Міністерство освіти і науки України Національний технічний університет України «Київський Політехнічний Інститут імені Ігоря Сікорського» Кафедра конструювання електронно-обчислювальної апаратури

**Лабораторна робота №2**

**з дисципліни “ Апаратні прискорювачі ”**

**Виконав:** студент групи ДК-02 Дубовик В. Ю.

**Перевірив:** доц. Короткий Є В.

***Київ – 2023***

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

Досить актуальна задача, оскільки у сучасні інженерії (у тому числі в галузі телекомунікації, при цифровій обробці сигналу) використовуються комплексні числа

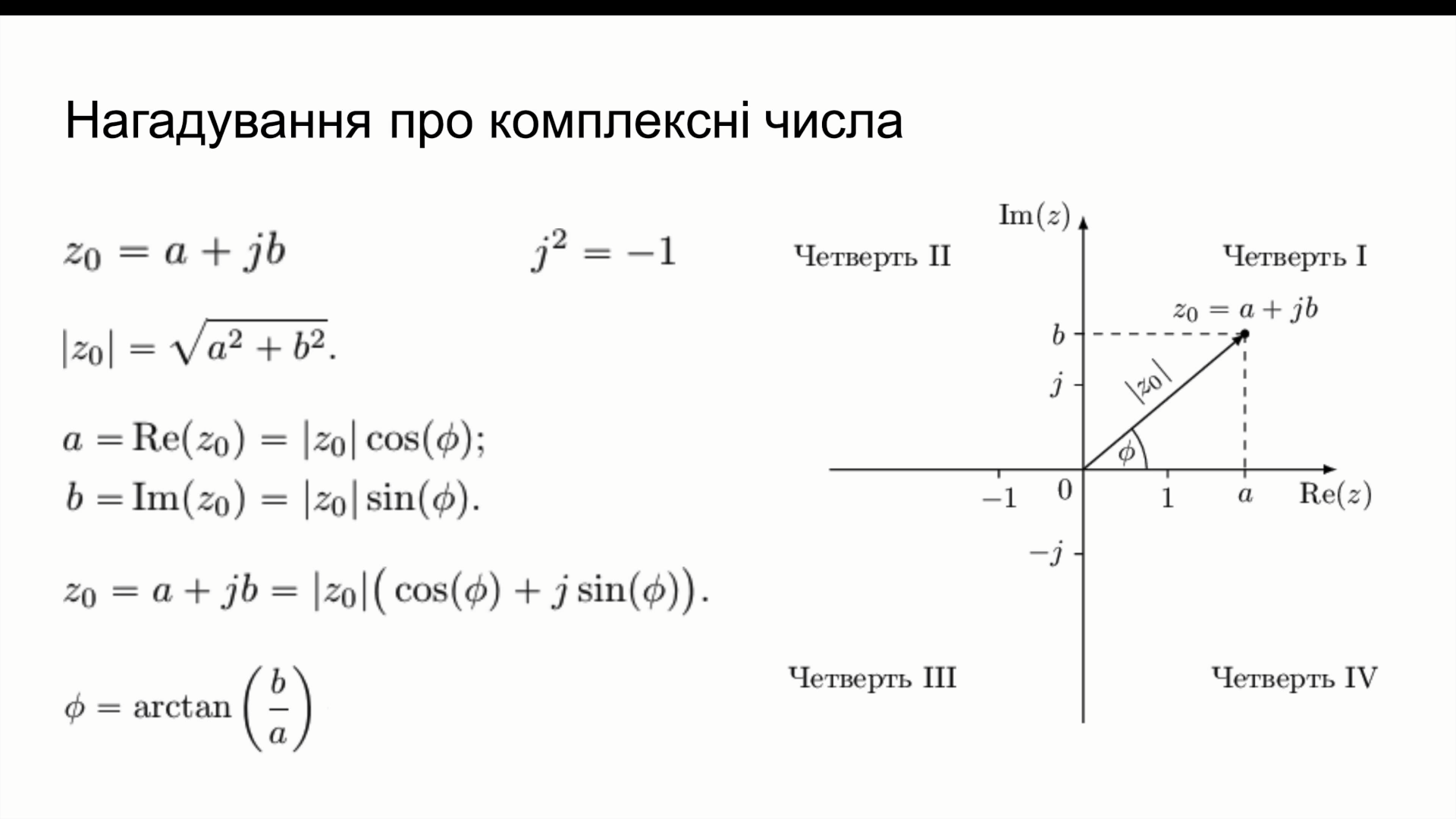
Індивідуальний номер: 20

Тоді розрядність цілої частини дорівнює 20, а розрядність дробової 12.

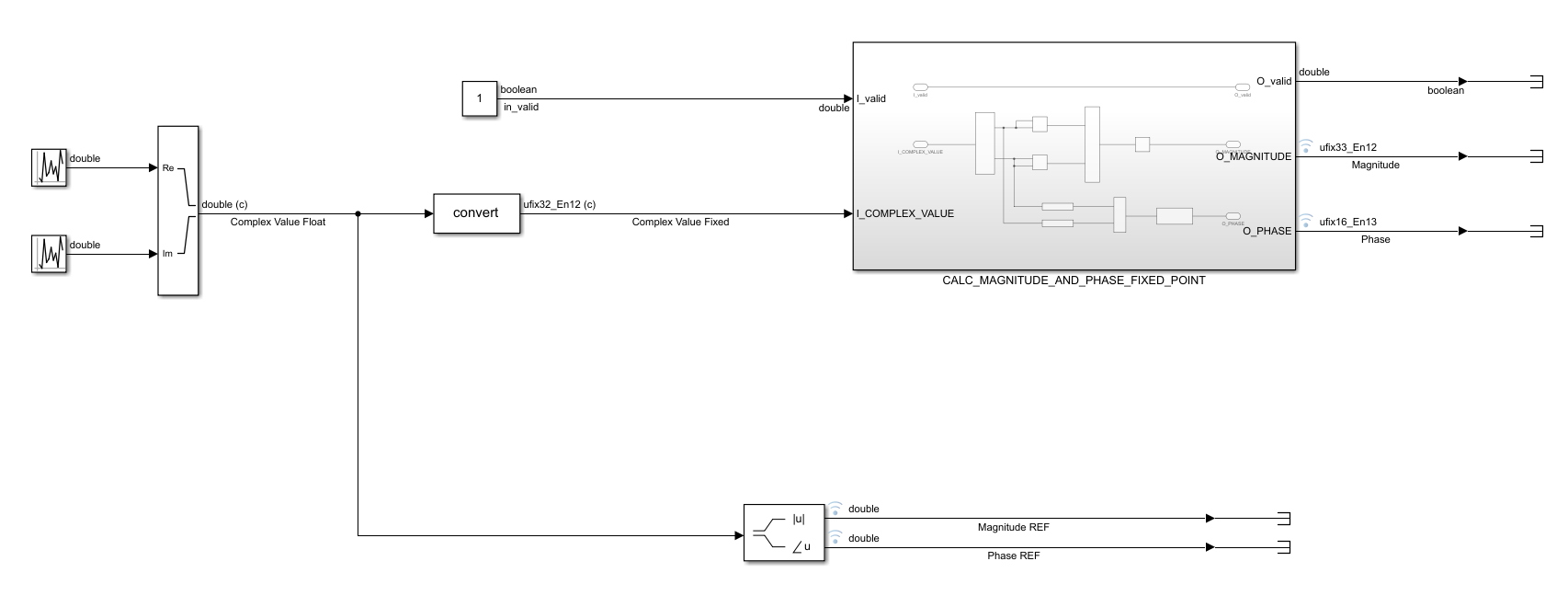
Оскільки номер парний, то представлення числа беззнакове.

