library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity hill\_cipher is

port (

clk : in std\_logic;

reset : in std\_logic;

plaintext : in std\_logic\_vector(7 downto 0);

key : in std\_logic\_vector(15 downto 0);

ciphertext : out std\_logic\_vector(7 downto 0)

);

end entity hill\_cipher;

architecture rtl of hill\_cipher is

-- Matrice de clé 2x2

signal key\_matrix : std\_logic\_vector(31 downto 0);

-- Signaux intermédiaires

signal plaintext\_num : integer range 0 to 25;

signal ciphertext\_num : integer range 0 to 25;

signal temp : integer range 0 to 625;

begin

-- Charger la matrice de clé

key\_matrix <= key;

process(clk, reset)

begin

if reset = '1' then

ciphertext <= (others => '0');

elsif rising\_edge(clk) then

-- Convertir le texte clair en nombre

plaintext\_num <= to\_integer(unsigned(plaintext)) - to\_integer(unsigned'("00100000"));

-- Effectuer le chiffrement de Hill

temp <= (to\_integer(unsigned(key\_matrix(15 downto 8))) \* plaintext\_num) mod 26

+ (to\_integer(unsigned(key\_matrix(7 downto 0))) \* plaintext\_num) mod 26;

ciphertext\_num <= temp mod 26;

-- Convertir le texte chiffré en sortie

ciphertext <= std\_logic\_vector(to\_unsigned(ciphertext\_num + to\_integer(unsigned'("00100000")), 8));

end if;

end process;

end architecture rtl;