

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Кафедра радіофізики та комп'ютерних технологій

Звіт
про виконання
лабораторної роботи № 12
"Основи мови VHDL"

Виконала
студентка групи ФЕІ - 41
Литвин Віра

Перевірив
доц. Рабик В.Г.

Львів - 2013

вивчення мови VHDL та набуття практичних навиків по проектуванню цифрових пристроїв з її використанням; реалізація принципових схем цифрових пристроїв в САПР QUARTUS II; експериментальна перевірка їх роботи на ПЛІС FPGA Cyclone III лабораторного стенду DE0.

3. З допомогою САПР QUARTUS II реалізувати довільну логічну функцію від трьох змінних, задану таблицею істинності, на мові VHDL. Виконати моделювання в часовій області цієї логічної функції в САПР QUARTUS II та експериментально перевірити роботу пристрою на лабораторному стенді DE0.

4. Записуємо в нього наступний код.

```

Library IEEE;
use IEEE.std_logic_1164.all;
entity lab_12_1 is
    port( A: in std_logic_vector (2 downto 0);
          Q: out std_logic_vector (7 downto 0));
end entity lab_12_1;
architecture Behave of lab_12_1 is
    begin
        process (A)
            begin
                case A is
                    when "000" => Q <= "00000001";
                    when "001" => Q <= "00000010";
                    when "010" => Q <= "00000100";
                    when "011" => Q <= "00001000";
                end case;
            end
        end process;
    end
end architecture Behave of lab_12_1;

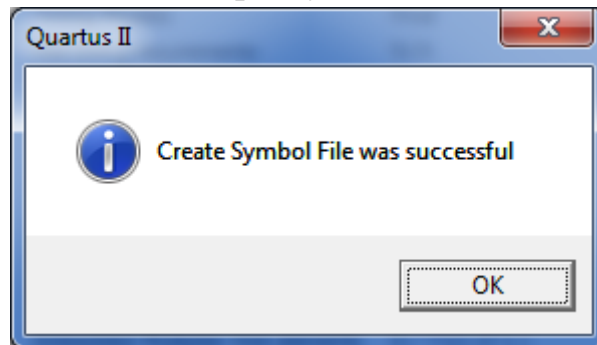
```

```

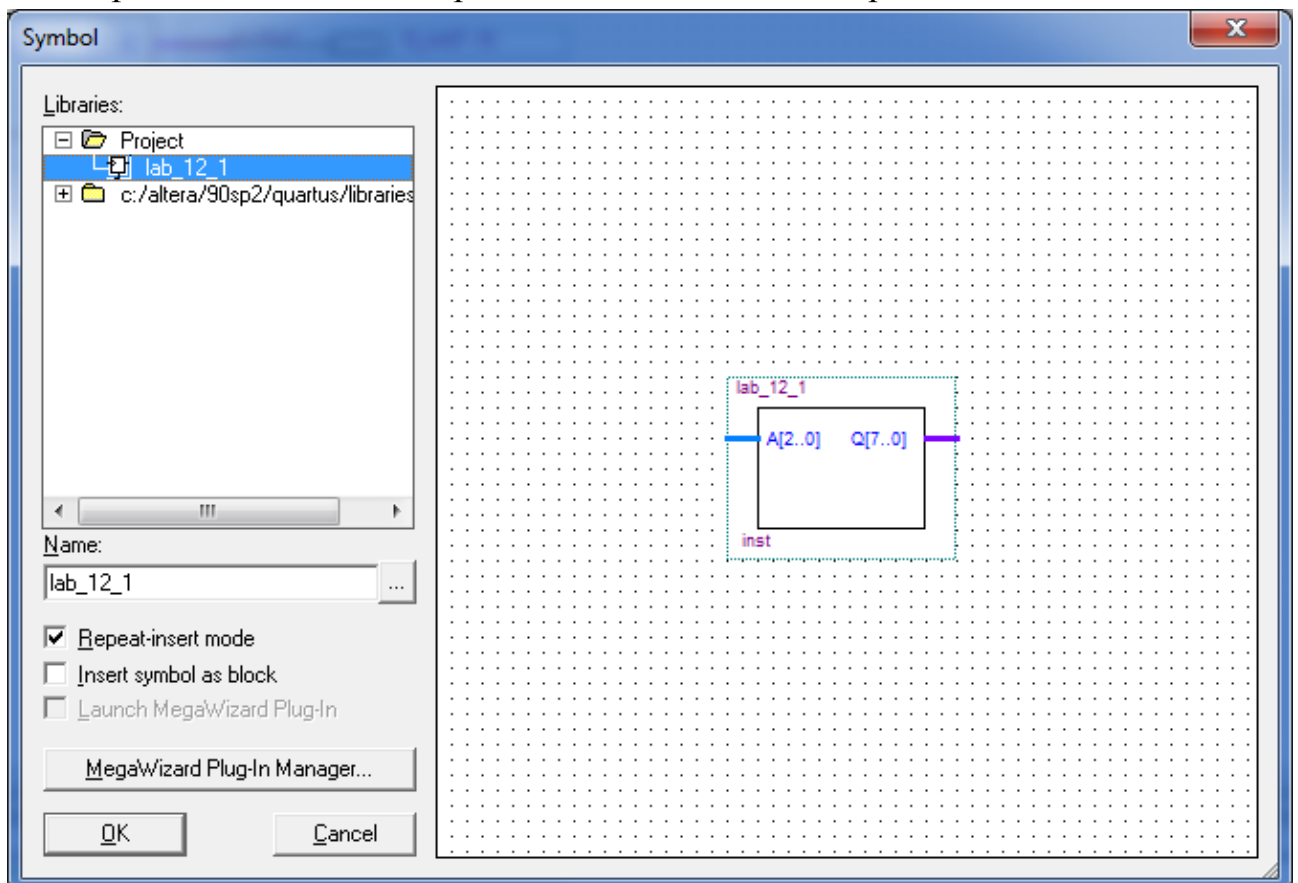
when "100" => Q <= "00010000";
when "101" => Q <= "00100000";
when "110" => Q <= "01000000";
when "111" => Q <= "10000000";
when others => Q <= "00000000";
end case;
end process;
end Behave;

