**Лабораторна робота №2**

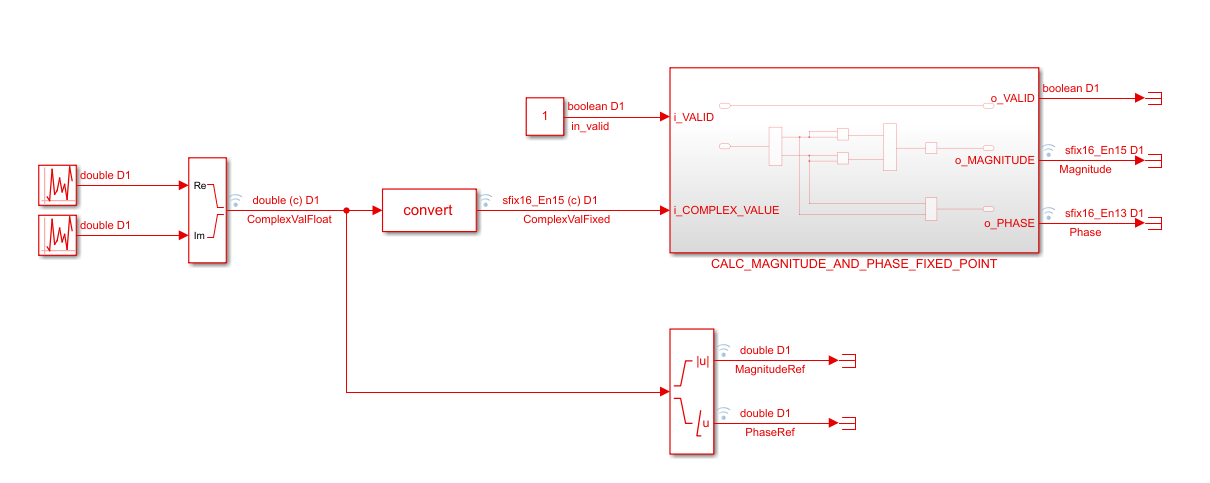
**Вакуленко Максим ДК-02 Варіант 16**

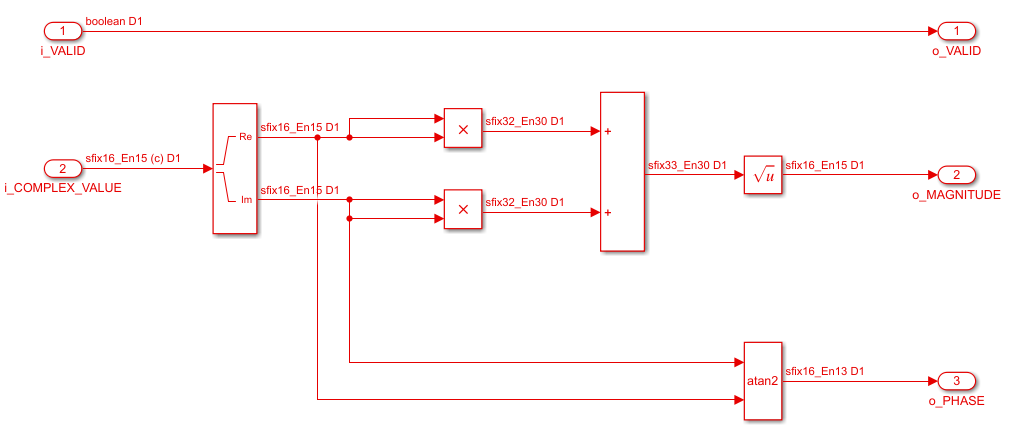
**Мета**

В Simulink реалізувати підсистему, що розраховує модуль і аргумент комплексного числа для вхідних даних у форматах з фіксованою комою і плаваючою комою

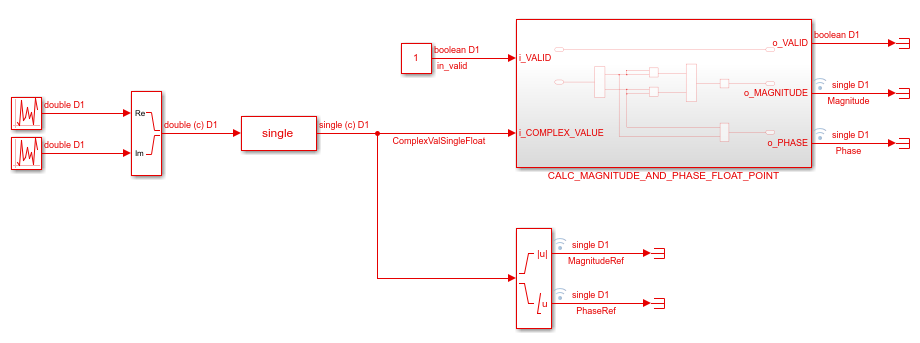
**Загальна інформація**

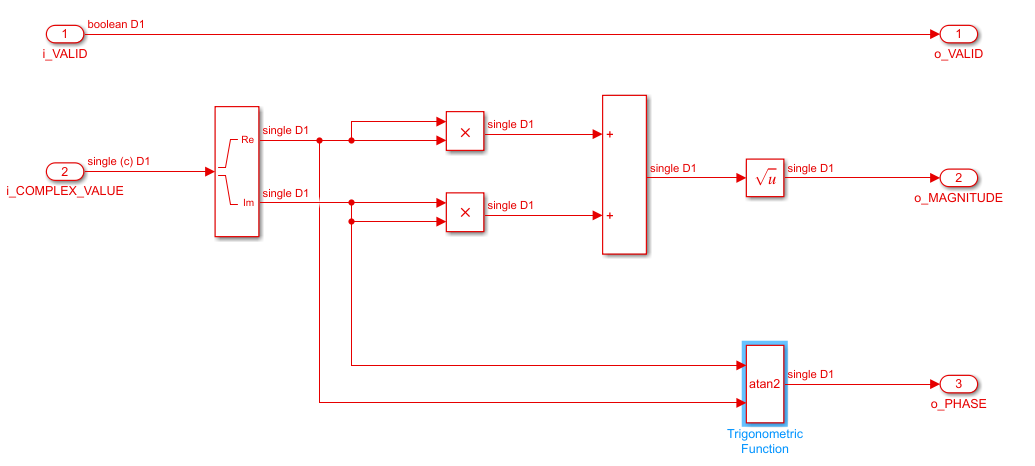
Блок-схема для вхідних даних з фіксованою комою:





Блок-схема для вхідних даних з плаваючою комою:





