**Lab 4:**

**Noise-generator with sequential VHDL**

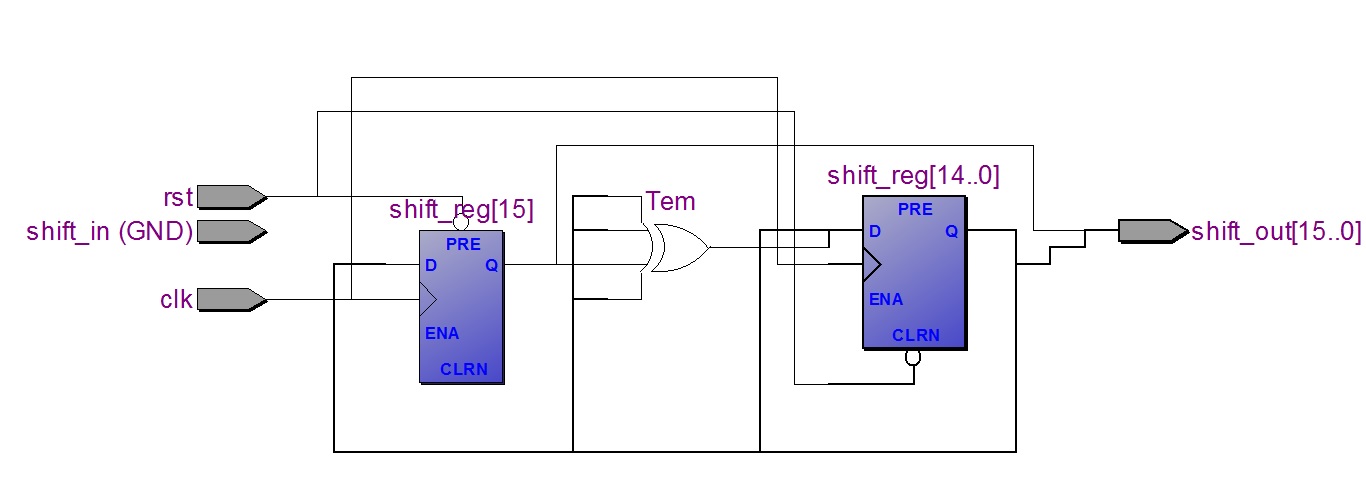
**INTRODUCTION:**

In this we have designed a 16-bit Linear Feedback Shift Register (LFSR) to generate a random noise signal.

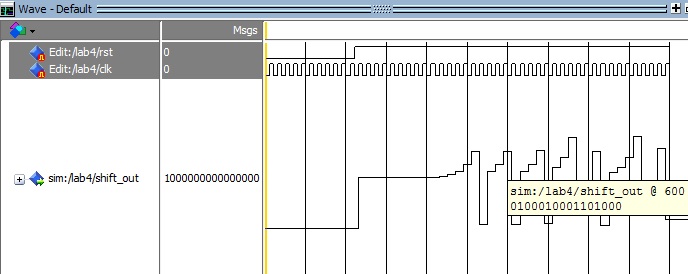
We have taken two signals, Reset (rst) and Clock (clk) as an input signals. The operation is done on the rising edge of the clock. We will give a high frequency clock signal to witness a random noise wave.

The LFSR consists of an xor-gate feedback for the binary addition without a carry.

**RTL VIEW:**

****

**WAVEFORM:**

****

**CODE:**

Library IEEE;

USE IEEE.std\_logic\_1164.all;

Entity lab4 IS

PORT( rst, clk : IN std\_logic;

shift\_in : IN std\_logic;

shift\_out: OUT std\_logic\_vector(15 downto 0)

);

END lab4;

--15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

--rst clk Figure: Linear feedback shit-register (LFSR)

ARCHITECTURE behavior OF lab4 IS

signal shift\_reg: std\_logic\_vector(15 downto 0);

Signal Tem: std\_logic;

BEGIN

Tem <= shift\_reg(15) xor shift\_reg(14) xor shift\_reg(12) xor shift\_reg(3);

Process (clk,rst)

BEGIN

IF rst='0' THEN

shift\_reg<=(X"8000");

ELSIF clk'event and clk='1' THEN

shift\_reg(15 downto 1)<=shift\_reg(14 downto 0);

shift\_reg(0)<= Tem;

END IF;

END Process;

shift\_out<=shift\_reg;

End behavior;

**Conclusion:**

As a result, we will get an output signal which looks like analog. In other words we can say that this technique is an example of Digital to Analog Converter (DAC) as we have given two digital signals (rst, clk) as input and we got an analog signal which looks like a random noise.