Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

# Лабораторна робота №4

з дисципліни «Комп’ютерна схемотехніка» на тему

**«**РОЗРОБЛЕННЯ СХЕМ КОМБІНАЦІЙНОЇ ЛОГІКИ**»**

Виконав: студент групи ІО-31

Мукосій Б.Ю.

Перевірив: Нікольський С. С.

Київ-2025

# Мета роботи:

* Ознайомлення з принципами побудови та моделювання схем комбінаційної логіки.
* Навчитись реалізовувати логічні функції різними способами за допомогою логічних елементів (І, АБО, НЕ).
* Перевірити працездатність схем комбінованої логіки шляхом моделювання різних комбінацій вхідних сигналів.

# Хід роботи

1. Потрібно описати мовою Verilog схему виключаюче «АБО» (XOR). Таблиця істинності для XOR (табл. 1):

Таблиця 1

Таблиця істинності

| x1 | x2 | f |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1. Схема, яка реалізує виключаюче «АБО» на рис. 1.

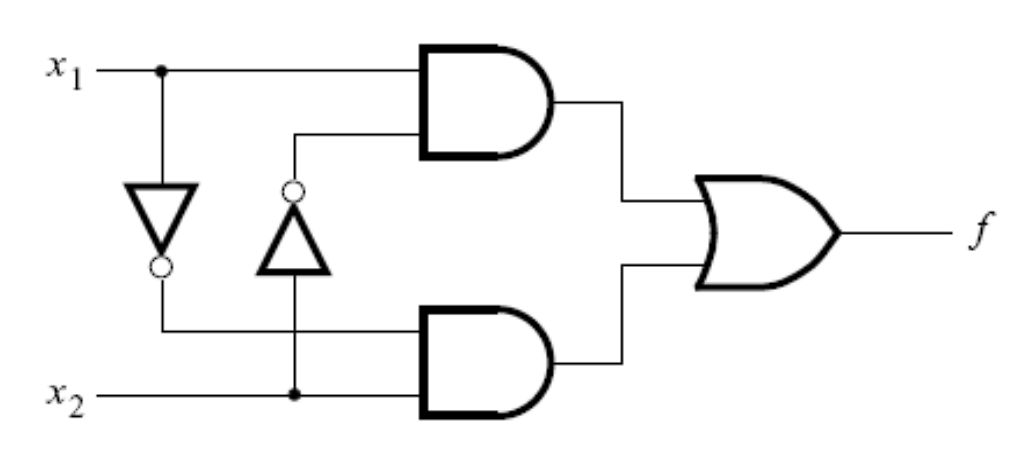


Рисунок 1.

1. Створюю новий проєкт в середовищі ModelSim (рис. 2)

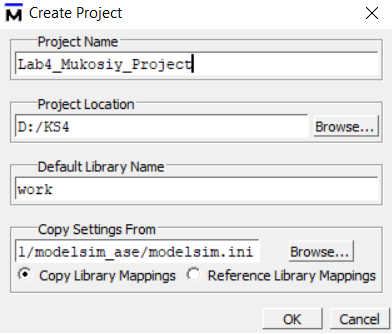


Рисунок 2.

1. Обираю команду Create New File та задаю ім’я файлу (рис. 3).

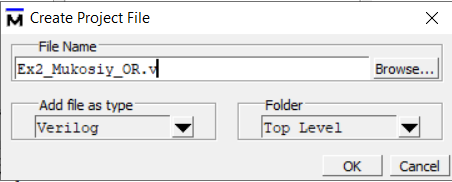


Рисунок 3.

1. Відкриваю текстовий редактор та набираю текст програми у створений файл. Вигляд програмного коду представлено на рис. 4.