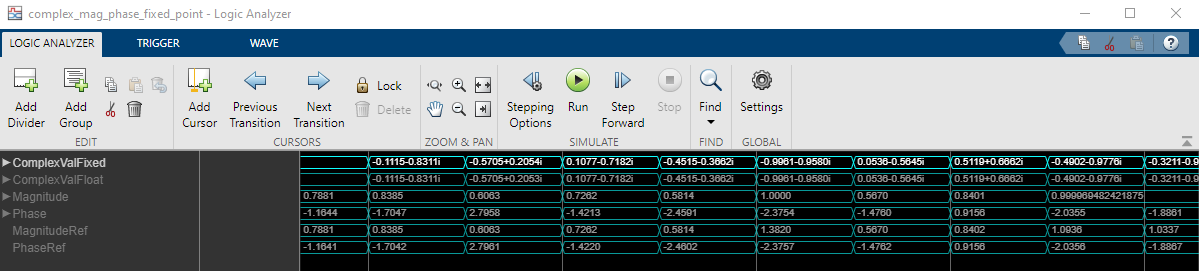
**Методика вибору налаштувань fixed point типу даних для представлення комплексного вхідного аргументу у форматі з фіксованою комою залежно від номеру варіанту**. Індивідуальний номер варіанту наведено в кінці документу. Для парних номерів варіантів представлення беззнакове. Для непарних номерів варіантів представлення знакове. Розрядність цілої частини N = K. Розрядність дробової частини M = 32 - K. Де K - номер варіанту. Налаштування fixed point типів даних на виході: для виходу o\_PHASE розрядність цілої частини 3 біта, розрядність дробової частини 13 біт, для виходу o\_MAGNITUDE розрядність цілої частини N+1 біт, розрядність дробової частини M біт, де N та M визначені раніше розрядності цілої і дробової частин fixed point типу вхідного аргументу i\_COMPLEX\_VALUE.

У якості джерела даних використати блоки “Uniform Random Number” з приведенням результату до типу з фіксованої комою заданого формату, або типу з плаваючою комою одинарної точності (float single), залежно від моделі, яка будується в simulink. У якості параметра seed для першого блоку “Uniform Random Number” використати номер варіанту. Для кожного наступного блоку “Uniform Random Number” збільшувати значення seed на 1. Діапазон значень, які видає блок “Uniform Random Number” повинен дорівнювати діапазону значень, який може бути представлений у форматі з фіксованою комою для кожного варіанту. У звіті необхідно обгрунтувати/вивести граничні значення діапазону чисел, які видає блок “Uniform Random Number”.

**Завдання:**

1. В Simulink побудувати блок схеми обчислювачів модуля і аргументу комплексного числа для вхідного аргументу з фіксованої комою і плаваючою комою. Обчислювачі для вхідних даних з фіксованою комою і плаваючою комою будувати в окремих моделях Simulink.
2. Для моделі обчислювача з вхідними даними у фіксованій комі та для моделі обчислювача з вхідними даними у плаваючій комі в логічному аналізаторі Simulink переглянути залежність від часу даних на вході обчислювача, а також даних на виході кожного обчислювача (розраховані значення модуля і аргументу комплексного числа) і еталонних значень результату (значення модуля і аргументу розраховані у блоці “Complex to Magnitude-Angle”). Переконатися, що еталонні значення результату або дорівнюють розрахованим значенням, або відрізняються на незначне значення похибки.

Типовий приклад результату:

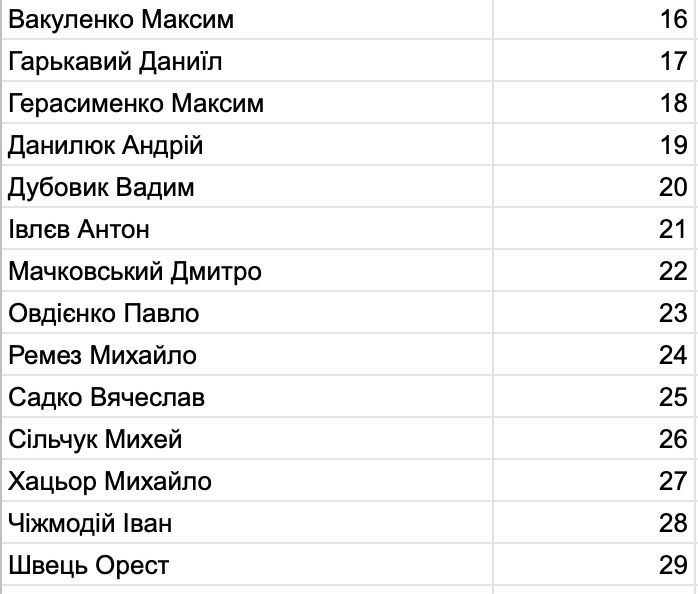


1. Створити звіт, в якому відобразити створені в Simulink блок схеми (з відображенням типів даних та з відображенням вмісту підсистем) і результати моделювання для перших десяти комбінацій на входах. Приклади подання наведені вище.
2. Якщо додати у звіт згенерований код на Verilog та результат синтезу згенерованого коду в Quartus для створеної підсистеми (звіт по апаратним витратам, результат виклику RTL Viewer), можна отримати +2 додаткових бали.
3. Якщо створити тестбенч в Matlab для створеної підсистеми і додати в звіт результат симуляції тестбенча в Modelsim/Questasim, можна отримати +2 додаткових бали.
4. Завантажити звіт і файли (файли моделі та у випадку наявності файли згенерованого HDL коду, файли проекту Quartus та тестбенчу) в репозиторій студента на github. Датою завершення виконання роботи вважається дата завантаження файлів в репозиторій.

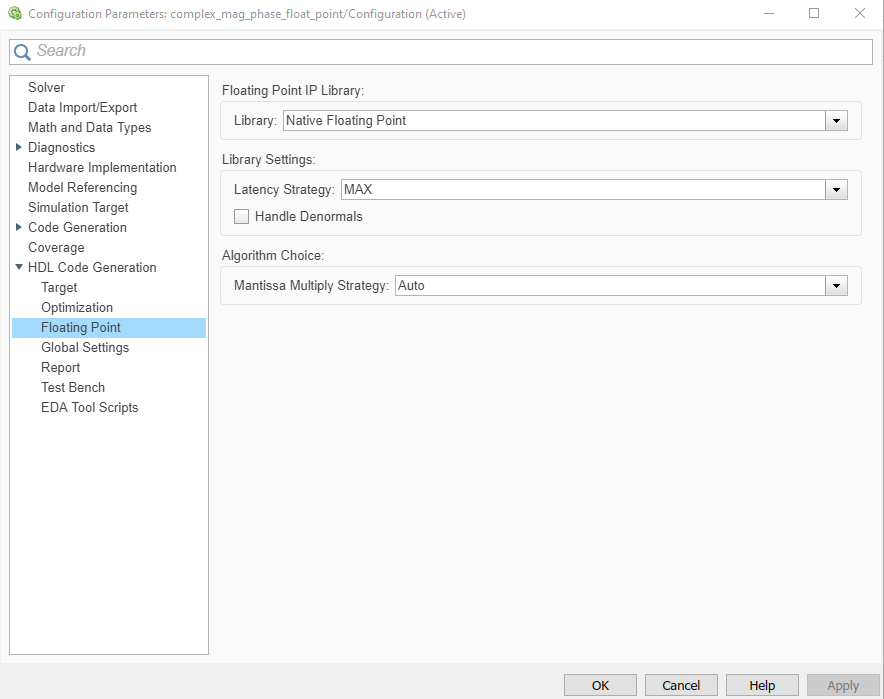
**Додатки**

Варіант:

Вакуленко Максим 16

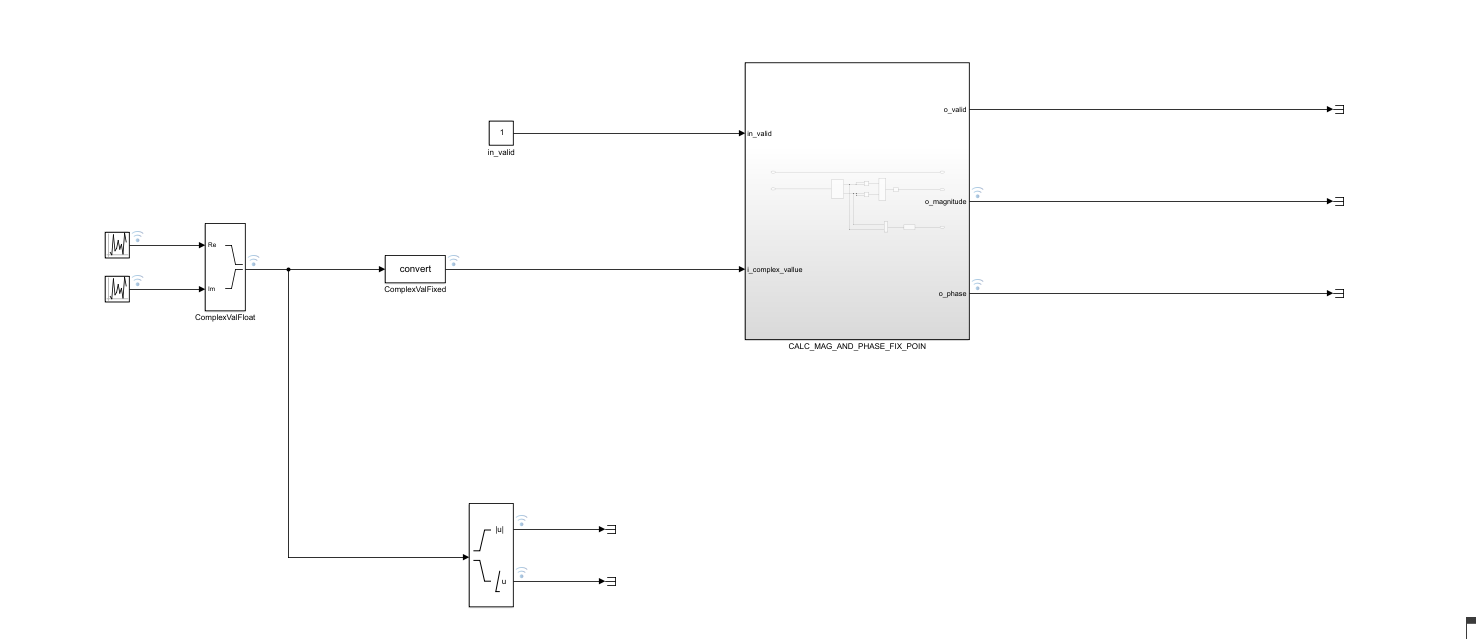


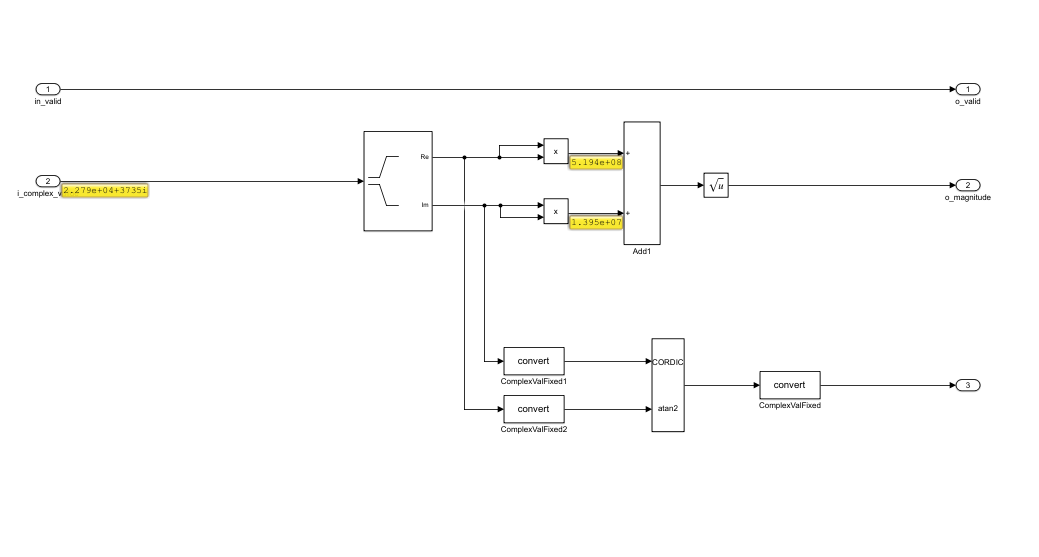
Налаштування для генерації HDL обчислювача з плаваючою комою:



Хід роботи:

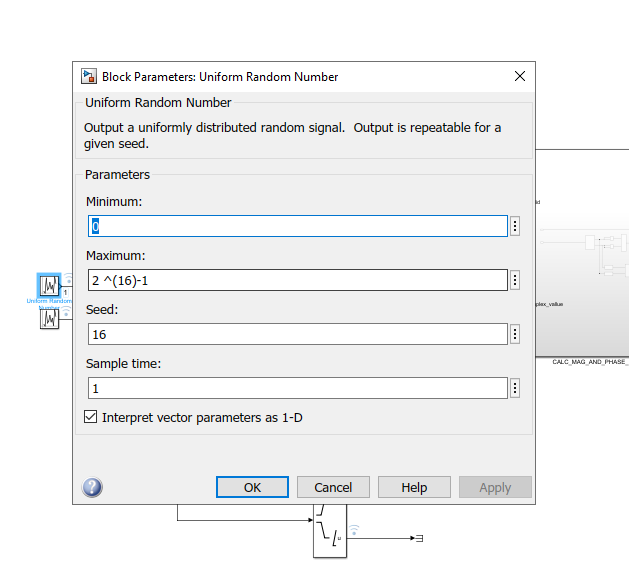
Побудована схема для чисел з фіксованою точкою:



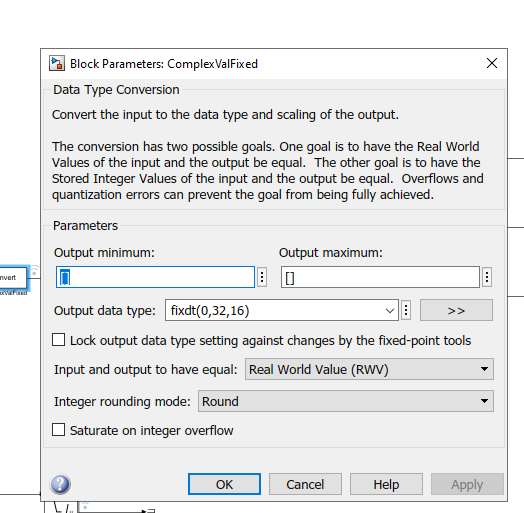
Підсистема для чисел з фіксованою точкою:

Налаштування блоку Uniform Random Number(беззнакові числа):

