ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS
UNIVERSITÉ HASSAN II DE CASABLANCA

[MSEI & GEM]

Encryption project implemented in Quartus

# LINEAR FEEDBACK SHIFT REGISTER


The group members :

- SELKAOUI Jihane
- HALOUI Hadil
- OULASRI Meriam

supervised by :

S.EL MOUMNI


2023/2024

# Introduction:

**VHDL** (VHSIC Hardware Description Language) is a hardware description language that can model the behavior and structure of digital systems at multiple levels of abstraction, ranging from the system level down to that of logic gates, for design entry, documentation, and verification purposes.

Developed for the US military VHSIC program in the 1980s, VHDL is defined by IEEE ( Institute of Electrical and Electronics Engineers) standards; there are two common variants: VHDL-1987 and VHDL-1993. VHDL can be used for designing hardware and for creating test entities to verify the behavior of that hardware also it is used as a design entry format by a variety of EDA tools, including synthesis tools such as Quartus® Prime Integrated Synthesis, simulation tools, and formal verification tools.

# RNGs and LFSRs:

In the digital age, secure communication is paramount. Encryption and decryption processes form the backbone of data security, safeguarding information from unauthorized access and breaches ,central to these processes is the generation of random numbers, which play a crucial role in the encryption algorithms and LFSRs (Linear Feedback Shift Registers) are widely recognized for their efficiency in generating pseudo-random number sequences, which are integral to cryptographic protocols.

So in this project we will explore the application of LFSRs (Linear Feedback Shift Registers) within a cryptographic protocol for encryption and decryption , so our objective is to generate random numbers using LFSR.

### a) Random Number Generators:

**Random numbers** are useful for a variety of purposes, such as generating data encryption keys, simulating and modeling complex phenomena and for selecting random samples from larger data sets a random number is one that is drawn from a set of possible values, each of which is equally probable, i.e., a uniform distribution.

Random numbers are generated by various methods. The two types of generators used for random number generation are pseudo random number generator (PRNG) and true random number generator (TRNG)and its generated through three techniques i.e., linear feedback shift register, linear congruential generator and blum blum shub.

### b) linear feedback shift register:

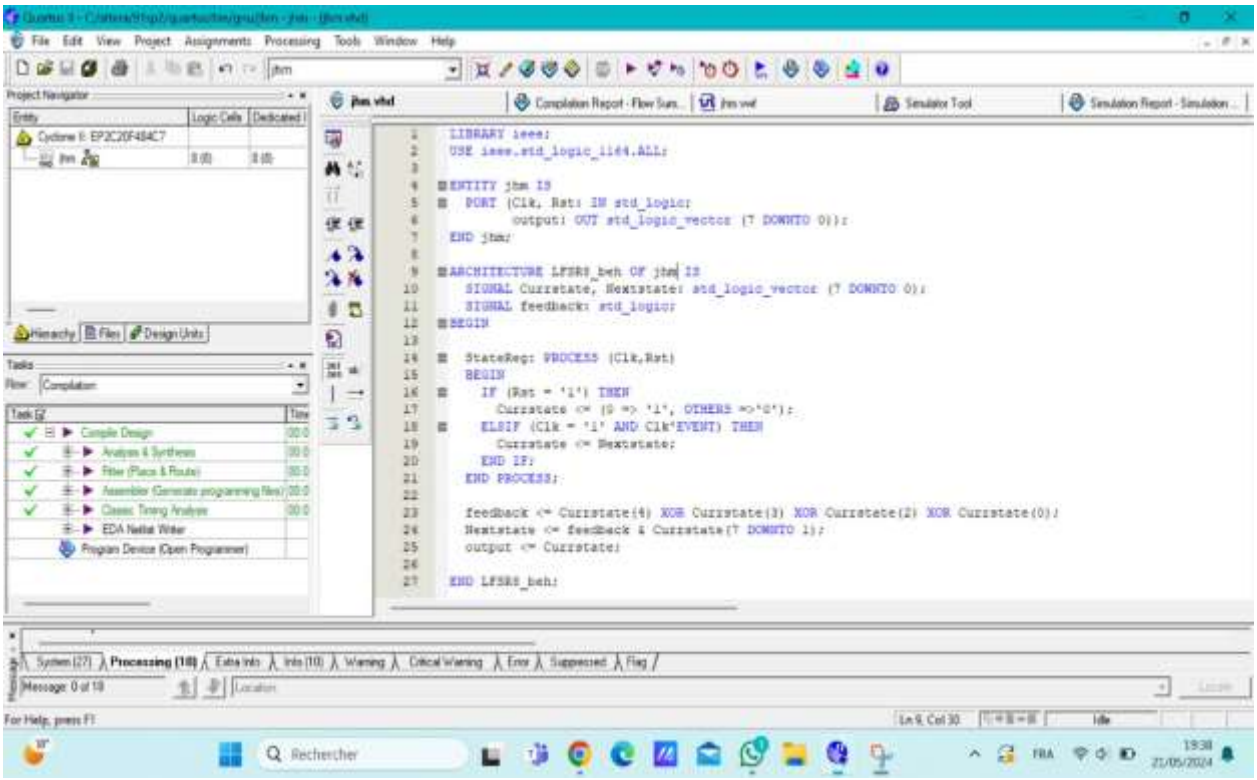
**A linear feedback shift register (LFSR)** is a shift register whose input bit is a linear function of its previous state. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined 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 which appears random and which has a very long cycle.

