Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота № 6**

з дисципліни «Комп’ютерна схемотехніка»

**«ПРИСТРОЇ ДЛЯ ПЕРЕТВОРЕННЯ ЧИСЕЛ.**

**ТИПОВІ ВУЗЛИ КОМП’ЮТЕРА»**

Виконав:

 студент групи ІО-32

Душко Р.В.

Номер залікової книжки: 3206

Перевірив:

Викладач Нікольський С.С.

Київ 2025 р.

**Лабораторна робота № 6**

**Тема**: Пристрої для перетворення чисел. Типові вузли комп’ютера.

**Мета роботи**: Ознайомитись з методами побудови арифметичних комбінаційних пристроїв, зокрема багаторозрядних суматорів. Реалізувати 6-розрядний суматор двома способами — структурно, через каскадування однорозрядних суматорів, та поведінково — з використанням оператора додавання.

**Хід роботи**

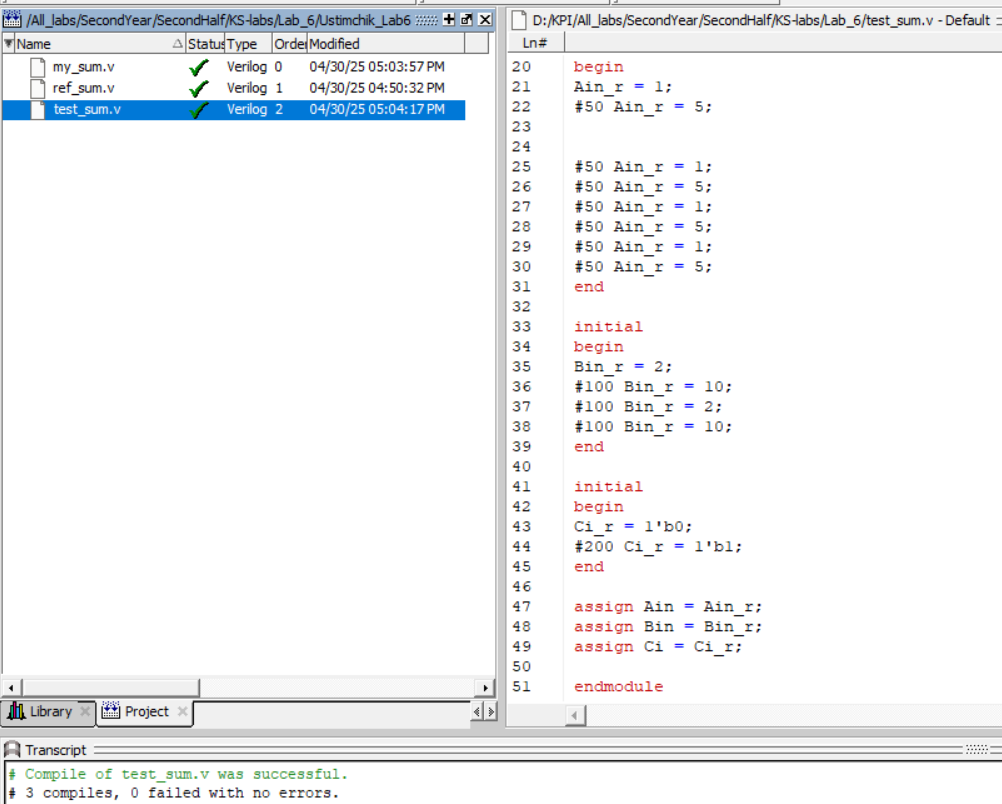
Варіант:

320610 – 1100101001102, звідси :