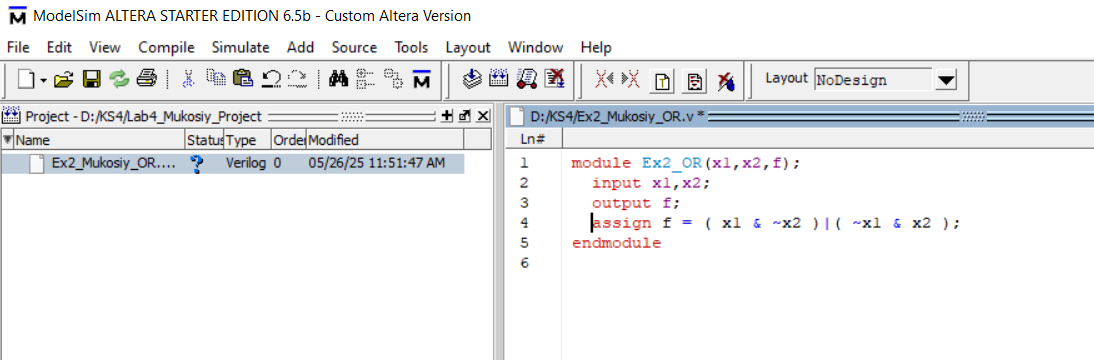


Рисунок 4.

1. Компілюю проєкт (рис. 5). Для цього вибираю у меню Compile команду Compile All.

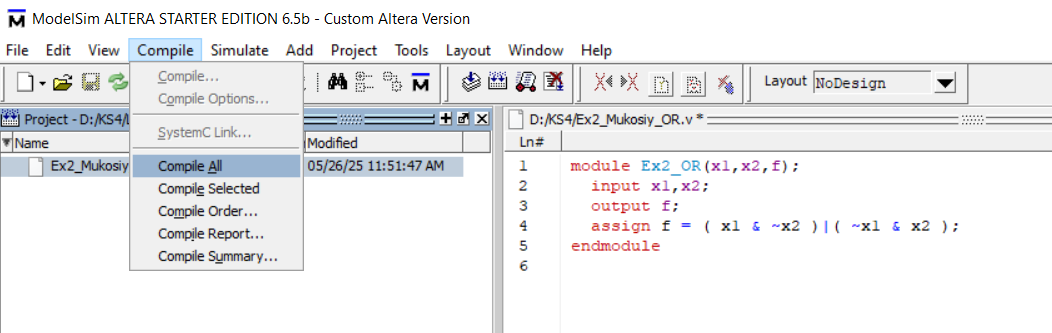


Рисунок 5.

1. Інформація про результат компіляції з'явиться у вікні повідомлень (Transcript). Якщо компіляція завершилася успішно, зміниться вигляд статусу біля імені файлу (рис. 6).

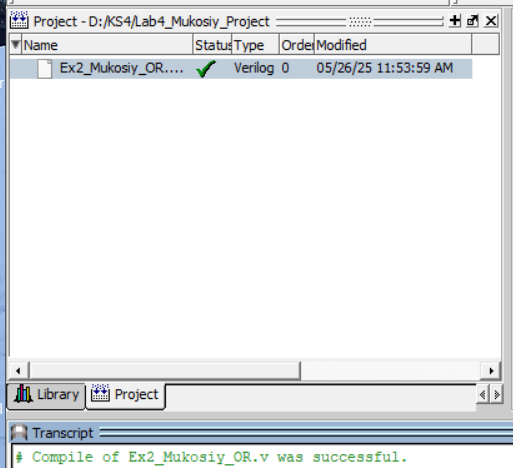


Рисунок 6.

1. Переходжу до режиму моделювання. Для цього в меню Simulate вибираю команду Start Simulation. У діалоговому вікні Start Simulation, що відкрилося, вказую файл верхнього рівня ієрархії для моделювання (створений файл з робочої бібліотеки work, рис. 7).

