

Group members :

- SABBAR Anas
- CHAGRI Imad
- EL-ARIF Marwane
- MEBSOUT Ilyass

Supervised by :

- Pr. S EL MOUMNI

Encryption-Decryption: XOR VHDL

Encryption Module :

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity XOR_Encrypt is

Port (plaintext : in STD_LOGIC_VECTOR (7 downto 0);

key : in STD_LOGIC_VECTOR (7 downto 0);

ciphertext: out STD_LOGIC_VECTOR (7 downto 0));

end XOR_Encrypt;

architecture Behavioral of XOR_Encrypt is

begin

process(plaintext, key)

begin

ciphertext <= plaintext xor key;

end process;

end Behavioral;

Explanation :

- The XOR_Encrypt entity takes plaintext and key as 8-bit input vectors and produces ciphertext as an 8-bit output vector.
- Inside the architecture, the plaintext is XORed with the key using the xor operator, and the result is assigned to ciphertext.

Decryption Module:

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity XOR_Decrypt is

Port (ciphertext : in STD_LOGIC_VECTOR (7 downto 0);

key : in STD_LOGIC_VECTOR (7 downto 0);

plaintext : out STD_LOGIC_VECTOR (7 downto 0));

end XOR_Decrypt;

architecture Behavioral of XOR_Decrypt is

begin

process(ciphertext, key)

begin

plaintext <= ciphertext xor key;

end process;

end Behavioral;

Explanation :

- The XOR_Decrypt entity takes ciphertext and key as 8-bit input vectors and produces plaintext as an 8-bit output vector.
- Inside the architecture, the ciphertext is XORed with the key using the xor operator, and the result is assigned to plaintext.

Summary :

1. **Encryption and Decryption:** Both use the XOR operation with the same key.
2. **VHDL Implementation:** Simple modules that XOR input vectors with a key to produce output vectors.