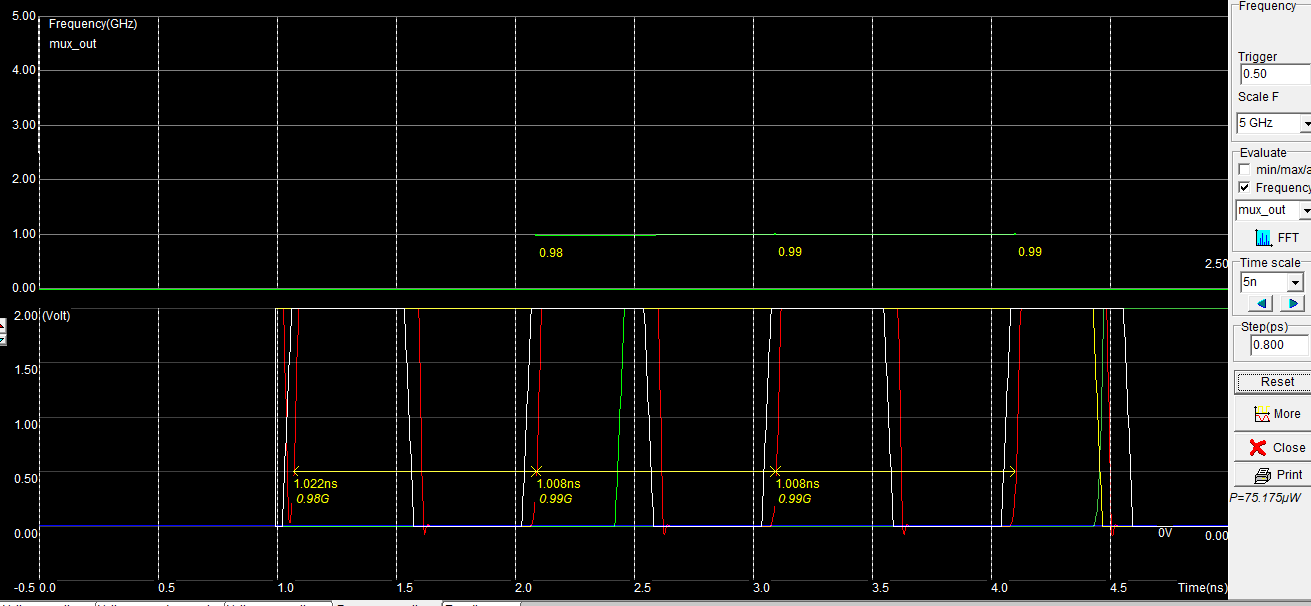
Voltage vs time –



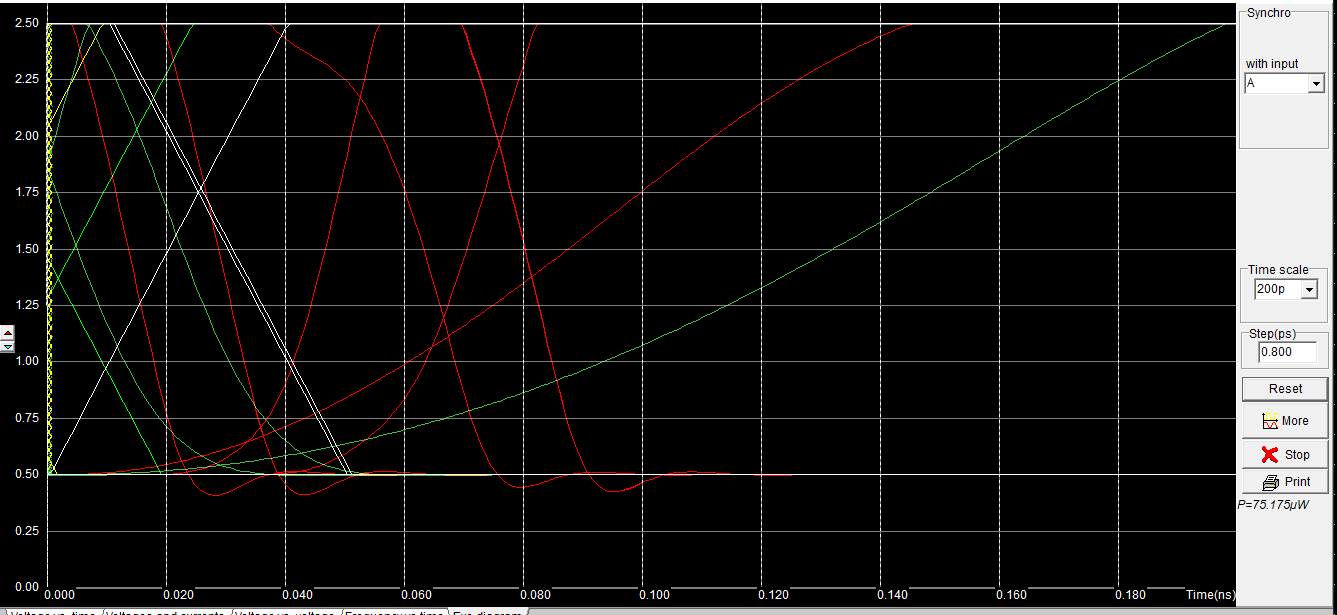