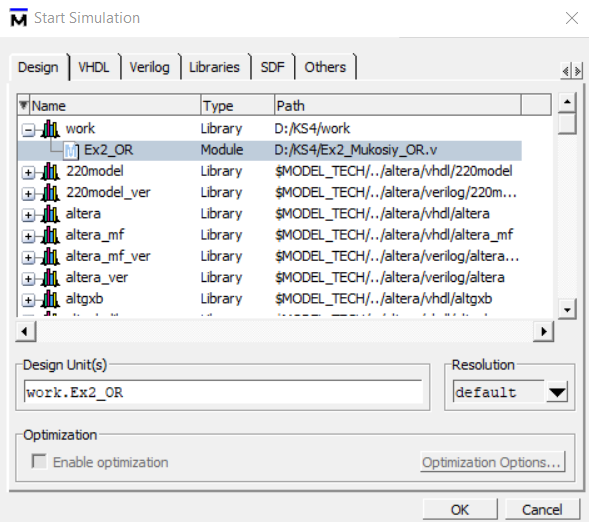


Рисунок 7.

1. Створюю файл із вхідними тестовими сигналами. У меню File вибираю команду New – Source – Do. У новому вікні, що відкрилося, вводжу код і зберігаю файл під ім'ям Stimul.do Результат на рис. 8.

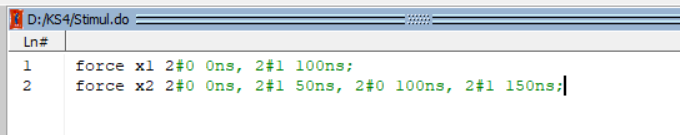


Рисунок 8.

1. Роблю вікно Object активним та додаю дані сигнали в окреме вікно для перегляду сигналів, які присутні в даному модулі (рис. 9).

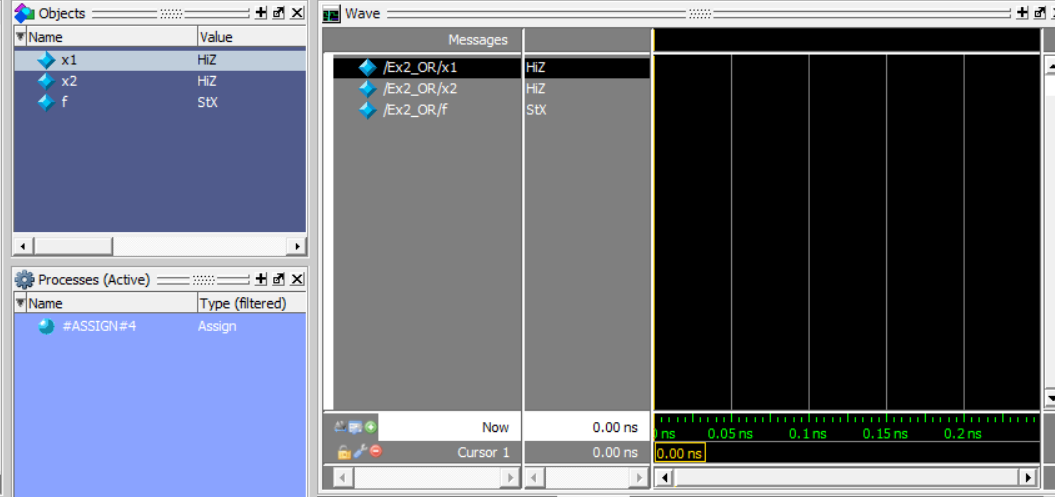


Рисунок 9.

1. Підключаю тестовий файл до проєкту в меню Tools, вибираючи команду TCL – Execute Macro та вказую файл Stimul.do (рис. 10).

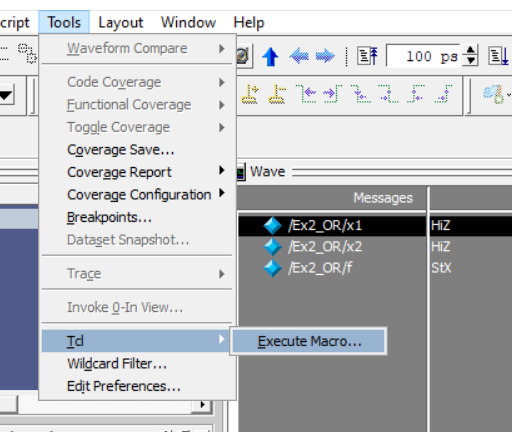


Рисунок 10.

