

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Кафедра конструювання електронно-обчислювальної апаратури

**Курс: «Функціонально-логічне проектування»**  
**Лабораторна робота №1**  
**«Моделювання найпростіших логічних схем»**

Виконав студент  
Дем'янчук Тарас Миколайович  
гр. ДК-12  
Викладач:  
Омелян А.В.

Київ – 2023

**Тема роботи:** моделювання найпростіших логічних схем.

**Мета роботи:** ознайомитись з особливостями побудови та моделювання логічних схем в графічному редакторі Quartus II, навчитись виконувати моделювання логічних схем за допомогою вбудованого симулятора та із застосуванням плати налагодження.

### Основні теоретичні відомості

#### Логічні елементи та базові логічні функції

**Логічний елемент І (AND).** Логічний елемент І реалізує логічну операцію кон'юнкції. Словесний опис: на виході логічного елементу І формуватиметься одиниця тоді і тільки тоді, коли на всі його входи подано одиниці. Аналітичний запис операції кон'юнкції:

$$y = x_2 \cdot x_1 \text{ або } y = x_2 \wedge x_1 \text{ або } y = x_2 \& x_1$$

**Логічний елемент АБО (OR).** Логічний елемент АБО реалізує логічну операцію диз'юнкції. Словесний опис: на виході логічного елементу АБО формуватиметься одиниця тоді і тільки тоді, коли хоча б на один його вхід буде подано одиницю. Аналітичний запис операції диз'юнкції:

$$y = x_2 \vee x_1$$

**Логічний елемент «Виключне АБО» (XOR).** Логічний елемент «Виключне АБО» реалізує логічну операцію сума по модулю 2. Словесний опис функціонування: якщо елемент «Виключне АБО» має два вхідні сигнали, то на його виході формуватиметься одиниця тоді і тільки тоді, коли на обох його входах сигнали відрізняються. Аналітичний запис операції Виключне АБО:

$$y = x_2 \oplus x_1 = \bar{x}_2 \cdot x_1 \vee x_2 \cdot \bar{x}_1.$$

Теореми та тотожності алгебри логіки:

- 1) Ідемпотентні закони
- 2) Комутативні закони
- 3) Асоціативні закони
- 4) Дистрибутивні закони
- 5) Закони заперечення
- 6) Закони двоїстості (закони Де Моргана)
- 7) Закони подвійного заперечення
- 8) Закони поглинання (абсорбції)
- 9) Операції склеювання
- 10) Закони узагального склеювання

### **Приоритетність операцій:**

У алгебрі логіки, а отже і при записі логічних рівнянь прийнято наступну пріоритетність операцій: якщо в логічний вираз входять операції диз'юнкції та кон'юнкції, то кон'юнкція має вищий пріоритет виконання (виконується першою), а диз'юнкція – нижчий пріоритет. Порядок операцій можна довільно змінювати за допомогою дужок.

Якщо додатково в розгляд ввести операцію суми по модулю 2, то можна показати, що для того, щоб зберегти правильність результату виконання обчислень, її пріоритет має бути вищим за пріоритет диз'юнкції і нижчим за пріоритет кон'юнкції, тобто:

Пріоритет кон'юнкції > Пріоритет суми по модулю 2 > Пріоритет диз'юнкції

Операція заперечення є одномісною, тому застосовується до аргументу в першу чергу.