Voltage vs voltage –



Freq vs time -

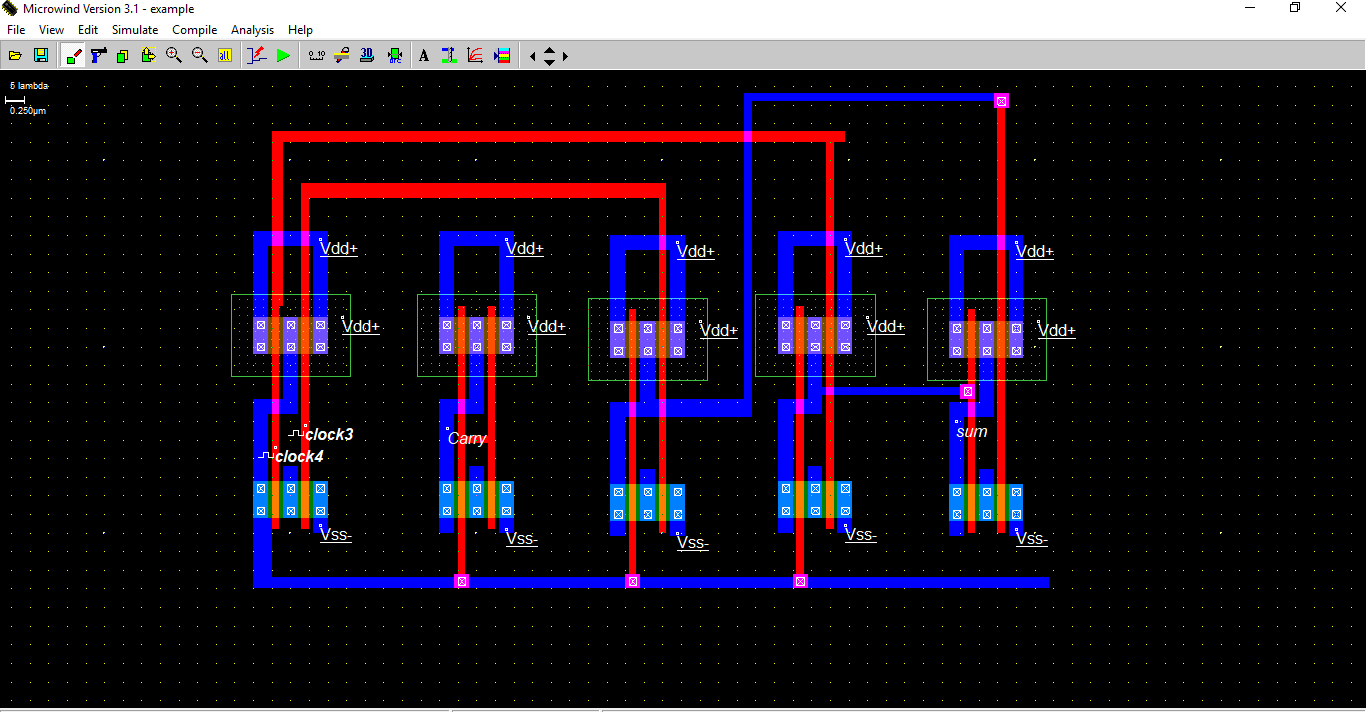