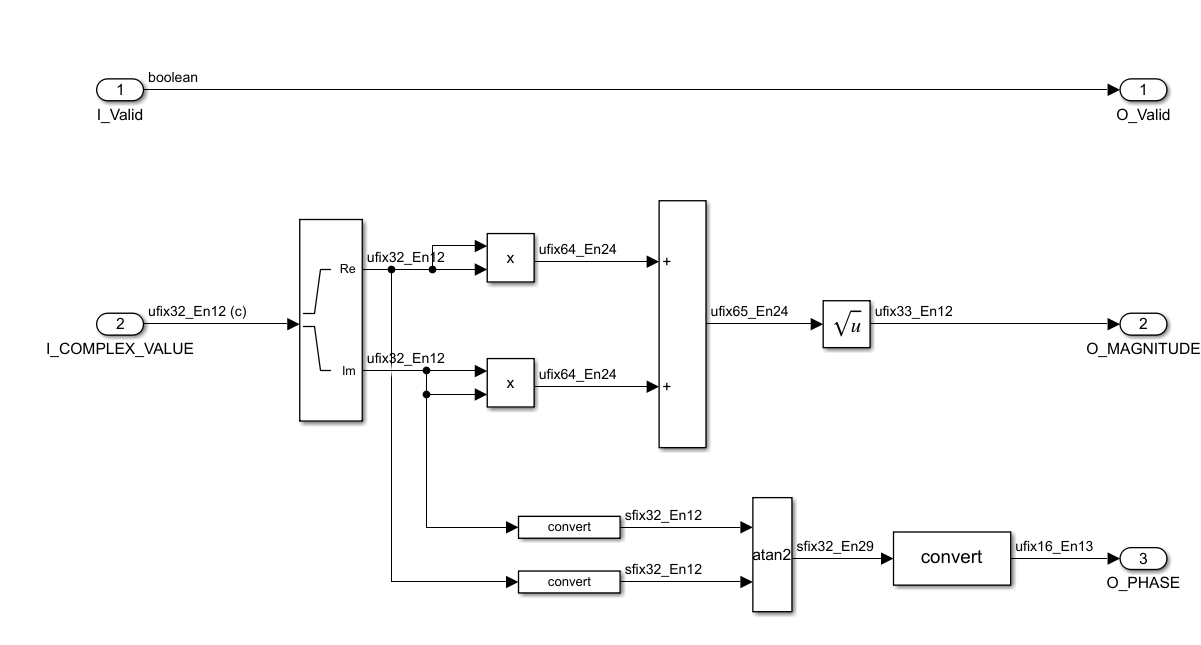
**Хід роботи**

Нагадування щодо комплексних чисел:



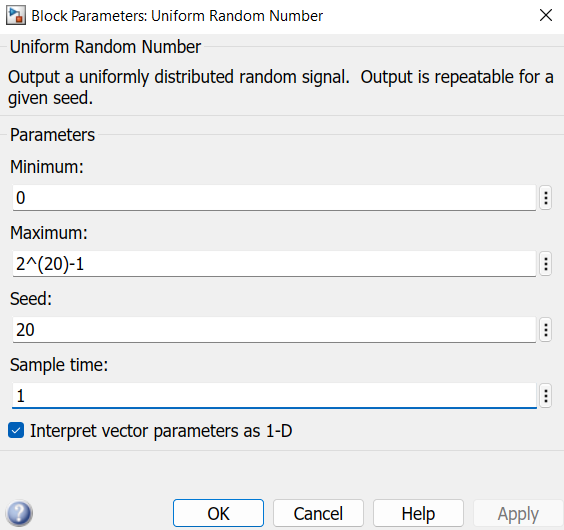
1. Побудуємо блок схеми обчислювачів модуля і аргументу комплексного числа для вхідного аргументу з фіксованої і плаваючою комою.

*Мал.1* схема обчислювача з фіксованою комою.