## **Порядок виконання роботи**

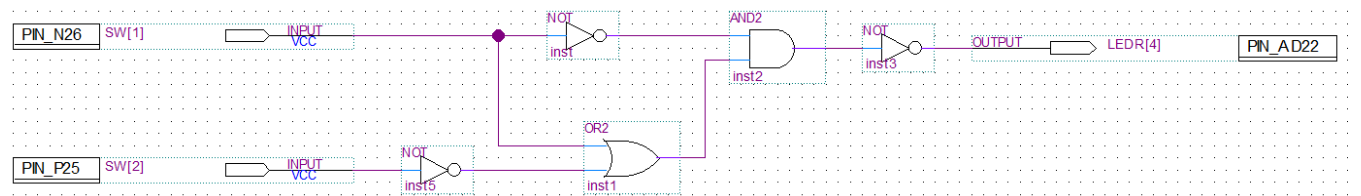
1. Побудувати в графічному редакторі системи Quartus II логічну схему відповідно до заданого варіанту (спрощення схеми не виконувати).
2. Створити файл з сигналами для моделювання, який би покривав всі можливі комбінації вхідних сигналів для схеми.
3. Провести моделювання у вбудованому симуляторі Quartus II та отримати часові діаграми. Впевнитись, що схема функціонує відповідно до заданого логічного рівняння.
4. Створити програму на зручній для Вас мові програмування, яка б реалізовувала роботу логічної схеми. Введення сигналів має бути реалізоване з клавіатури. Порівняти результати роботи програми з результатами моделювання в симуляторі Quartus II.
5. Обравши на платі налагодження необхідні засоби введення-виведення (кнопки та світлодіоди) модифікуйте схему та налаштуйте відповідні виводи ПЛІС. Створіть та завантажте прошивку на плату налагодження. Вводячи через кнопки чи перемикачі сигнали переконайтесь, що апаратна реалізація схеми функціонує згідно заданого логічного рівняння.

## ХІД РОБОТИ

1. Побудувати в графічному редакторі системи Quartus II логічну схему відповідно до заданого варіанту (варіант 4) (спрощення схеми не виконувати).


$$4. \quad \overline{\bar{x}_1 \cdot (x_1 \vee \bar{x}_2)}$$

Побудовано логічну схему



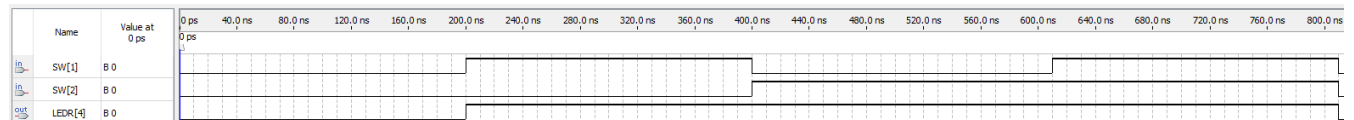
2. Створити файл з сигналами для моделювання, який би покривав всі можливі комбінації вхідних сигналів для схеми.

Створено файл з сигналами для моделювання

 Simulation Waveform Editor - [LW1.sim.vwf (Read-Only)]

3. Провести моделювання у вбудованому симуляторі Quartus II та отримати часові діаграми. Впевнитись, що схема функціонує відповідно до заданого логічного рівняння.

Проведено моделювання у вбудованому симуляторі Quartus II та отримано часові діаграми.



Переконалися, що схема функціонує відповідно до заданого логічного рівняння (результати теоретичного розрахунку логічного рівняння нижче).

$$4. \quad \overline{x_1} \cdot (x_1 \vee \overline{x_2})$$

$x_1$	$x_2$	OUT
0	0	0
1	0	1
0	1	1
1	1	1

$$1) \quad \overline{0} \cdot (0 + \overline{0}) = \overline{1 \cdot 1} = 0$$

$$2) \quad \overline{1} \cdot (1 + \overline{0}) = \overline{0 \cdot 1} = 1$$

$$3) \quad \overline{0} \cdot (0 + \overline{1}) = \overline{1 \cdot 0} = 1$$

$$4) \quad \overline{1} \cdot (1 + \overline{1}) = \overline{0 \cdot 1} = 1$$

4. Створити програму на зручній для Вас мові програмування, яка б реалізовувала роботу логічної схеми. Введення сигналів має бути реалізоване з клавіатури. Порівняти результати роботи програми з результатами моделювання в симуляторі Quartus II.