1. Запускаю моделювання з меню Simulate командою Run – Run–All (рис. 11).

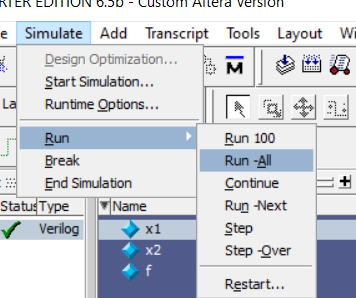


Рисунок 11.

1. Змінюю масштаб діаграми та переглядаю результат (рис. 12).

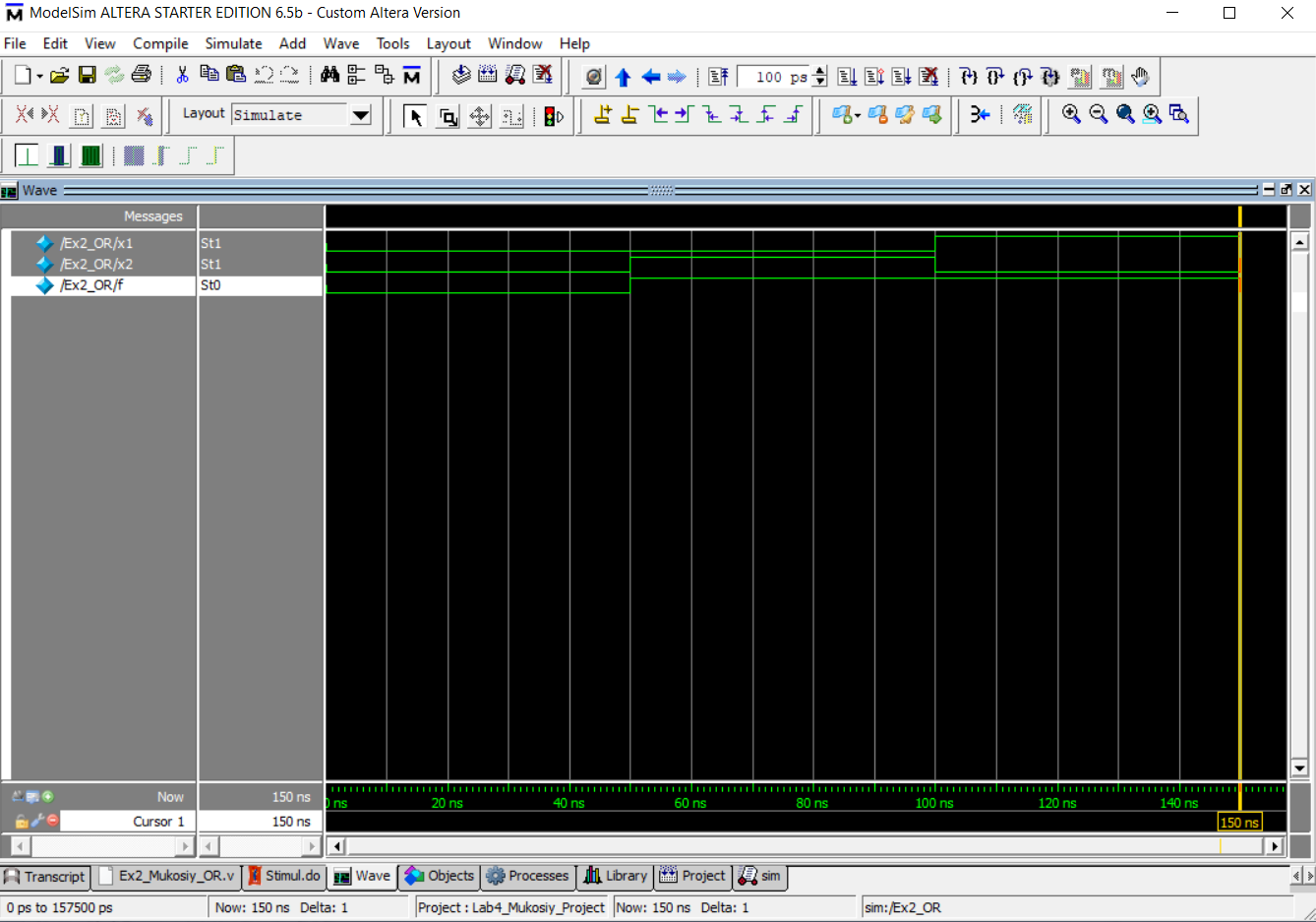


Рисунок 12.

1. Результат, що на рис. 12 підтвердив, що схема виключаючого «АБО» працює вірно.

**Самостійне завдання**

1. Визначаю свій варіант перемикальної функції. Для цього номер варіанту переводжу в двійкову систему числення і записую шість його молодших розрядів у вигляді слова h6 h5 h4 h3 h2 h1. Значення hi підставляю в табл. 2 – 3.

Мій номер залікової книжки – 3116. Перекладаю його у двійкову систему числення і отримую двійкове число:

311910 = 110000101100₂.

Десять молодших розрядів отриманого двійкового числа (табл. 2) застосовую для визначення варіанта завдання.

Таблиця 2

Визначення варіанта завдання