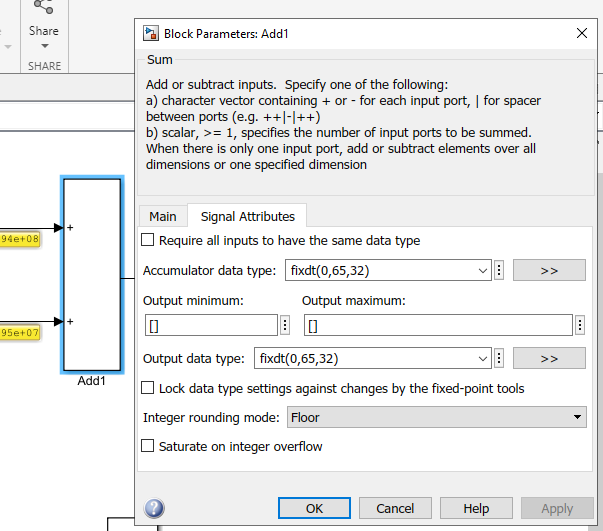
Оскільки у нас ціла та дробова частина має однакову розрядність 16 біт, то найбільше значення, яке приймає цілочисельне значення це 2^(16) -1. Найменше в свою чергу буде значення 0, як для беззакового формату. Важливо підмітити, що дробова частина у свою чергу приймає максимальне значення в (1 – h), де h це значення найменшої частини дробової частини, у нашому випадку це h = 2^(-16). Але для простоти запису було взято значення найбільшої цілої частини для генерації цисел.



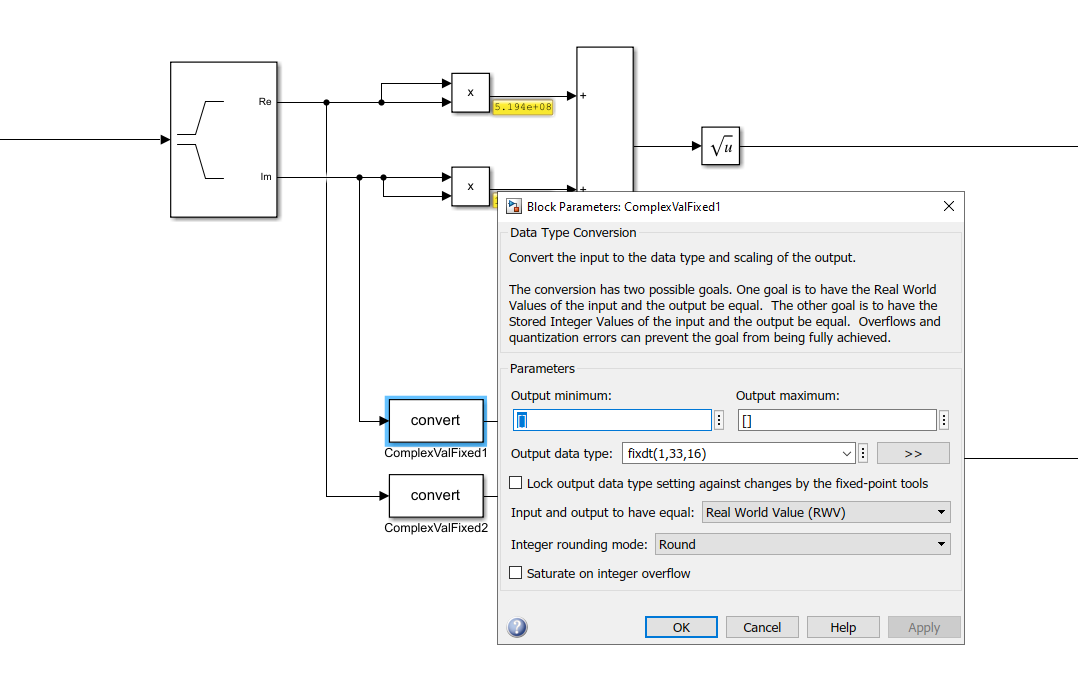
Налаштування блоку конвертації у комплексне число з фіксованою комою:



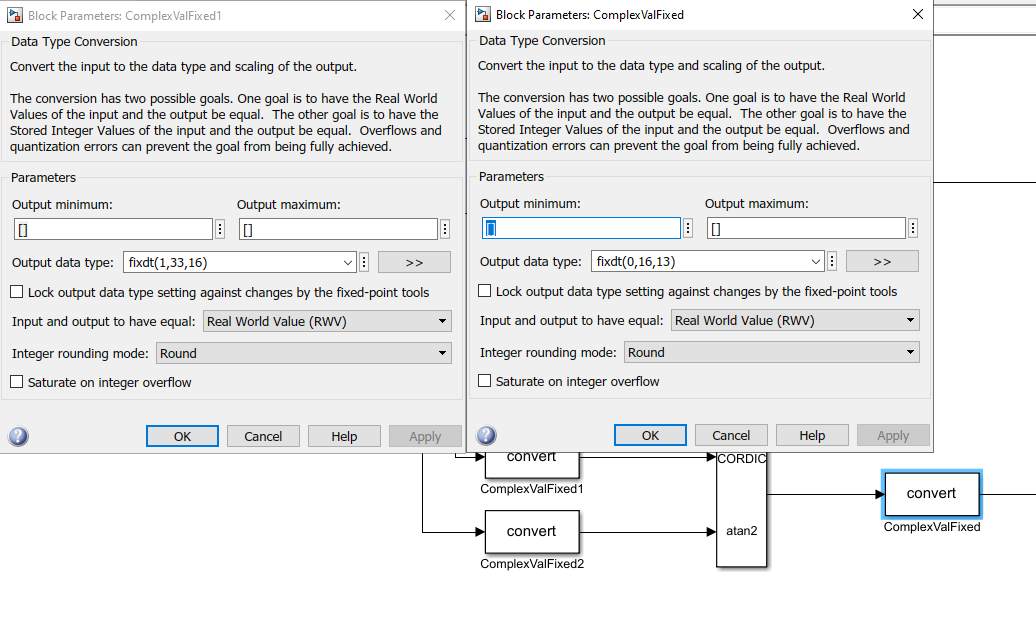
Налаштування блоку суми квадратів зі збільшенням розрядності на один:



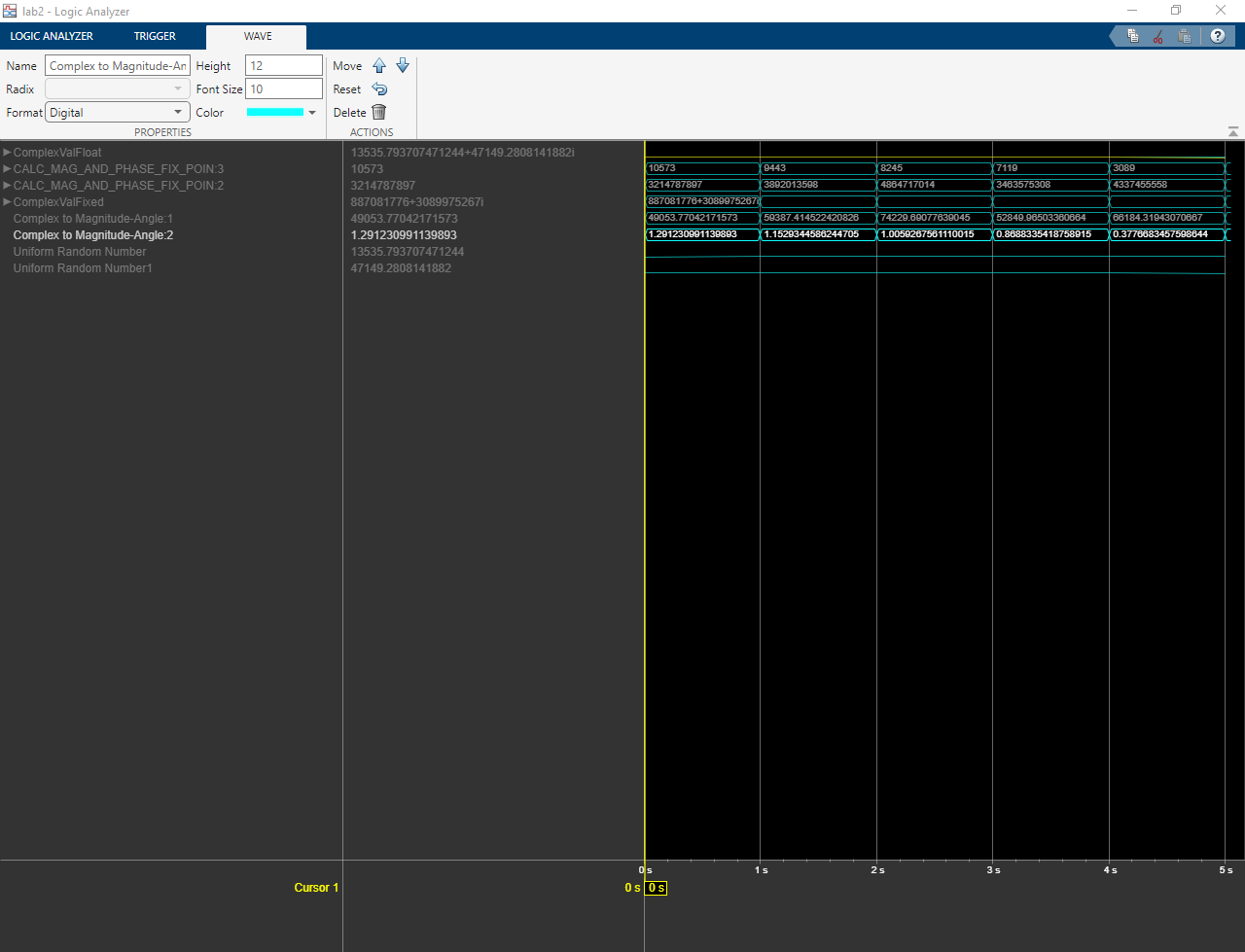
Оскільки для генерації коду потрібні вхідні дані на блок atan2 повинні бути знакові то ми розширили розрядність цілої частини на один для знакового біту і конвертували число.



Блоки конвертації на вході-виході atan2:

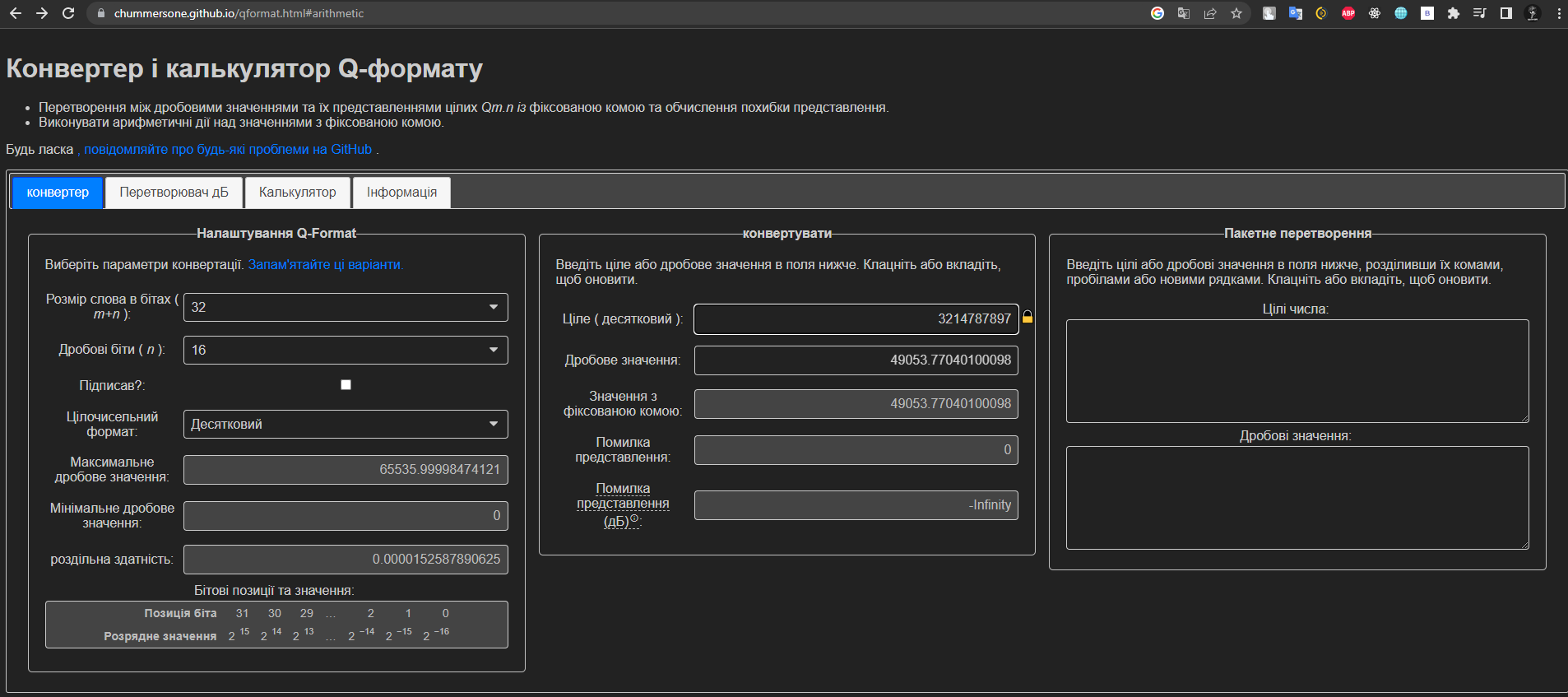


Отримані дані:

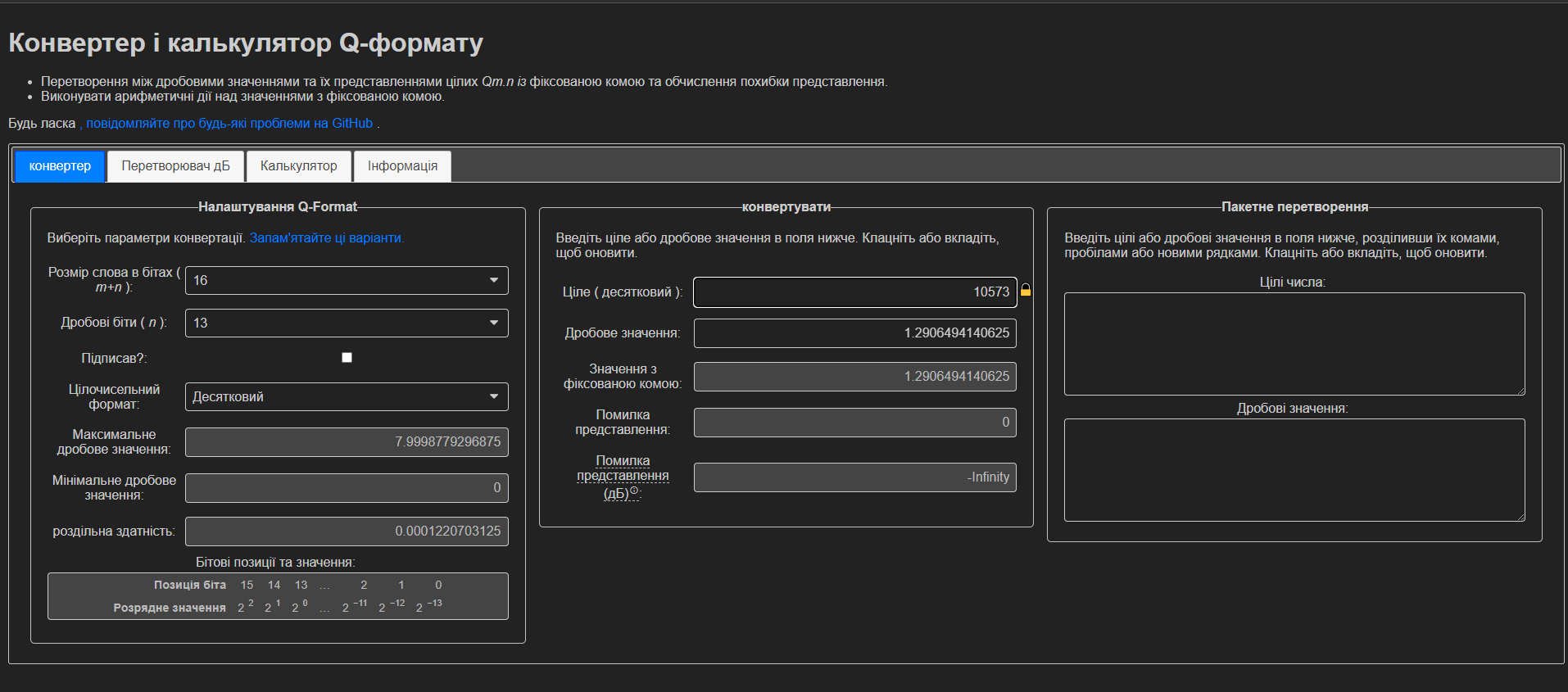


Явно перевести формат вихідних даних підсистеми у представлення з плаваючою крапкою не вийшло, тому скористємося калькулятором для порівняння:\

Magnitude:



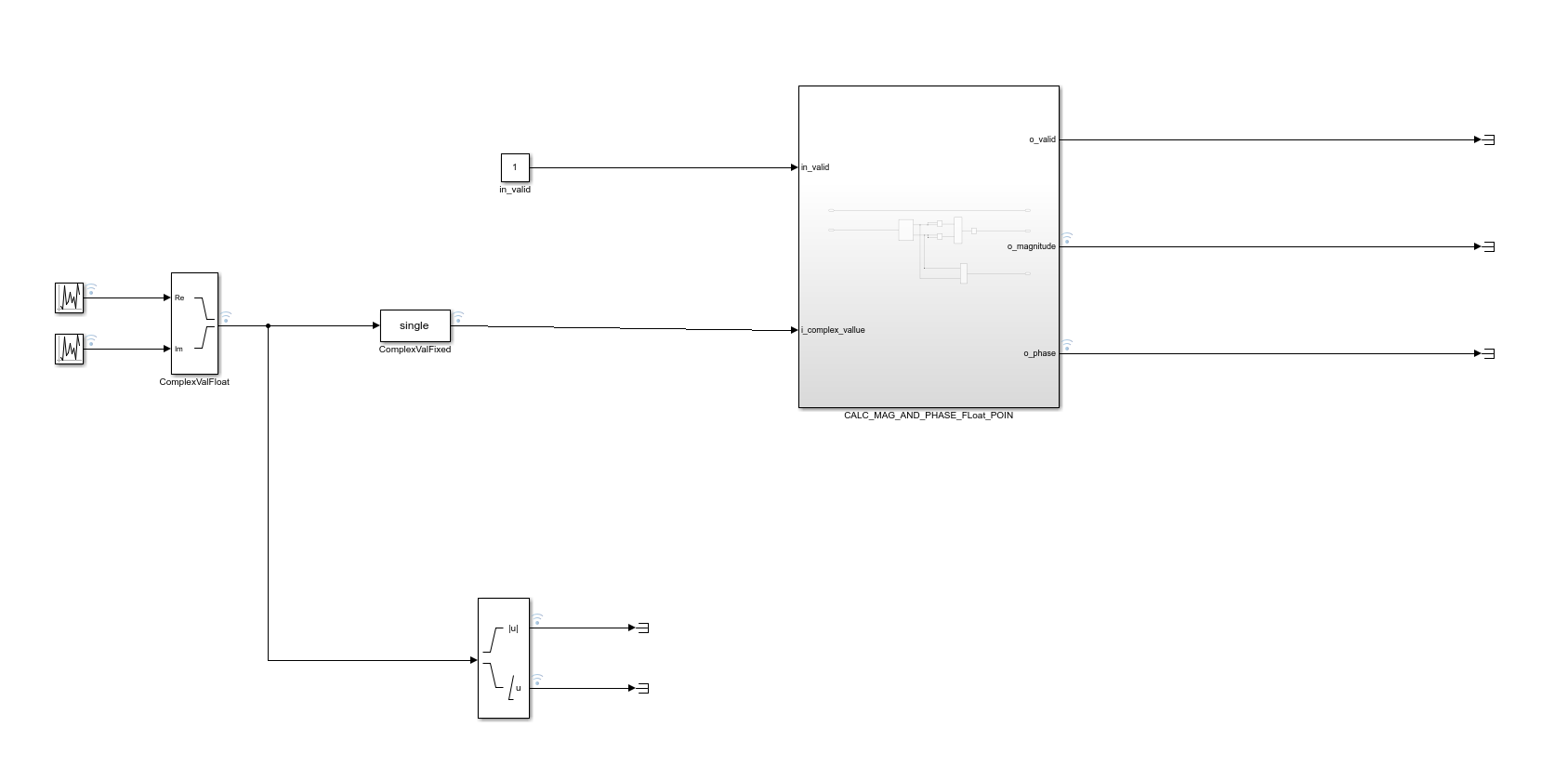
Phase:



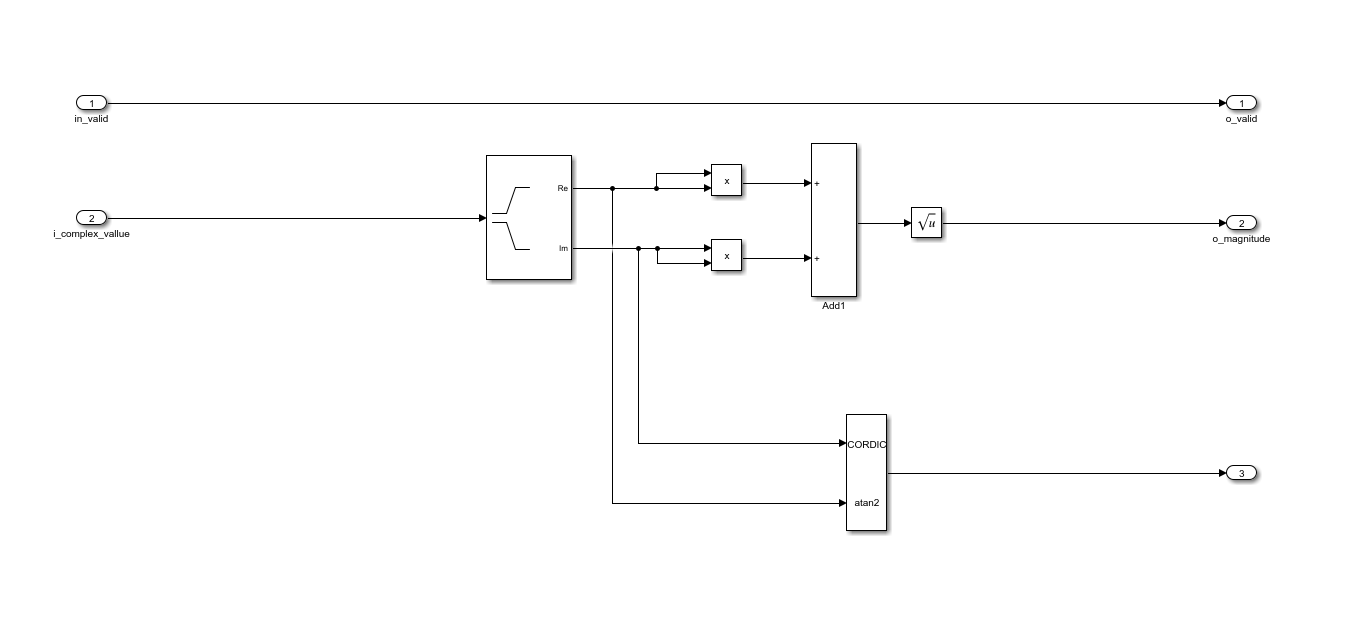
Як можна побачити дані майже однакові, відрізняються лише на невелику похибку.

**FLOAT PART:**

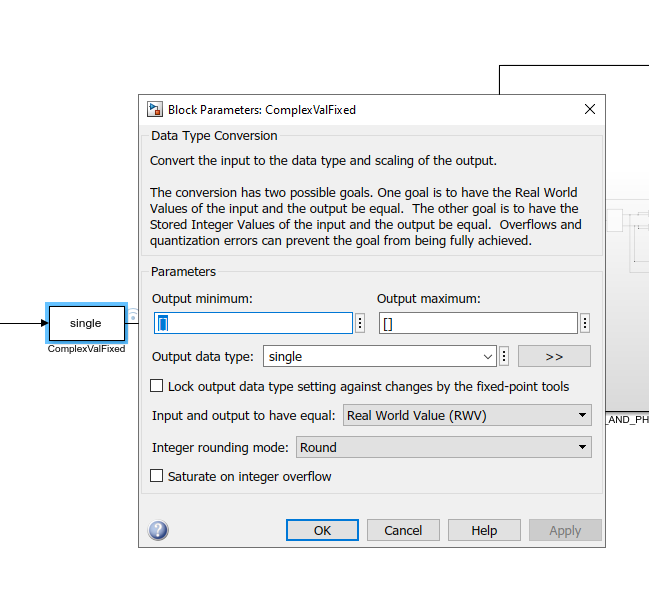
Побудована схема для чисел з плаваючою точкою:



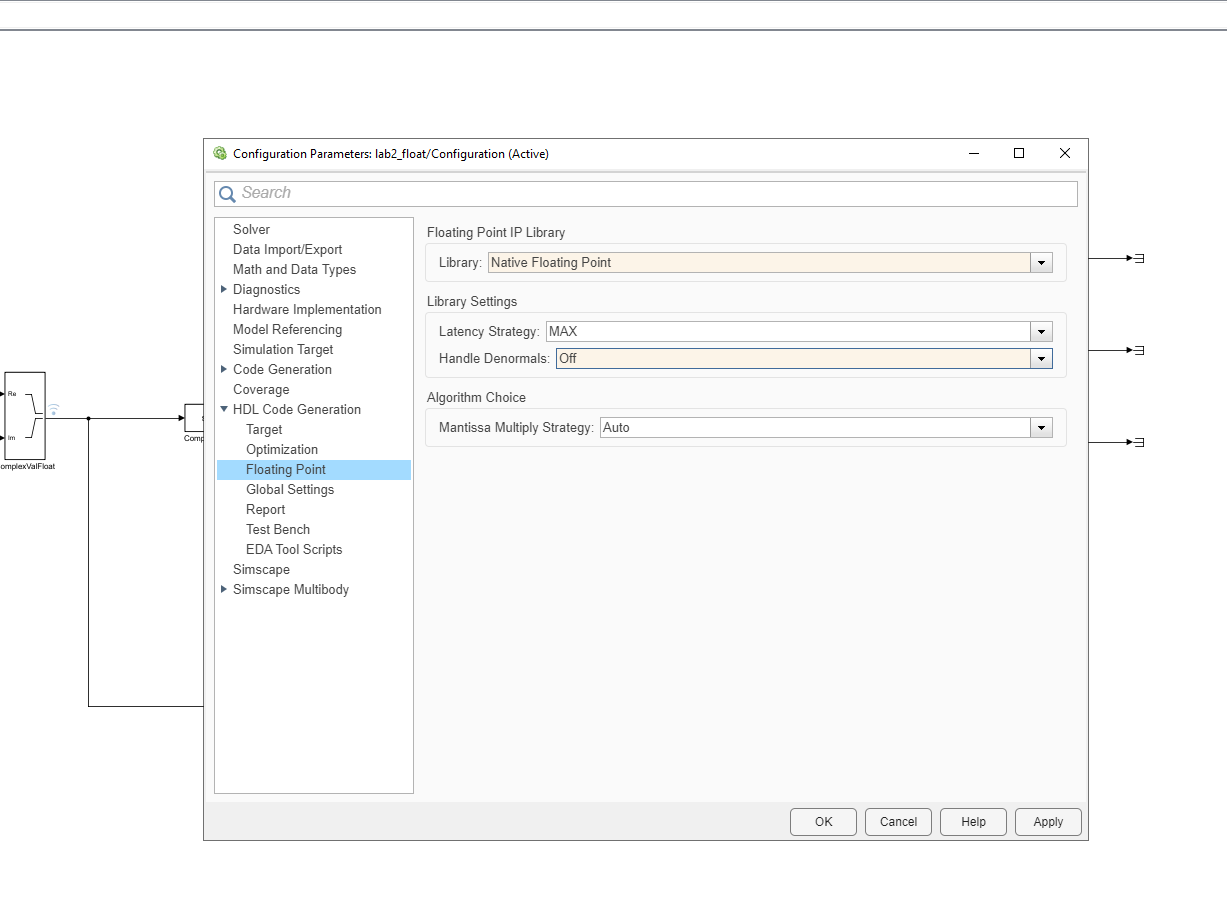
Підсистема для чисел з плаваючою точкою:

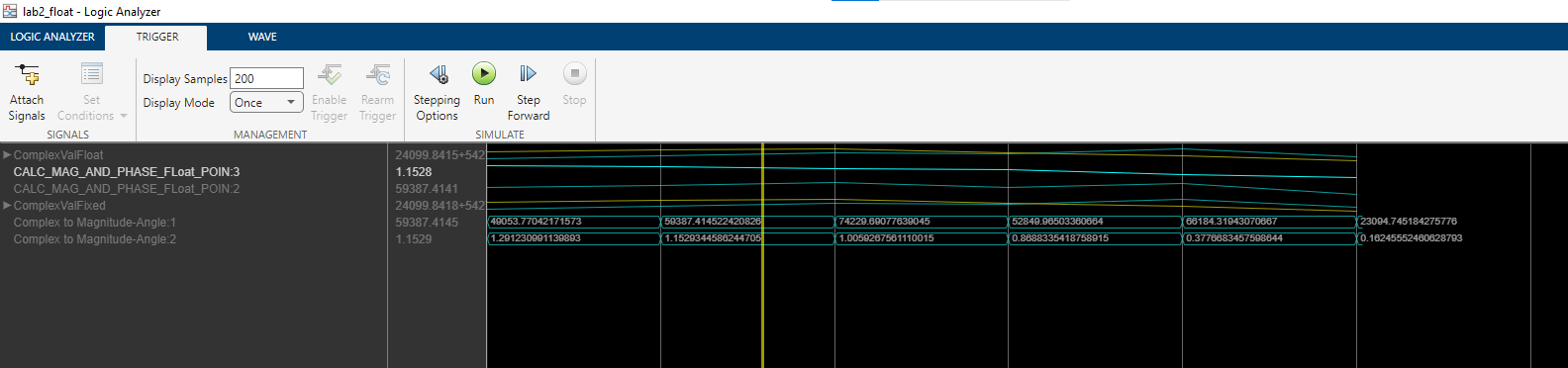


Конвертація у число в форматі fixed point



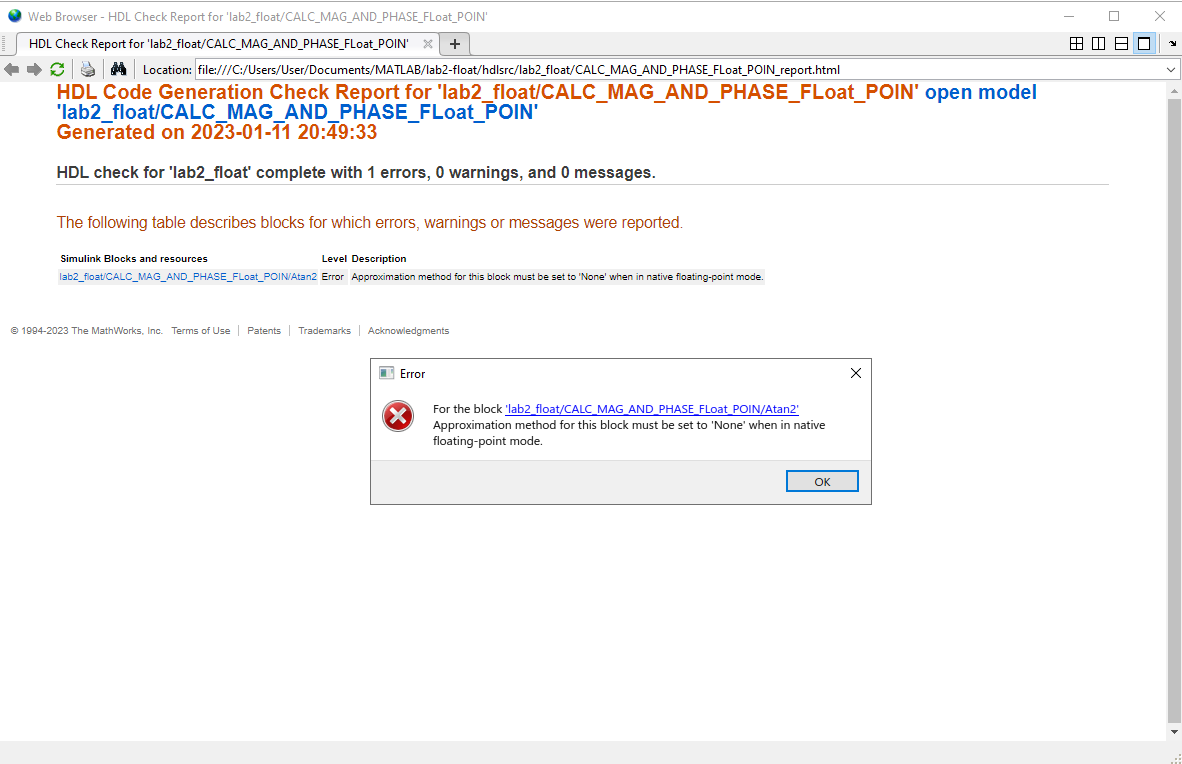
Налаштування генерації:



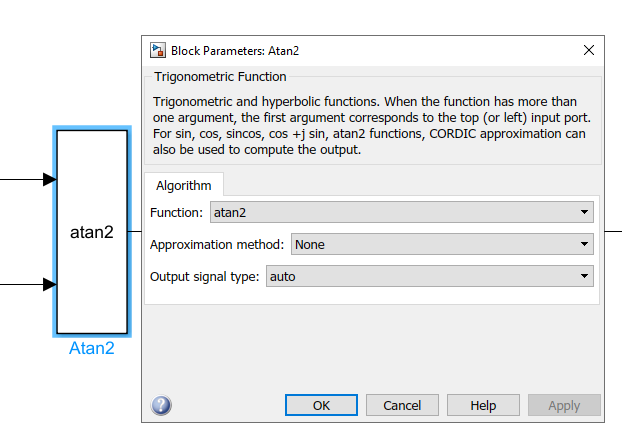
Дані симуляції:

Як можна побачити дані ідентичні, присутня лише невелика похибка у кінці.

При генерації коду для підсистеми виникла помилка.



Вирішення – вимкнення апроксимаційного методу



Висновок:

У ході цієї лабораторної роботи ми навчилися працювати з двома форматами представлення дробових чисел, перший – з фіксованою комою, другий – з плаваючою одинарної точності.

Побудували дві системи обрахунку модуля та фази для комплексних чисиле.

Для порівняння результатів був використаний готовий блок розрахунку (варто підмітити, що з uniform random number на готовий блок поступали числа у форматі double тобто з подвійної точності), як еталонний результат для порівняння.

Число у форматі single float виявилося найбличим етолонного через обмеження формату та округлення у блоках арифметики для підсистеми.