```

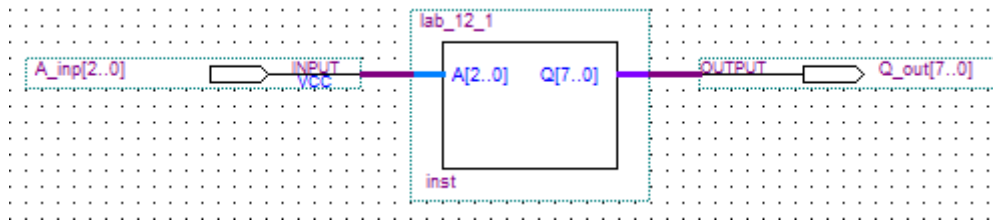
5. Компілюємо створений файл командою File - Create/_Update - Create Symbol Files for Current File. Після компіляції одержуємо підтвердження успішного створення символного файлу.



6. Створюємо новий файл, де складаємо принципову схему роботи щойноствореного пристрою. В меню компонент обираємо наш пристрій. І зі стандартних компонент обираємо вхідні та вихідні порти.



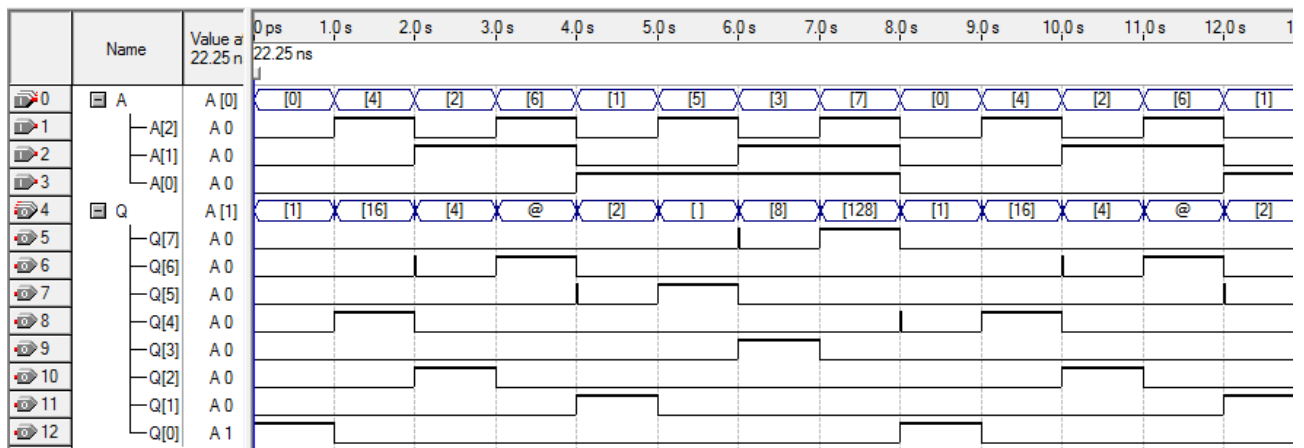
В результаті отримуємо наступну схему:



7. Компілюємо проект. Проводимо під'єднання до виводів ПЛІС.

	Node Name	Direction	Location	I/O Bank
1	A[2]	Input	PIN_H6	1
2	A[1]	Input	PIN_H5	1
3	A[0]	Input	PIN_J6	1
4	Q[7]	Output	PIN_D13	7
5	Q[6]	Output	PIN_F13	7
6	Q[5]	Output	PIN_F12	7
7	Q[4]	Output	PIN_G12	7
8	Q[3]	Output	PIN_H13	7
9	Q[2]	Output	PIN_H12	7
10	Q[1]	Output	PIN_F11	7
11	Q[0]	Output	PIN_E11	7

8. Створюємо новий файл для симуляції роботи пристрою. Задаємо вхідні сигнали. Запускаємо засіб симуляції.



9. Перевіряємо роботу програми на лабораторному стенді.

10. Виконуючи завдання 2, створюємо новий файл із розширенням *.vhd .

11. Знайдемо таблицю істинності семисегментного дешифратора.

№	X3	X2	X1	X0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1

8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0
A	1	0	1	0	0	0	0	1	0	0	0
B	1	0	1	1	1	1	0	0	0	0	0
C	1	1	0	0	1	1	1	0	0	1	0
D	1	1	0	1	1	0	0	0	0	1	0
E	1	1	1	0	0	1	1	0	0	0	0
F	1	1	1	1	0	1	1	1	0	0	0

12. Записуємо в створений файл наступний код, відповідно до таблиці.

