Name: Hao Luo

Student ID: 2737588

Lab Report no: 7

**LAB REPORT OF 8-bit pseudo-random number generator**

**OBJECTIVE:**

To design, implement, demonstrate and document an 8-bit pseudo-random number generator.

**CODE:**

**generator.vhd:**

entity generator is

Port (Clk,Rst : in std\_logic;

q : out std\_logic\_vector(7 downto 0) );

end generator;

architecture Behavioral of generator is

constant seed : std\_logic\_vector(7 downto 0) := "00000001";

signal num : std\_logic\_vector(7 downto 0):=seed;

begin

process(Clk,Rst)

begin

if Rst = '1' then

num <= seed;

elsif Rising\_edge(Clk) then

num <= (num(4) xor num(3) xor num(2) xor num(0)) & num(7 downto 1);

end if;

end process;

q<=num;

end Behavioral;

**gen\_tb.vhd:**

architecture Behavioral of gen\_tb is

component generator is

port ( Clk,Rst : in std\_logic;

q : out std\_logic\_vector(7 downto 0) );

end component;

signal Clk :std\_logic;

signal Rst :std\_logic;

--signal Load :std\_logic;

signal q : std\_logic\_vector(7 downto 0);

CONSTANT clk\_period : time := 10 ns;

begin

mapping : generator port map(Clk=>Clk, Rst=>Rst, q=>q);

stim\_proc: process

begin

Clk <= '1';

Rst <= '0';

--Load <= '1';

wait for clk\_period\*5;

Clk <= '0';

--Load <= '0';

wait for clk\_period\*5;

Clk <= '1';

wait for clk\_period\*5;

Clk <= '0';

wait for clk\_period\*5;

Clk <= '1';

wait for clk\_period\*5;

Clk <= '0';

wait for clk\_period\*5;

Clk <= '1';

wait for clk\_period\*5;

Clk <= '0';

wait for clk\_period\*5;

Clk <= '1';

wait for clk\_period\*5;

Clk <= '0';

wait for clk\_period\*5;

Clk <= '1';

wait for clk\_period\*5;

Clk <= '0';

wait for clk\_period\*5;

Clk <= '1';

wait for clk\_period\*5;

wait for clk\_period\*5;

end process;

end Behavioral;

**OPERATION:**

The main operation of an 8-bit pseudo-random number generator is algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. For our pseudo-random number generator, it loads 8 bits parallel data, using the ( q(4) , q(3) , q(2) , q(0) ) signals from the base number(seed) of the register are xored to generate a new value, then right shift 1 bit for the base number(seed), and use the new generated number to replace the missed 8 bit in the final result. This operation will be called recursively until it reaches the end of the period.

**RESULT:**

After finishing my testbench, I run my VHDL program and it generates a waveform as below