Projet : Implémentation du Chiffrement de César en VHDL

Module : Composants Programmables en VHDL

Réalisé par:

- CHABBI ABDERRAHMANE

Encadrant : Mr. Moumni

Date de Soumission : 21 mai 2024

Introduction

Contexte

Le chiffrement de César, également connu sous le nom de code de César, est une méthode de cryptographie par substitution simple dans laquelle chaque lettre du texte en clair est remplacée par une lettre située un nombre fixe de positions plus loin dans l'alphabet. Ce type de chiffrement doit son nom à Jules César, qui l'utilisait pour ses communications privées.

Objectif du Projet

Ce projet a pour objectif de réaliser une implémentation matérielle de l'algorithme de chiffrement de César en utilisant le langage de description matériel VHDL. Cette implémentation sera ensuite simulée à l'aide de l'outil de développement Quartus 13.1 .

Description de l'Algorithme

Le chiffrement de César déplace chaque lettre de l'alphabet d'un nombre fixe de positions, déterminé par une clé de chiffrement. Par exemple, avec une clé de 3, 'A' devient 'D', 'B' devient 'E', et ainsi de suite. Les lettres en fin d'alphabet sont déplacées circulairement, de sorte que 'X' devient 'A', 'Y' devient 'B', et 'Z' devient 'C'.

Plan du Document

Ce document décrit en détail l'implémentation de cet algorithme en VHDL. Il se compose des sections suivantes :

1. Code : le code VHDL développé.
2. 2. Simulation et Résultats : Les étapes de simulation effectuées et les résultats obtenus.
3. 3. Conclusion

Code:

```
Library IEEE;
Use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.STD_LOGIC_ARITH.ALL;
Use IEEE.STD_LOGIC_UNSIGNED.ALL;

Entity CaesarCipher is
    Port ( clk : in STD_LOGIC;
           Reset : in STD_LOGIC;
           Shift : in INTEGER range 0 to 25;
           Input_char : in STD_LOGIC_VECTOR(7 downto 0);
           Output_char : out STD_LOGIC_VECTOR(7 downto 0));
    End CaesarCipher;

    Architecture Behavioral of CaesarCipher is
    Begin
      Process(clk, reset)
      Variable char_ascii : integer range 0 to 255;
      Variable encrypted_ascii : integer range 0 to 255;
       Begin
        If reset = '1' then
           Output_char <= (others => '0');
```

```
        Elsif rising_edge(clk) then
            Char_ascii := to_integer(unsigned(input_char));

            -- Check if input is uppercase letter
            If char_ascii >= 65 and char_ascii <= 90 then
                Encrypted_ascii := ((char_ascii - 65) + shift) mod 26 + 65;
            -- Check if input is lowercase letter
            Elsif char_ascii >= 97 and char_ascii <= 122 then
                Encrypted_ascii := ((char_ascii - 97) + shift) mod 26 + 97;
            Else
                Encrypted_ascii := char_ascii; -- Leave non-letter characters
unchanged
            End if;

             Output_char                                                <=
std_logic_vector(to_unsigned(encrypted_ascii, 8));
        End if;
    End process;
End Behavioral;
```

L’écran de l’ordinateur ne fonctionnait plus, je n’ai donc pas pu réaliser l’ensemble du projet rapidement