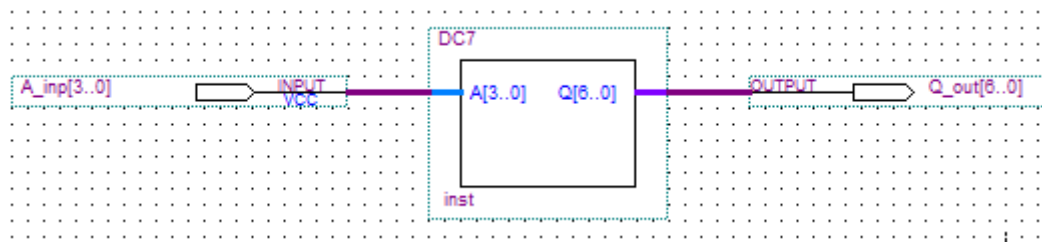
```

Library IEEE;
use IEEE.std_logic_1164.all;
entity DC7 is
    port( A: in std_logic_vector (3 downto 0);
          Q: out std_logic_vector (6 downto 0));
end entity DC7;
architecture Behave of DC7 is
    begin
        process (A)
            begin
                case A is
                    when "0000" => Q <= "0000001";
                    when "0001" => Q <= "1001111";
                    when "0010" => Q <= "0010011";
                    when "0011" => Q <= "0000110";
                    when "0100" => Q <= "1001100";
                    when "0101" => Q <= "0100100";
                    when "0110" => Q <= "0100000";
                    when "0111" => Q <= "0001111";
                    when "1000" => Q <= "0000000";
                    when "1001" => Q <= "0000100";
                    when "1010" => Q <= "0001000";
                    when "1011" => Q <= "1100000";
                    when "1100" => Q <= "1110010";
                    when "1101" => Q <= "1000010";
                    when "1110" => Q <= "0110000";
                    when "1111" => Q <= "0111000";
                    when others => Q <= "0000001";
                end case;
            end process;
        end Behave;
    end

```

13. Компілюємо створений файл командою File - Create/_Update - Create Symbol Files for Current File. Після компіляції одержуємо підтвердження успішного створення символьного файлу.

