**Лабораторна робота №2**

**Тема:** Базові логічні елементи

**Мета:** засвоїти знання про базові логічні елементи мови Verilog .

**Теоретичні відомості**

Основною структурною одиницею устрою, що описується за допомогою мови опису апаратури, є **модуль**. Всі модулі є незалежними об'єктами рівня проекту і не можуть включати всередині себе опису інших модулів. Але модуль може містити всередині себе посилання на інші модулі. Для розробки великих програм зручно розміщувати модулі в різних файлах в межах одного проекту.

У загальному випадку модуль має наступну структуру:

module Ім'я\_модуля (Інтерфейс\_модуля);

Опис\_інтерфейсу

...

Внутрішня\_реалізація\_модуля

...

Endmodule

**module, endmodule** – зарезервовані слова, що означають початок і кінець модуля. **Імя\_модуля** – ідентифікатор, за яким можна буде звернутися до даному модулю з інших частин програми. **Інтерфейс\_модуля** – перелік вхідних і вихідних сигналів модуля, при цьому вказівки типу та напрямку немає. **Опис\_інтерфейсу** – перелічуються тип та напрямки сигналів, які описано в інтерфейсі модуля. **Внутрішня\_реалізація\_модуля** – це послідовність операторів мови Verilog, яка описує поведінку даного модуля. Структурний опис архітектури представляє структуру об'єкта як композицію з компонентів, які з'єднані між собою і обмінюються сигналами. Функції, що реалізуються компонентами, в явному вигляді не вказуються. Структурний опис включає імена і типи компонентів, з яких складається схема, а також їх зв'язки.