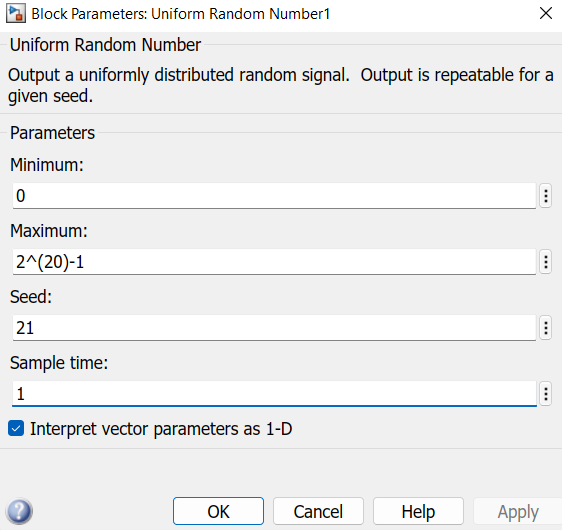


*Мал.2* Підсистема, де відбувається розрахунок модуля та аргументу комплексного числа

Далі розглянемо налаштування, які потрібно зробити для елементів даної блок-схеми:

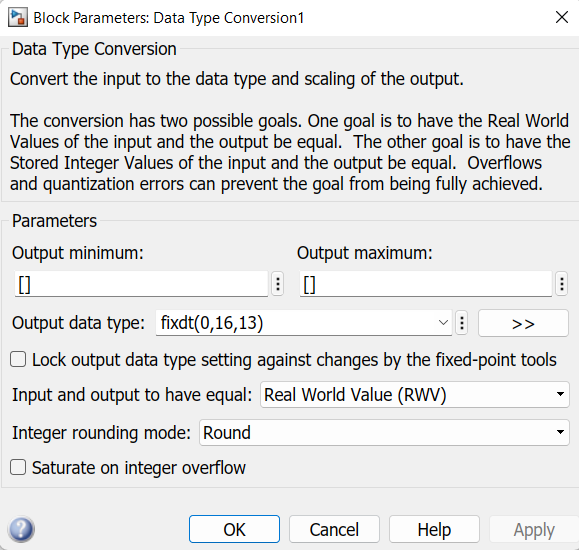


*Мал.*3 Налаштування Uniform Random Number 1.



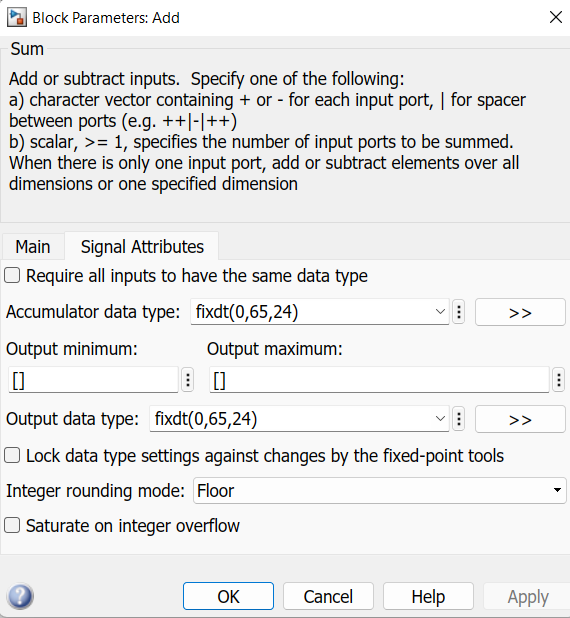
*Мал.*4 Налаштування Uniform Random Number 2.

У якості параметра SEED для першого блоку було використано номер варіанту, тобто 20. Для другого ж використовуємо номер варіанту + 1, тобто 21. Маємо беззнакове число, тоді мінімальне значення не від’ємне, а просто дорівнює нулю. Максимальне значення дорівнює 220 – 1.