Eye diagram –

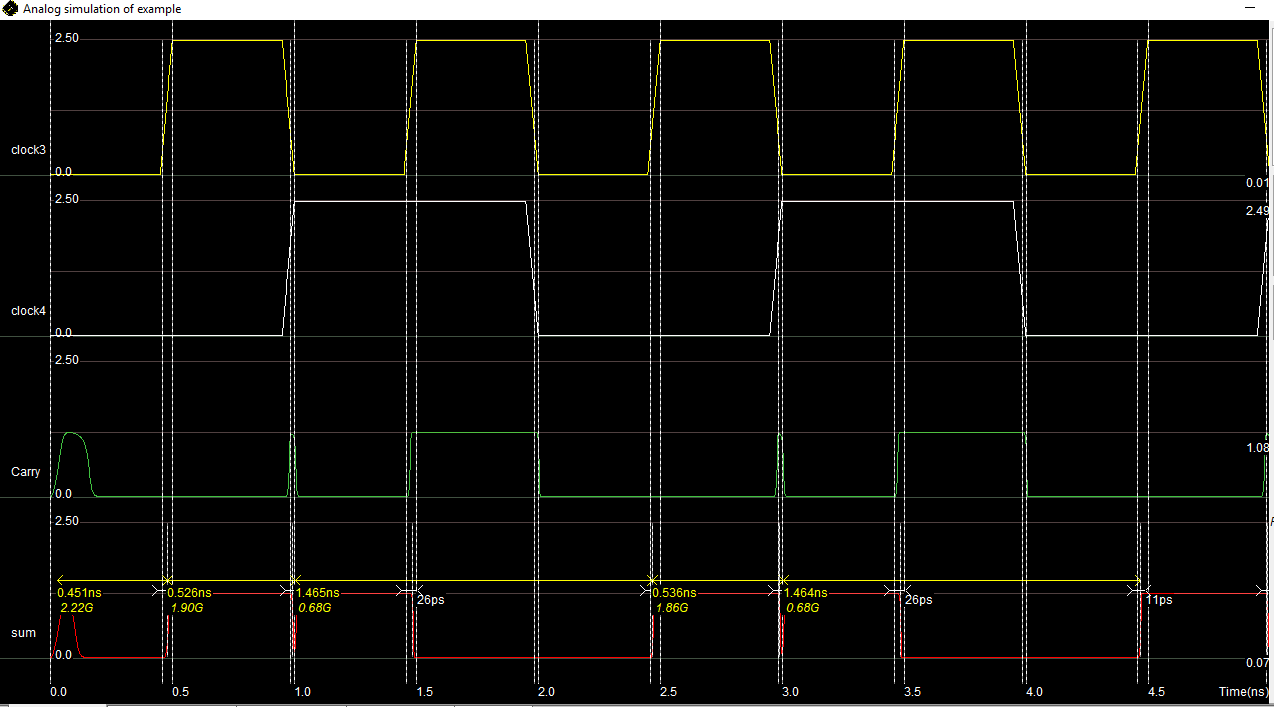