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

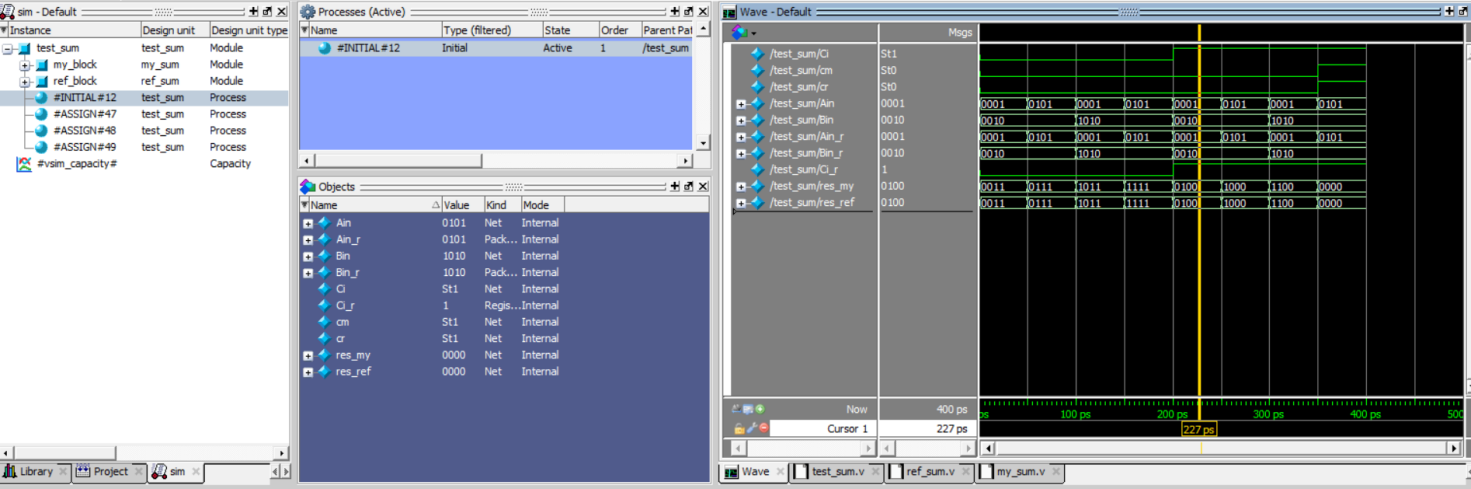
|  |  |
| --- | --- |
| h3 h2 h1 | Розрядність суматора |
| 1 1 0 | 7 |

Створюємо проект, а також файли на мові Verilog відповідно методички. Прописуємо код та проводимо компіляцію.

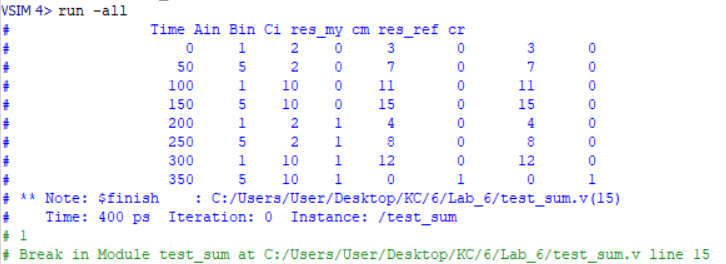


Компіляція пройшла успішно

Перейдемо у режим моделювання :

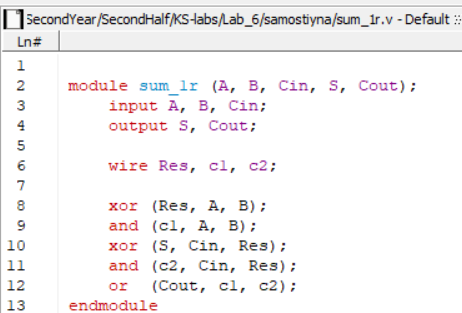


Результати з консолі:

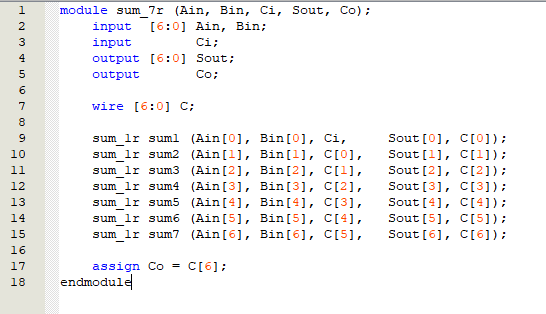


Ми бачимо, що на входи суматора подаються операнди Ain, Bin та сигнал переносу Ci. Результат додавання виводиться на res\_my — це результат нашого суматора, і cm — сигнал переносу з нього. Також паралельно працює модуль ref\_sum, який реалізований на поведінковому рівні, і виводить res\_ref та cr.Під час симуляції бачимо, що для кожної комбінації входів результати res\_my і res\_ref повністю збігаються, як і сигнали переносу cm і cr. Це підтверджує, що реалізація мого суматора my\_sum працює правильно. Симуляція виконана коректно.

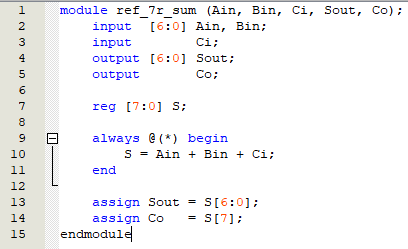
Створимо новий проєкт та створимо там файл на мові Verilog та напишемо там код для 7-ти розрядного суматора. За основу візьмемо код, який використовувався у прикладі вище . У результаті маємо отакий код:



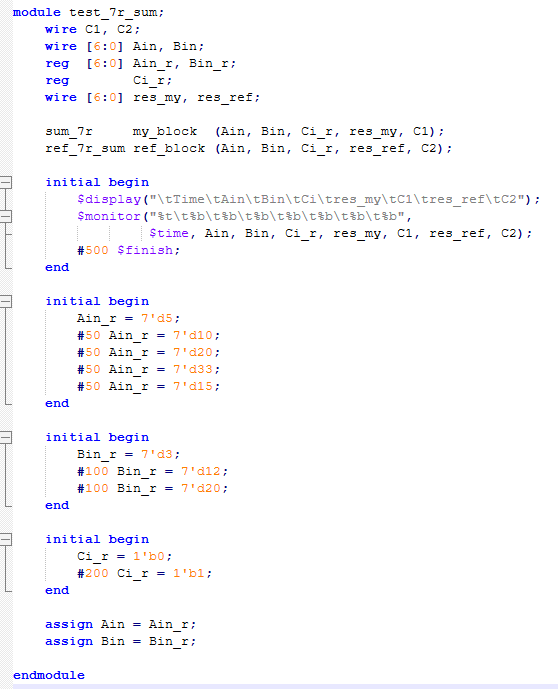
У цьому модулі реалізовано логіку повного однорозрядного суматора з урахуванням вхідного переносу. Сума обчислюється через два XOR, а перенос — через два AND і один OR. Цей модуль є базовим елементом для побудови багаторозрядного суматора.



У цьому модулі реалізовано 7-розрядний суматор на основі послідовного з'єднання 7 повних однорозрядних суматорів (sum\_1r). Кожен розряд враховує перенос із попереднього, що дозволяє точно передавати перенесення по всьому розряду. Структурна побудова дозволяє краще уявити внутрішню логіку додавання.