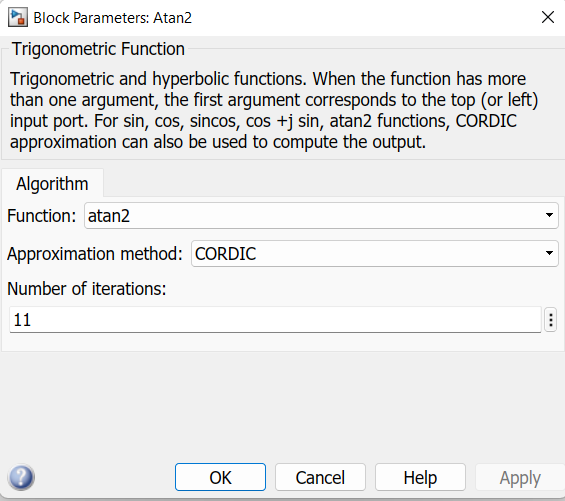
*Мал.*5 Налаштування Convert у підсистемі.

Згідно з умовами було поставлено розрядність цілої частини рівною 3, а дробової рівною 13. Отже маємо налаштування: 0 - беззнакове, 16 – загальна кількість розрядів, 13 – кількість розрядів відведена для дробової частини.

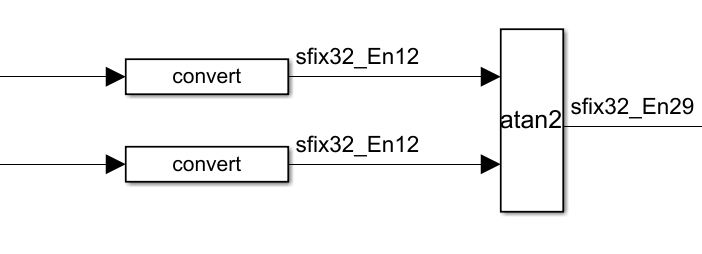


*Мал.*6 Налаштування блоку додавання.

Оскільки до цього виконувалося множення, то з 32 розрядів на вхід суматора приходить у двічі більше – 64 розряди. Під час операції додавання ж, розрядність цілої частини збільшується на 1, а дробове залишається таким самим. Отже на виході маємо 0-безнакове, 65 – розрядів цілої частини і 24 розряди дробової.

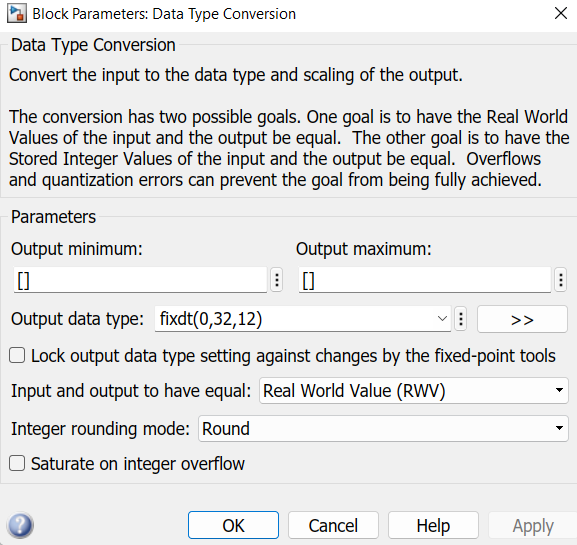


*Мал.*7 Налаштування блоку Atan2.

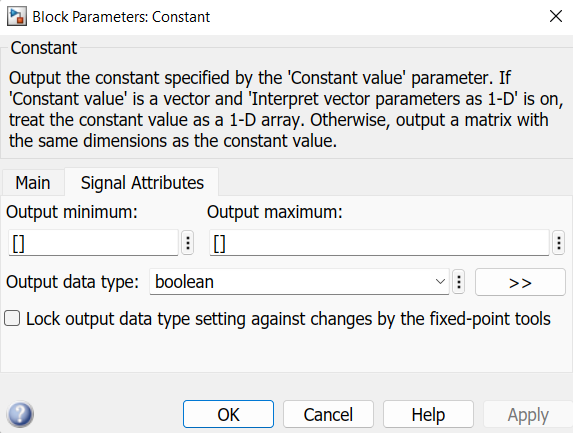


*Мал.* 8 Додаткова конвертація для генерації Verilog-коду.