| 1 | 0 | 1 | 1 | 0 | 0 |
| --- | --- | --- | --- | --- | --- |
| h6 | h5 | h4 | h3 | h2 | h1 |

1. Відповідно до визначеного варіанта (табл. 2) формую вихідну таблицю істинності для виконання завдання на підставі заданої таблиці істинності (табл. 3) і визначаю набір логічних елементів для побудови комбінаційних схем на підставі табл. 4. За необхідності виконаю мінімізацію за допомогою діаграми Вейча.

Таблиця 3

Параметри перемикальної функції

| *x3* | *x2* | *x1* | *f4* |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Таблиця 4

Елементний базис

| h1 h5 h2 | Логічні елементи |
| --- | --- |
| 0 0 0 | І-НЕ, І |

1. Щоб побудувати комбінаційну схему в елементному базисі І-НЕ знаходжу МДНФ за допомогою діаграми Вейча (рис. 13).

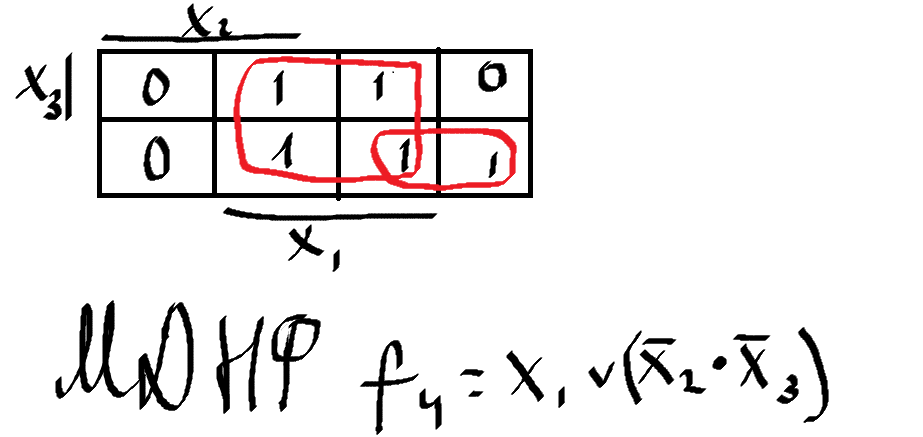
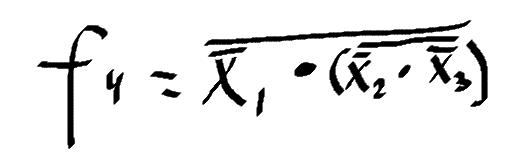


Рисунок 13.