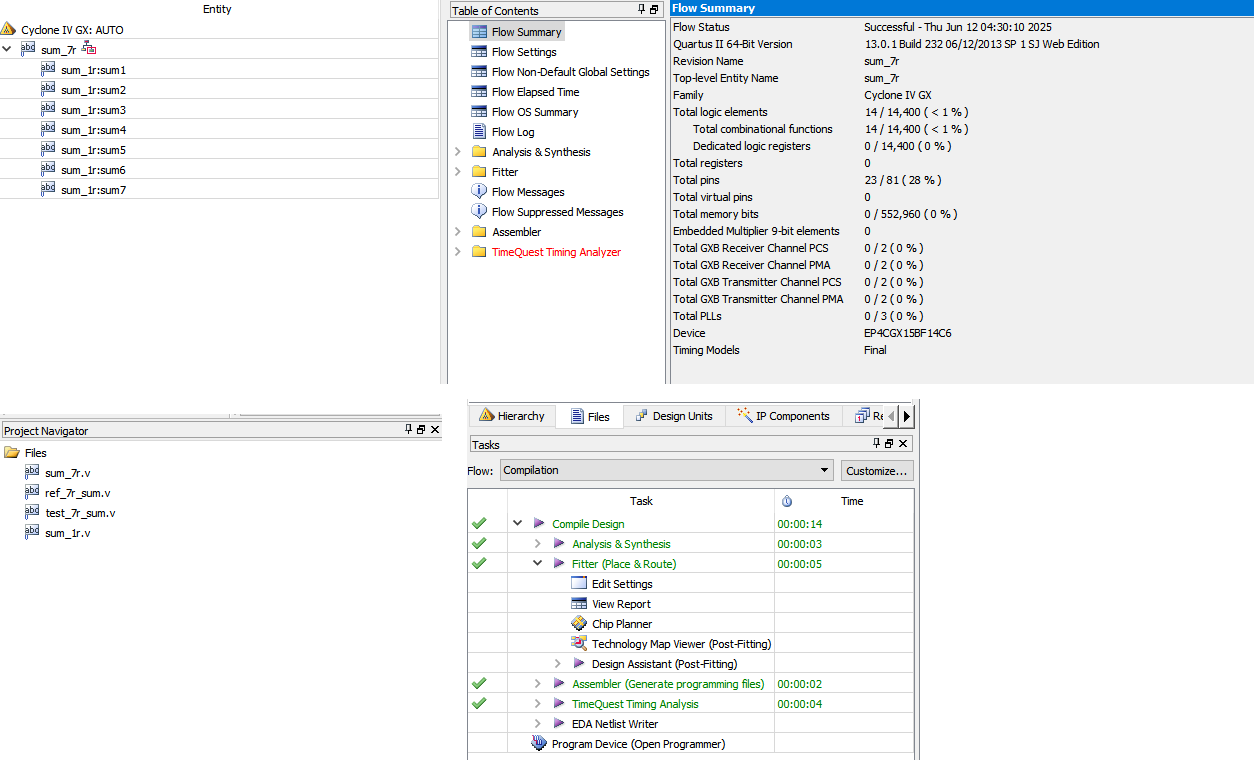


У даному модулі використовується поведінковий опис суматора, де результат обчислюється за допомогою оператора +. Це дозволяє швидко і просто реалізувати додавання з урахуванням переносу, без логічної деталізації. Такий модуль зручно використовувати як еталон для перевірки.