При генерації Verilog-коду виникала помилка, адже на вхід цього блоку мали приходити знакові числа, тому для цієї мети на вхід блоку Atan2 йдуть два конвертори.

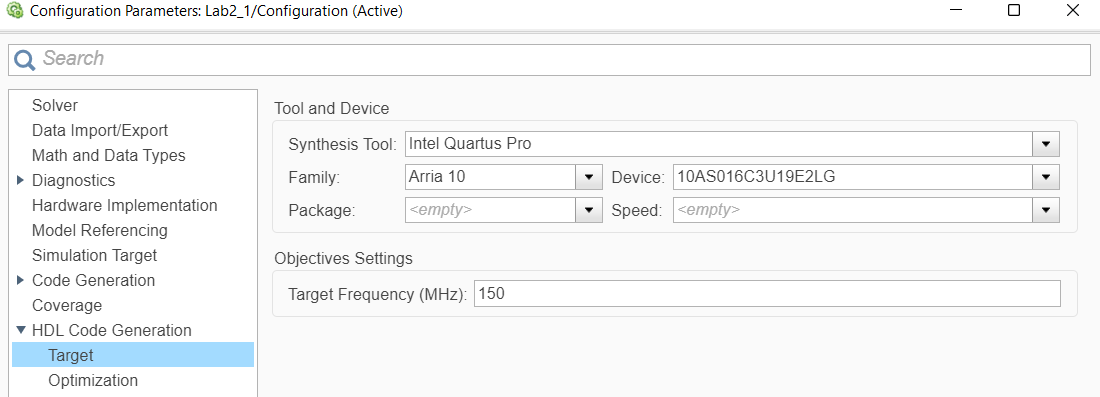


*Мал.*9 Налаштування convertor у загальній схемі.

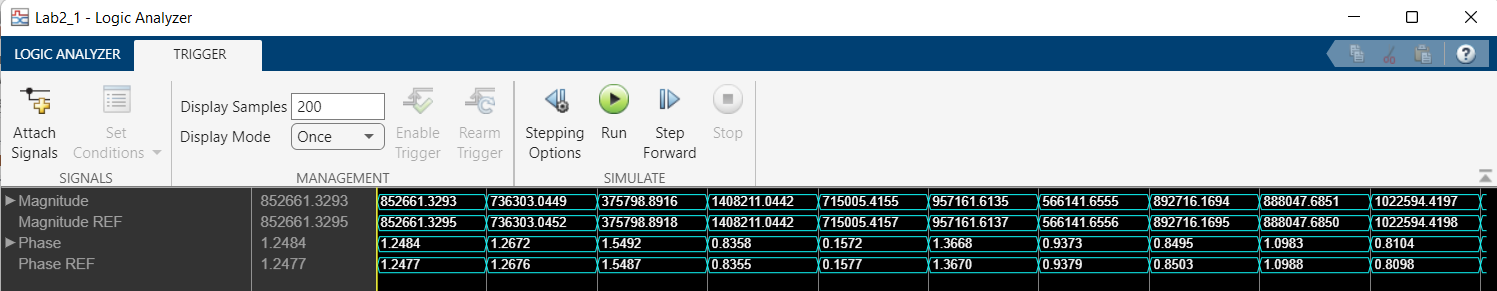


*Мал.*10 Налаштування типу константи.

Константа на мал.10 завжди встановлюється у одиничку і є сигналом валідності даних на вході та виході. Адже зазвичай при передачі даних від джерела цифрових кодів до приймача цифрових кодів завжди йде такий однобітний сигнал, щоб коли дані передаються не по кожному наступному такту, а раз у N тактів, щоби передавач міг вказати приймачу даних, що саме у цей момент часу дані валідні.