Реалізовано програму, яка функціонує у відповідності до роботи логічної схеми та порівняно результати роботи програми з результатами моделювання в симуляторі Quartus II (результати моделювання в симуляторі Quartus II в пункті 3).

```

1 #include <stdio.h>
2
3 int BulFunc(int x1, int x2)
4 {
5     int NOT_x1 = 0;
6     if(0 == x1) NOT_x1 = 1;
7
8     int NOT_x2 = 0;
9     if(0 == x2) NOT_x2 = 1;
10
11     int disjunctoin = x1 + NOT_x2;
12
13     int conjunction = NOT_x1 * disjunctoin;
14
15     int NOT_conjunction = 0;
16     if(0 == conjunction) NOT_conjunction = 1;
17
18     return NOT_conjunction;
19 }
20
21 int main()
22 {
23     int x1 = false, x2 = false;
24
25     while(true)
26     {
27         printf("Enter x1:"); scanf("%d", &x1);
28         printf("Enter x2:"); scanf("%d", &x2);
29
30         printf("BulFunc return = %d\n\n", BulFunc(x1, x2));
31     }
32
33     return 0;
34 }

```

```

[taras@tade23 FLP_Lw1]$ gcc main.c -o main
[taras@tade23 FLP_Lw1]$ ./main
Enter x1:0
Enter x2:0
BulFunc return = 0

Enter x1:1
Enter x2:0
BulFunc return = 1

Enter x1:0
Enter x2:1
BulFunc return = 1

Enter x1:1
Enter x2:1
BulFunc return = 1

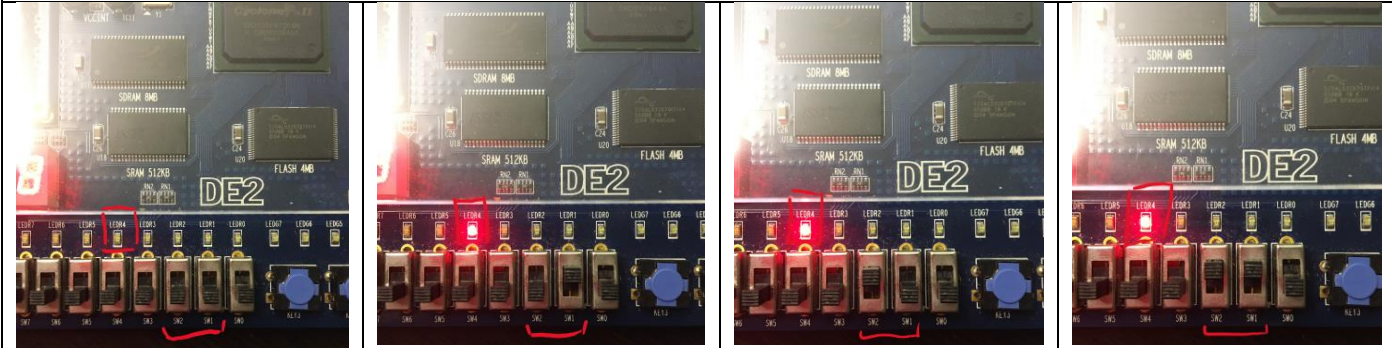
Enter x1:|

```

22:03  
Wednesday Mar

5. Обравши на платі налагодження необхідні засоби введення-виведення (кнопки та світлодіоди) модифікуйте схему та налаштуйте відповідні виводи ПЛІС. Створіть та завантажте прошивку на плату налагодження. Вводячи через кнопки чи перемикачі сигнали переконайтесь, що апаратна реалізація схеми функціонує згідно заданого логічного рівняння.

Переконалися, що апаратна реалізація схеми функціонує згідно заданого логічного рівняння (результати теоретичного розрахунку логічного рівняння в пункті 3).



Де: SW1 та SW2 – X0 та X1 відповідно, LEDR4 – OUT.

### Висновок

Протягом виконання даної лабораторної роботи було:

- у першому пункті побудовано логічну схему у відповідності до заданого варіанту.
- у другому пункті створено файл з сигналами для моделювання.
- у третьому пункті проведено модулювання у вбудованому симуляторі та отримано часові діаграми, а також переконалися, що схема функціонує відповідно до заданого логічного рівняння.
- у четвертому пункті написано програму на мові <C>, що реалізовувала роботу логічної схеми. Також переконалися, що вона дійсно працює у відповідності до заданого рівняння.
- у п'ятому пункті переконалися, що апаратна реалізація працює і відповідності до заданого рівняння.