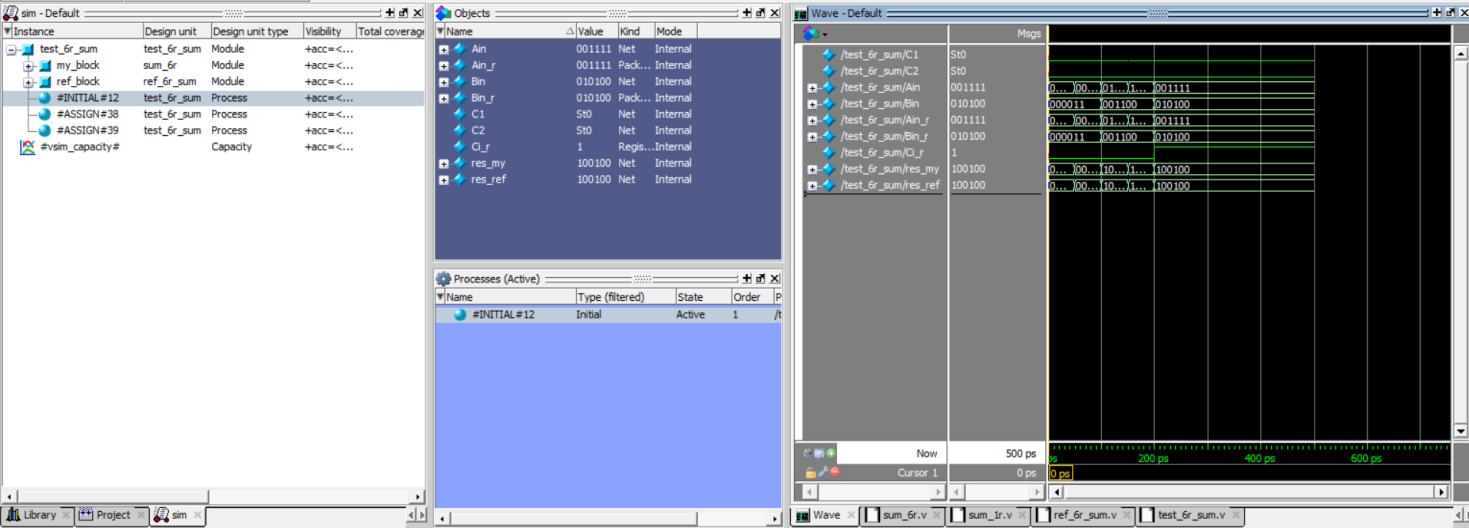


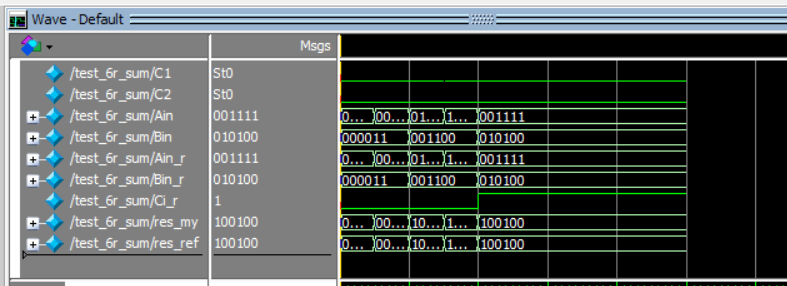
У модулі тестбенча реалізовано перевірку роботи суматора шляхом подачі різних комбінацій вхідних значень Ain, Bin та Ci. Результати виводяться в консоль через $monitor. Порівняння результатів структурної та поведінкової реалізацій дозволяє впевнено перевірити правильність роботи власного суматора.

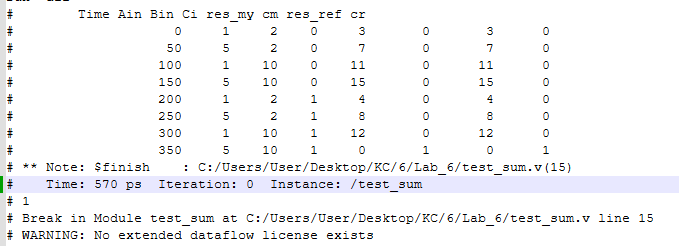
Тепер скомпілюємо всі ці файли .



Перейдемо у режим модулювання та перевіримо чи правильні результати:







Все виконано вірно, відповідно до теоретичних даних (вимог).

**Висновки:** У даній лабораторній роботі було реалізовано 7-розрядний суматор двома способами: структурно та поведінково. Структурна модель (sum\_7r) була побудована з семи однорозрядних повних суматорів (sum\_1r), з’єднаних послідовно. Поведінкова реалізація (ref\_7r\_sum) виконувала додавання за допомогою оператора "+". Для перевірки обох підходів було створено тестовий модуль (test\_7r\_sum), який подавав різні вхідні дані та порівнював результати. Симуляція у ModelSim показала повний збіг результатів, що підтверджує правильність роботи структурної моделі.

**Посилання на git\_hub репозиторій:**

https://github.com/Romchik235/Circuit-Design/tree/main/Lab-6