Applications of LFSRs include generating pseudorandom numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementations of LFSRs are common.

- VHDL program

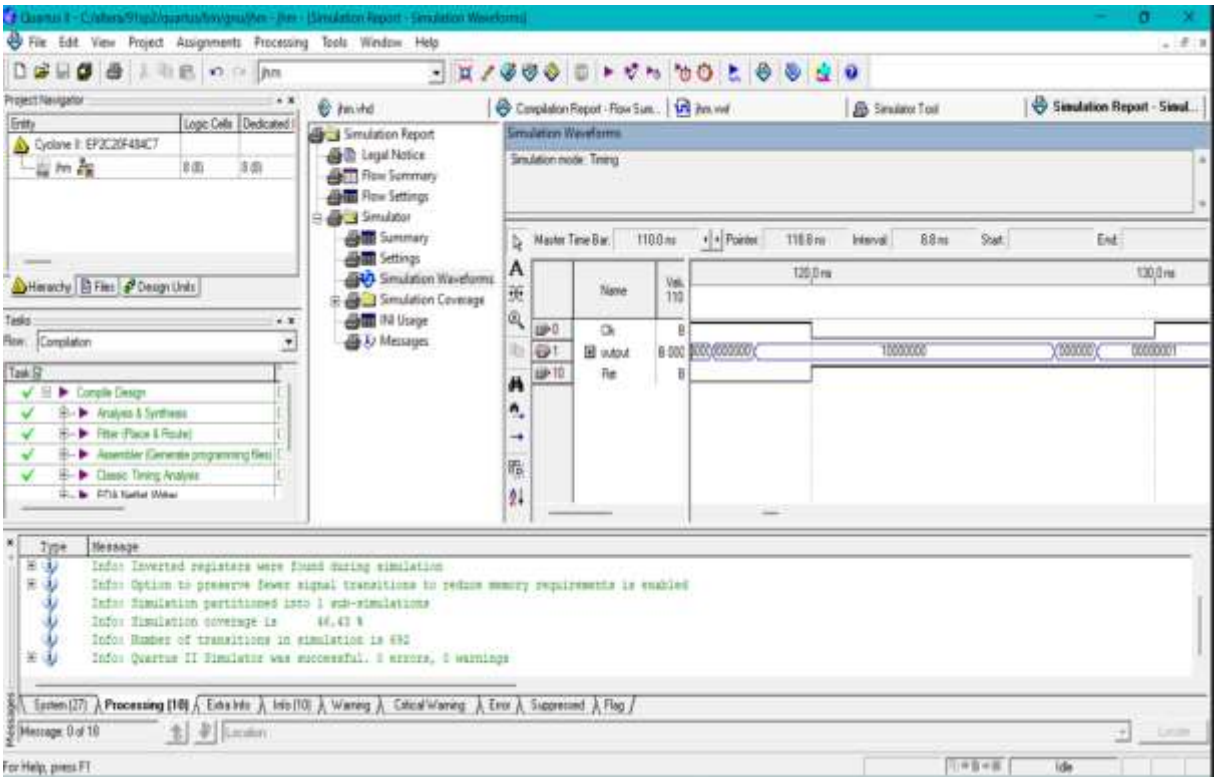
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY LFSR8 IS
 PORT (Clk, Rst: IN std_logic;
      output: OUT std_logic_vector (7 DOWNTO 0));
END LFSR8;