# Half Adder and Full Adder :

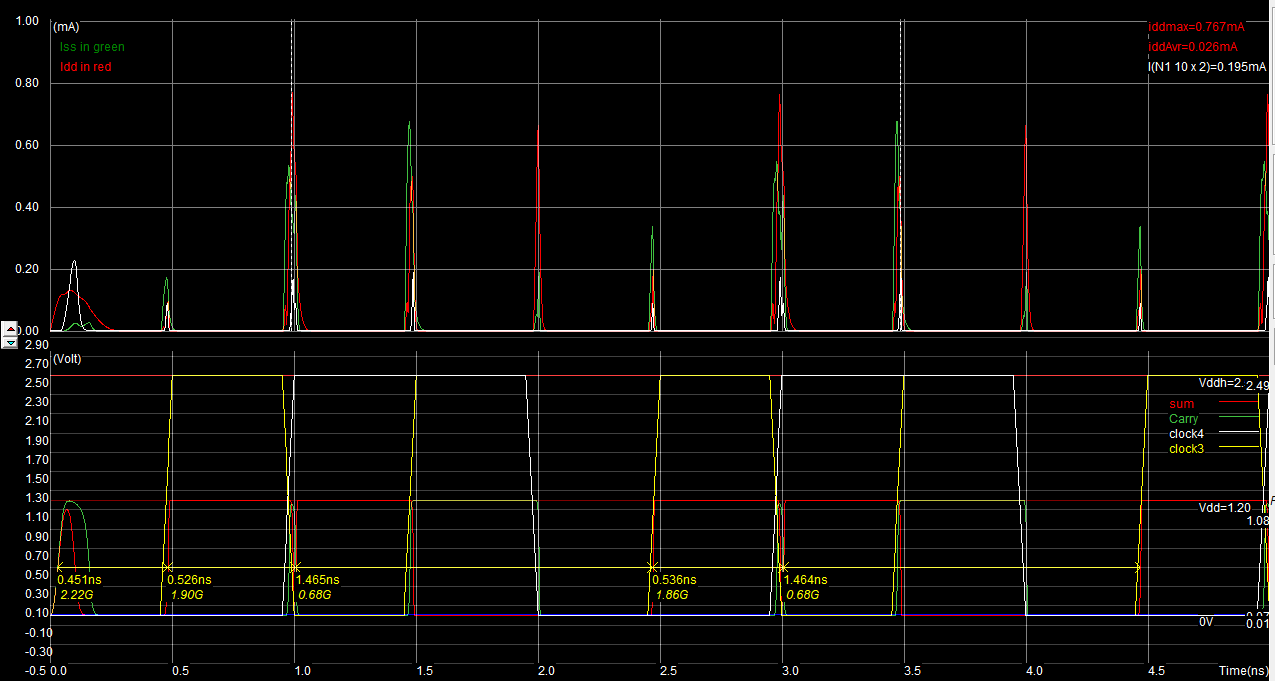
## Half Adder :