За правилом де Моргана:

Комбінаційна схема в базисі І-НЕ представлена на рис. 14.

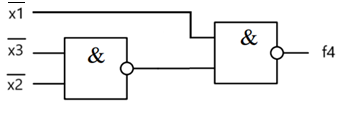
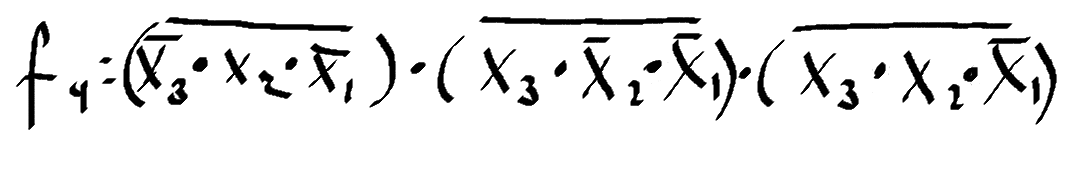
**

Рисунок 14.

Для побудови функції в базисі І, І-НЕ використаємо ДКНФ:



Комбінаційна схема в базисі АБО, І-НЕ представлена на рис. 15.

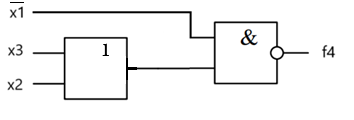


Рис. 15.

1. Дві комбінаційні схеми готові, далі виконую їх моделювання роботи в програмі ModelSim. Для цього створюю новий файл на мові Verilog з описом роботи пристрою в базисі І-НЕ (рис. 16).

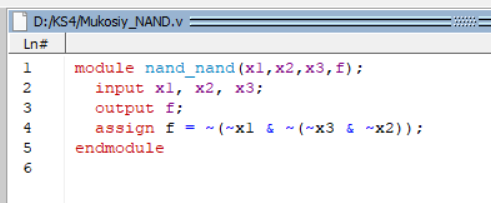


Рисунок 16.

1. Компілюю проєкт та створюю файл із вхідними тестовими сигналами (рис. 17).

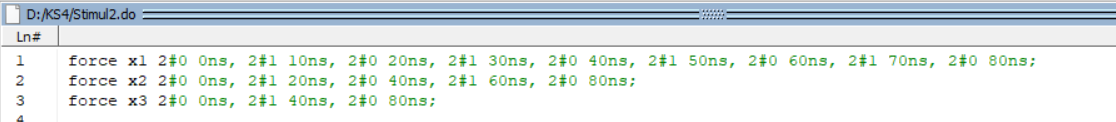


Рисунок 17.

1. Підключаю тестовий файл до проєкту в меню Tools, вибираючи команду TCL – Execute Macro та вказую файл Stimul2.1.do. Запускаю моделювання з меню Simulate командою Run – Run–All. Змінюю масштаб діаграми та переглядаю результат (рис. 18).