ARCHITECTURE LFSR8_beh OF LFSR8 IS
 SIGNAL Currstate, Nextstate: std_logic_vector (7 DOWNTO 0);
 SIGNAL feedback: std_logic;
BEGIN

 StateReg: PROCESS (Clk,Rst)
 BEGIN
  IF (Rst = '1') THEN
    Currstate <= (0 => '1', OTHERS =>'0');
  ELSIF (Clk = '1' AND Clk'EVENT) THEN
    Currstate <= Nextstate;
  END IF;
```

```
END PROCESS;

 feedback <= Currstate(4) XOR Currstate(3) XOR Currstate(2)
XOR Currstate(0);
 Nextstate <= feedback & Currstate(7 DOWNTO 1);
 output <= Currstate;

END LFSR8_beh;
```




This VHDL code implements an 8-bit Linear Feedback Shift Register (LFSR) using the Galois configuration. The LFSR has two inputs, Clk (clock) and Rst (reset), and produces an 8-bit output. It consists of two main signals, Currstate and Nextstate, which hold the current and next state of the register, respectively. The process block updates the current state (Currstate) on the rising edge of the clock, unless the reset signal is active, in which case Currstate is initialized to a predefined state with the first bit set to 1 and the rest to 0. The feedback bit is calculated by XORing bits 4, 3, 2, and 0 of the current state. This feedback bit is then concatenated with the upper 7 bits of the current state, shifted right by one position, to form the next state (Nextstate). The current state is continuously assigned to the output. This configuration generates a pseudo-random sequence of states based on the initial state and the feedback taps.

## ❖ simulation



when the reset signal (Rst) is asserted high, the Currstate will be set to the predefined initial state where the first bit is '1' and the remaining bits are '0' (i.e., 00000001). Upon deasserting the reset (Rst set to '0') and providing a clock signal (Clk), the LFSR begins to shift its state on each rising edge of the clock. The feedback bit, calculated as the XOR of bits 4, 3, 2, and 0 of the current state, is inserted into the leftmost position of the next state, while the other bits are shifted right. The simulation waveform will display the output reflecting the current state (Currstate), showcasing a sequence of pseudo-random binary values that evolve with each clock cycle. This sequence continues until the LFSR returns to its initial state, demonstrating the cyclic nature of the register's behavior. Each transition of the LFSR state will be visible in the simulation results, confirming the correct implementation of the feedback mechanism and state transitions as defined by the VHDL code.

# Conclusion:

Throughout this project, we have extensively explored the application of LFSRs (Linear Feedback Shift Registers) in the realm of encryption and decryption. Our investigation revealed that LFSRs, due to their simplicity and efficiency, are highly effective in generating pseudo-random number sequences that form the basis of secure cryptographic keys.

And with the results that have been displayed above in which we have found that linear feedback shift register is very low memory cost consuming, which has a seed value to initiate the random number generation. LFSRs (Linear Feedback Shift Registers) have a long cycle duration to generate random as a true random number generator. It is more suitable for cryptography.