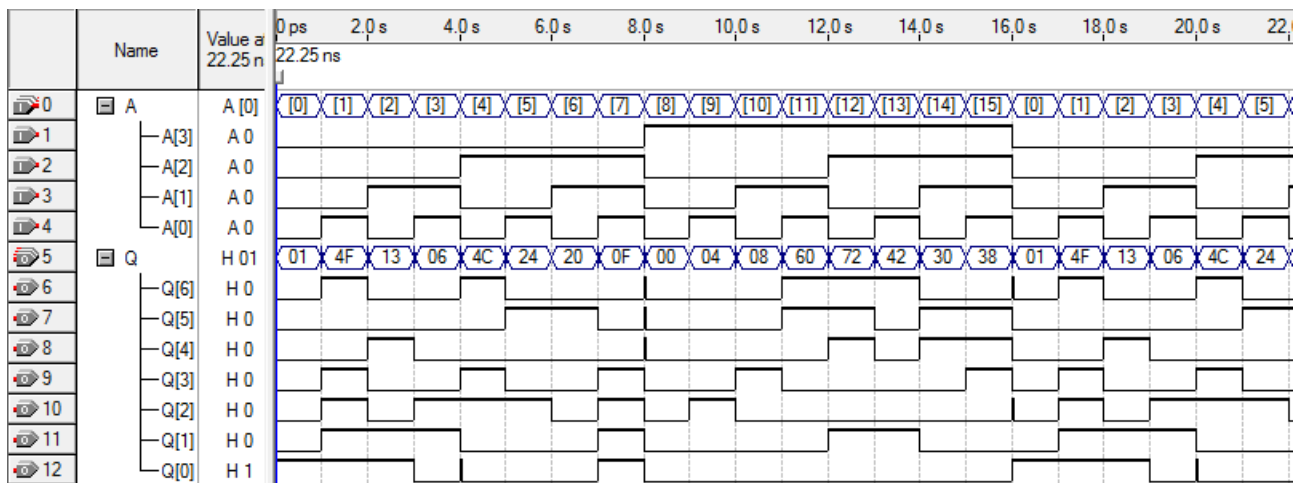
14. Створюємо новий файл, де складаємо принципову схему роботи щойноствореного пристрою. В меню компонент обираємо наш пристрій. І зі стандартних компонент обираємо вхідні та вихідні порти. В результаті отримуємо наступну схему:



15. Компілюємо проект. Проводимо під'єднання до виводів ПЛІС.

1	A[3]	Input	PIN_G4	1	B1_N0
2	A[2]	Input	PIN_H6	1	B1_N0
3	A[1]	Input	PIN_H5	1	B1_N0
4	A[0]	Input	PIN_J6	1	B1_N0
5	Q[6]	Output	PIN_F14	7	B7_N0
6	Q[5]	Output	PIN_B17	7	B7_N1
7	Q[4]	Output	PIN_A17	7	B7_N1
8	Q[3]	Output	PIN_E15	7	B7_N0
9	Q[2]	Output	PIN_B16	7	B7_N1
10	Q[1]	Output	PIN_A16	7	B7_N1
11	Q[0]	Output	PIN_D15	7	B7_N0

16. Створюємо новий файл для симуляції роботи пристрою. Задаємо вхідні сигнали. Запускаємо засіб симуляції.



17. Перевіряємо роботу програми на лабораторному стенді.

18. Виконуючи завдання 3, створюємо новий файл із розширенням *.vhd .