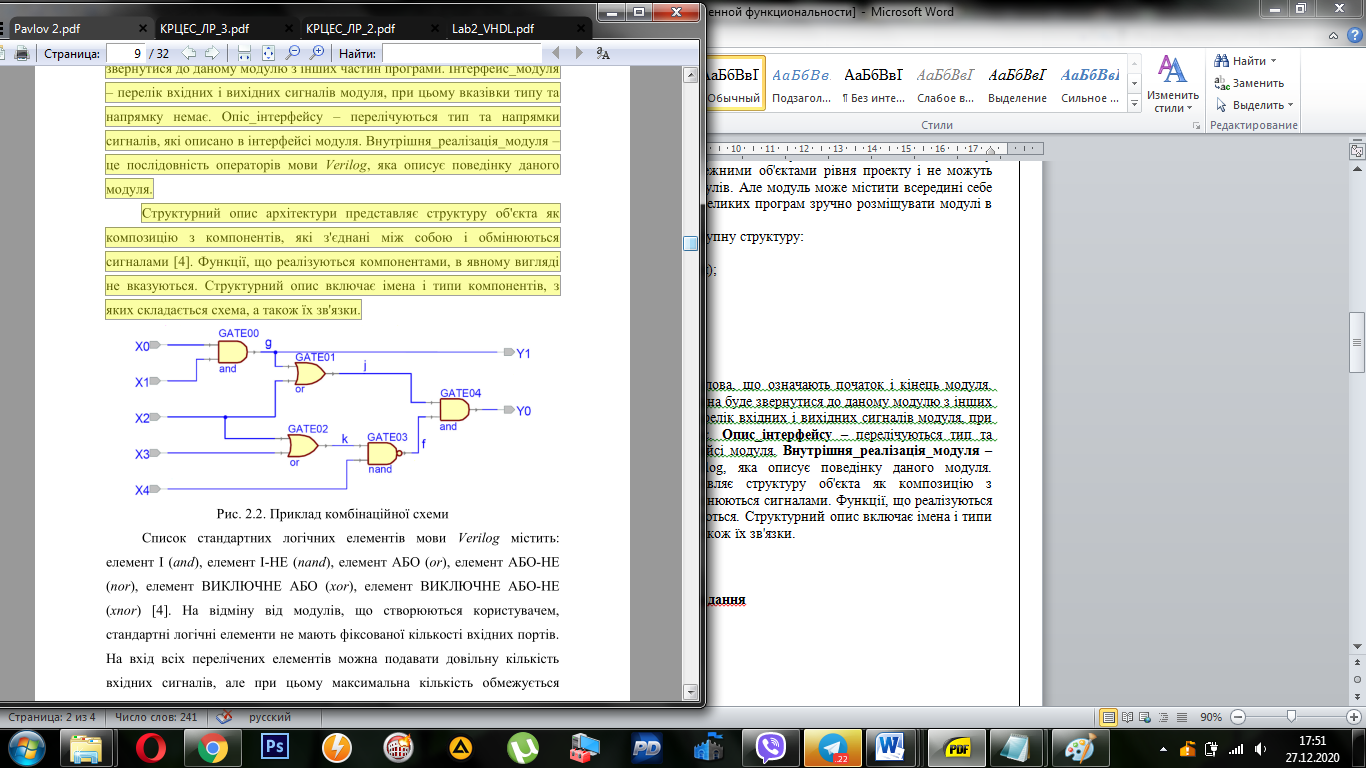


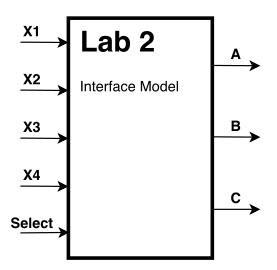
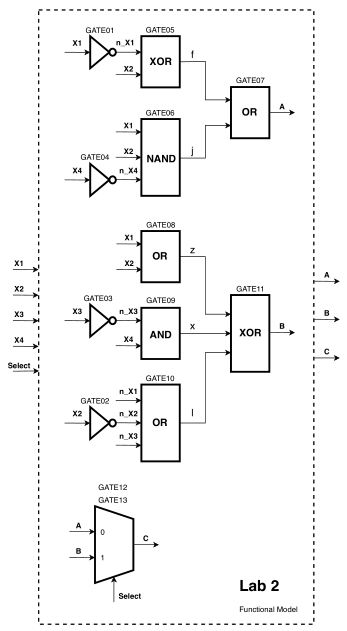
Рис. 1. Приклад комбінаційної схеми

Список стандартних логічних елементів мови Verilog містить: елемент І (and), елемент І-НЕ (nand), елемент АБО (or), елемент АБО-НЕ (nor), елемент ВИКЛЮЧНЕ АБО (xor), елемент ВИКЛЮЧНЕ АБО-НЕ (xnor). На відміну від модулів, що створюються користувачем, стандартні логічні елементи не мають фіксованої кількості вхідних портів.

На вхід всіх перелічених елементів можна подавати довільну кількість вхідних сигналів, але при цьому максимальна кількість обмежується програмою синтезу. Окрім перерахованих вище примітивів мова Verilog містить **шість базових логічних примітивів: buf, not, bufif1, bufif0, notif1, notif0.** Перши два являють собою стандартний буфер та інвертор, а решта – буфери та інвертори с додатковим входом дозволу видачі вихідного сигналу.

**Завдання**

Реалізувати програму, яка описує схему.

** **

**Код програми**

module lab2(X1,X2,X3,X4,SELECT,A,B,C);

input X1,X2,X3,X4,SELECT;

// Входные сигналы модуля

output A,B,C;

// Выходные сигналы модуля

wire X1,X2,X3,X4,SELECT,A,B,C;

wire h,j,d,e,f,l,x,z,y,w,u,v;

not GATE01(n\_X1,X1);

not GATE02(n\_X2,X2);

not GATE03(n\_X3,X3);

not GATE04(n\_X4,X4);

xor GATE05(f,X2,n\_X1 );

nand GATE06(j,n\_X4,X2,X1);

or GATE07(A,f,j);

or GATE08(z,X1,X2);

and GATE09(x,n\_X3,X4);

or GATE10(l,n\_X3,n\_X1,n\_X2);

xor GATE11(B,l,z,x);

bufif1 GATE15 (C,B,SELECT);

bufif0 GATE16 (C,A,SELECT);

endmodule

**Висновок:** під час виконання лабораторної роботи ми засвоїли знання про базові логічні елементи мови Verilog . Реалізували програму, яка описує схему згідно завдання.