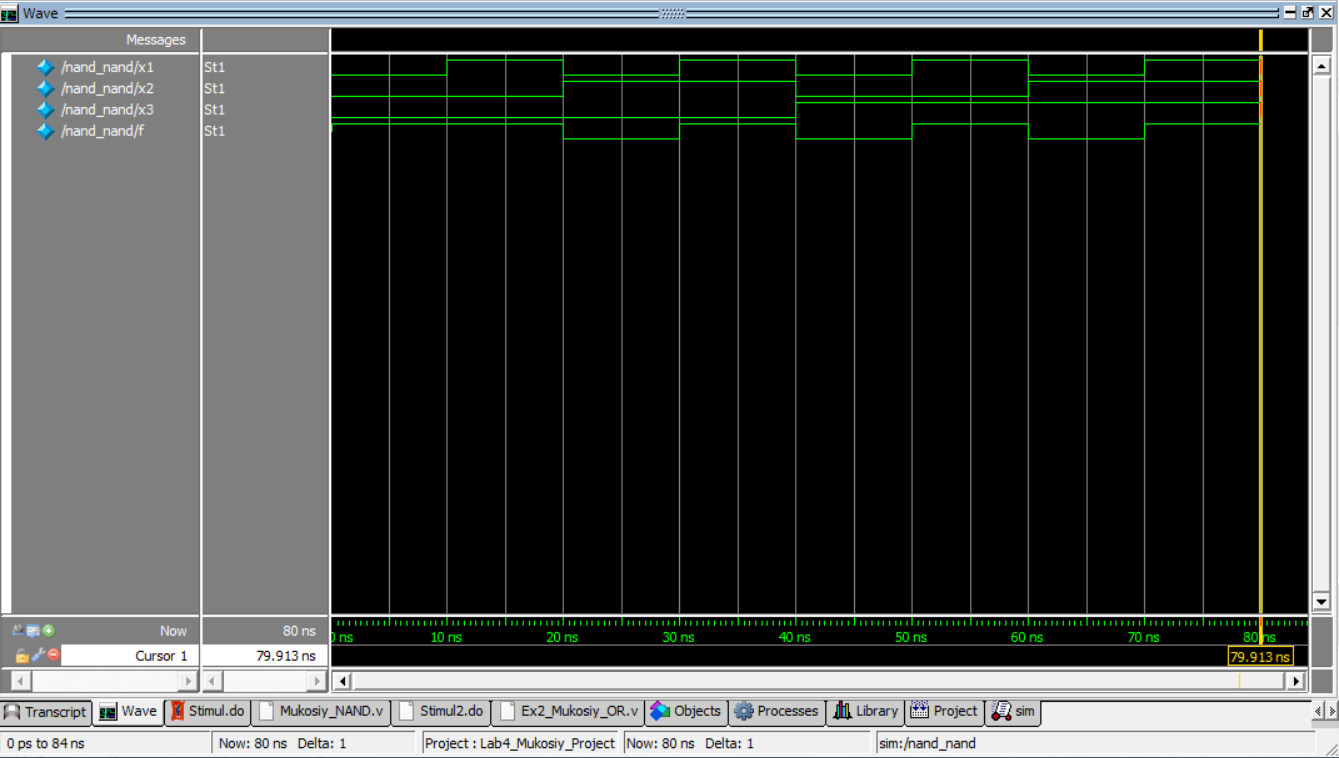


Рисунок 18.

Порівнявши вихідні сигнали (рис. 18) та таблицю істинності (табл. 3) можна дійти висновку, що схема спроєктована вірно.

1. Створюю новий файл на мові Verilog з описом роботи пристрою в базисі I, І-НЕ (рис. 19).

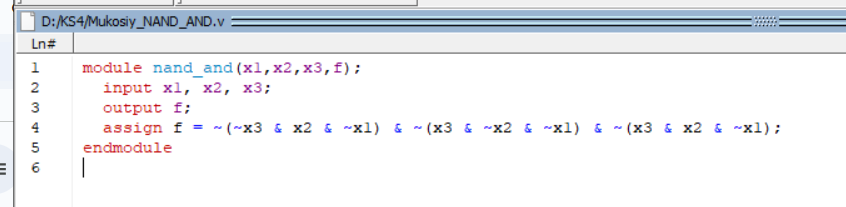


Рисунок 19.

1. Компілюю проєкт та створюю файл із вхідними тестовими сигналами (рис. 20).

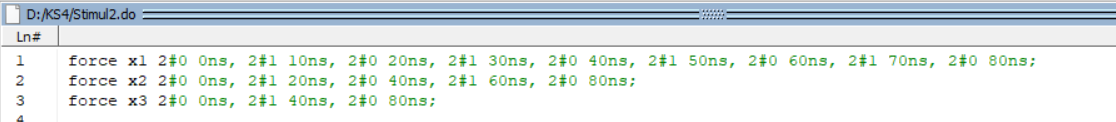


Рисунок 20.