Обираємо функцію від трьох змінних. Нехай це буде трьохвходове виключаюче або. Таблиця істинності такої функції:

A2	A1	A0	$A1 \oplus A0$	$A2 \oplus (A1 \oplus A0)$
0	0	0	0	0
0	0	1	1	1

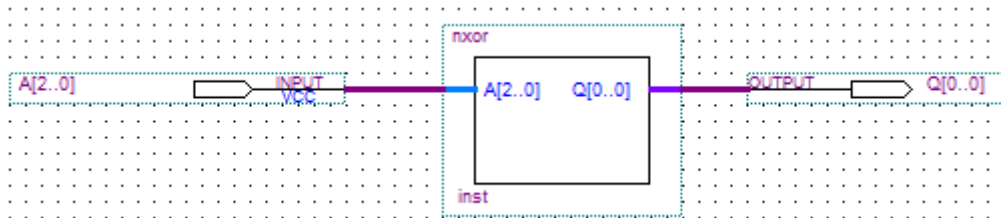
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

19. Записуємо в створений раніше файл наступний код відповідно до таблиці істинності.

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity nxor is
    port( A: in std_logic_vector (2 downto 0);
          Q: out std_logic_vector (0 downto 0));
end entity nxor;
architecture Behave of nxor is
    begin
        process (A)
            begin
                case A is
                    when "000" => Q <= "0";
                    when "001" => Q <= "1";
                    when "010" => Q <= "1";
                    when "011" => Q <= "0";
                    when "100" => Q <= "1";
                    when "101" => Q <= "0";
                    when "110" => Q <= "0";
                    when "111" => Q <= "1";
                    when others => Q <= "0";
                end case;
            end process;
        end Behave;
    end
```

20. Компілюємо створений файл командою File - Create/_Update - Create Symbol Files for Current File. Після компіляції одержуємо підтвердження успішного створення символьного файлу.

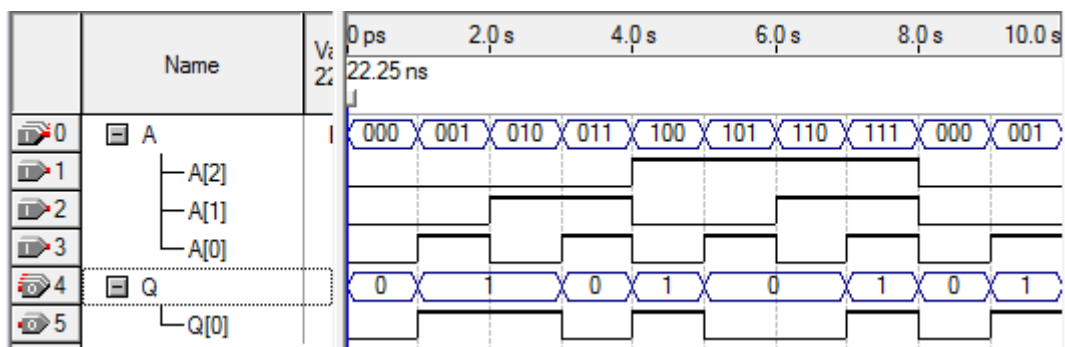
21. Створюємо новий файл, де складаємо принципову схему роботи щойноствореного пристрою. В меню компонент обираємо наш пристрій. І зі стандартних компонент обираємо вхідні та вихідні порти. В результаті отримуємо наступну схему:



22. Компілюємо проект. Проводимо під'єднання до виводів ПЛІС.

1		A[2]	Input	PIN_H6	1
2		A[1]	Input	PIN_H5	1
3		A[0]	Input	PIN_J6	1
4		Q[0]	Output	PIN_J1	1

23. Створюємо новий файл для симуляції роботи пристрою. Задаємо вхідні сигнали. Запускаємо засіб симуляції.



24. Перевіряємо роботу програми на лабораторному стенді.

Висновок:

Під час виконання цієї лабораторної роботи розглянуто основні особливості мови VHDL та використано її можливості для проектування цифрових пристроїв.

Для набуття навичок проектування пристроїв за допомогою опису їх мовою VHDL було реалізовано трирозрядний дешифратор, дешифратор семисегментного коду та логічну функцію від трьох змінних - трьохвходове виключаюче або .