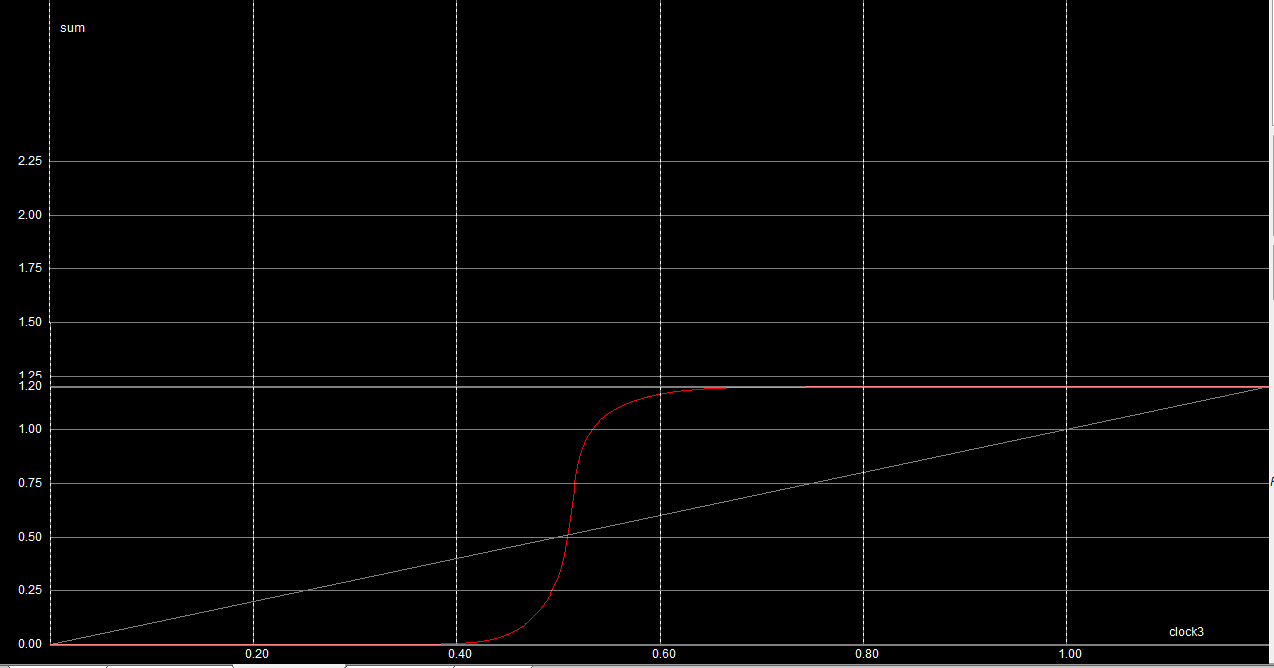
Voltage vs Time -



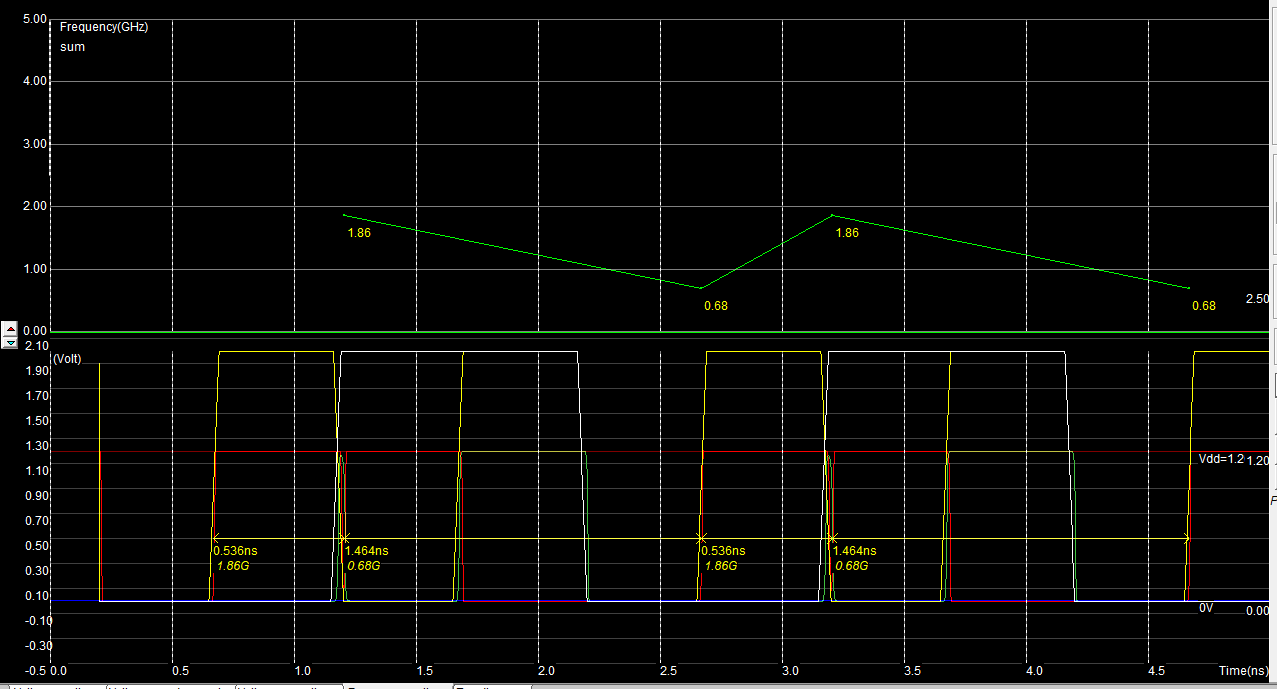
Voltages and Currents -



Voltage vs Voltage -



Frequency vs Time -



Eye Diagram -