1. Підключаю тестовий файл до проєкту в меню Tools, вибираючи команду TCL – Execute Macro та вказую файл Stimul2.do. Запускаю моделювання з меню Simulate командою Run – Run–All. Змінюю масштаб діаграми та переглядаю результат (рис. 21).

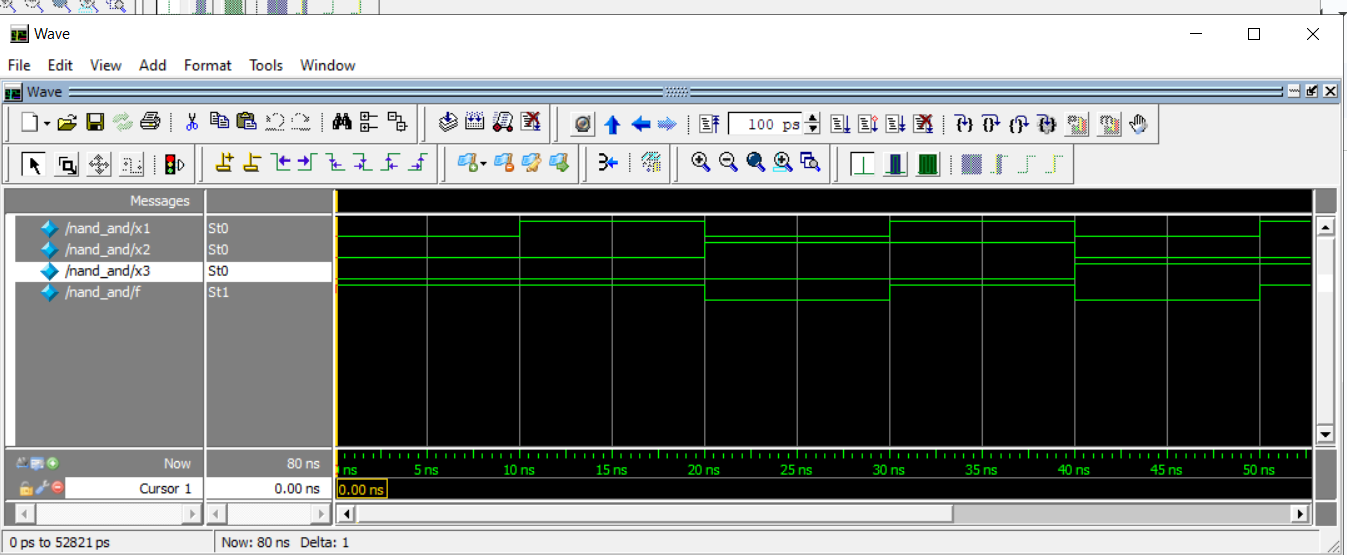


Рисунок 21.

Порівнявши вихідні сигнали (рис. 19) та таблицю істинності (табл. 3) можна дійти висновку, що схема спроєктована вірно.

# Висновки:

У ході виконання цієї лабораторної роботи було промодельовано роботу комбінаційної логічної схеми в середовищі ModelSim для заданої логічної функції. Було реалізовано дві різні схеми: одна побудована виключно на логічних елементах І-НЕ, інша – з використанням елементів І-НЕ та АБО. Після подання на входи однакових комбінацій сигналів та запуску моделювання було отримано часові діаграми вихідного сигналу f4. Аналіз результатів показав, що обидві реалізації функціонують однаково та видають ідентичні значення виходу в усіх тестових ситуаціях. Це підтвердило правильність побудови обох схем та еквівалентність їх логічної поведінки. Проведене моделювання дозволило переконатися, що задану логічну функцію можна реалізувати різними способами, використовуючи різні елементи логічного базису, а також дало змогу закріпити навички роботи з системою ModelSim та аналізу роботи цифрових схем.