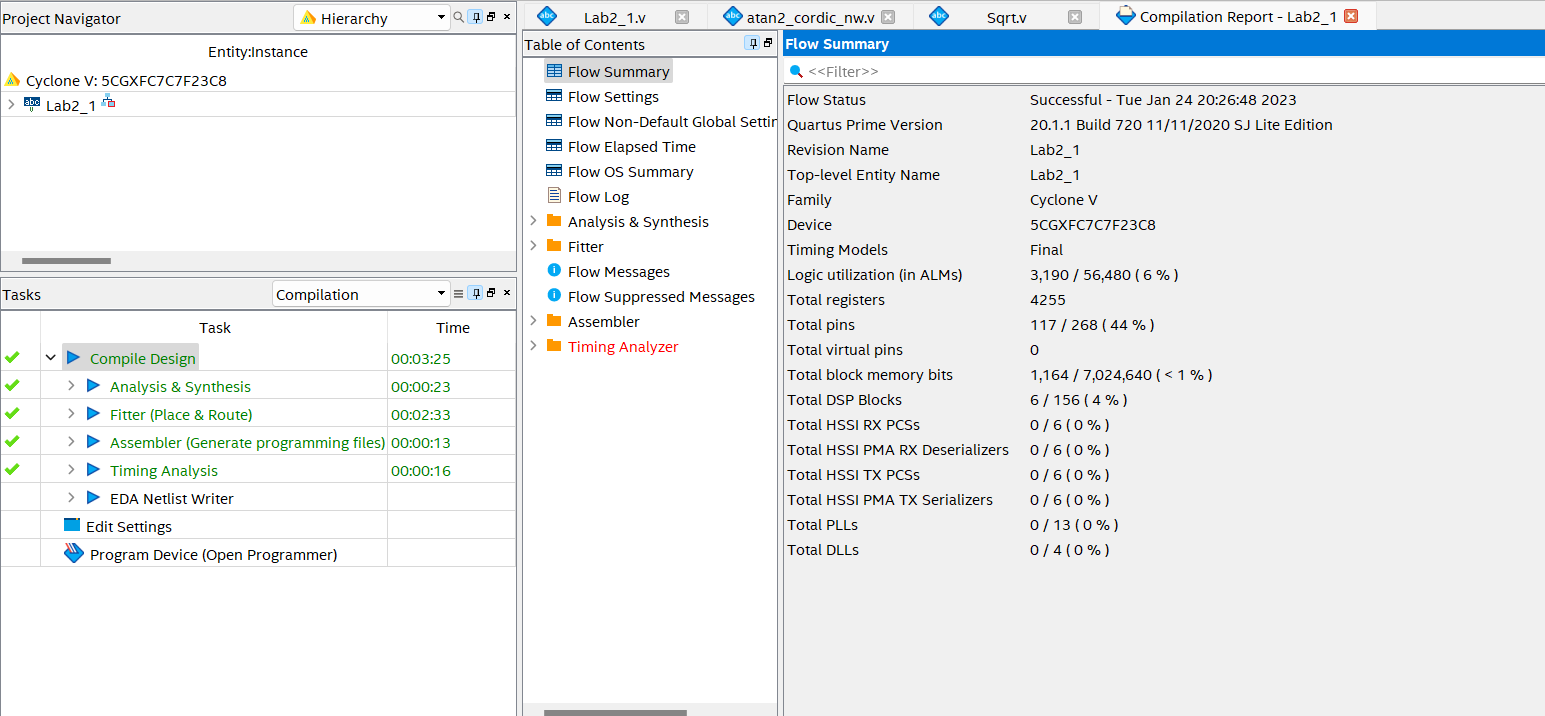
*Мал.*11 Налаштування параметрів.



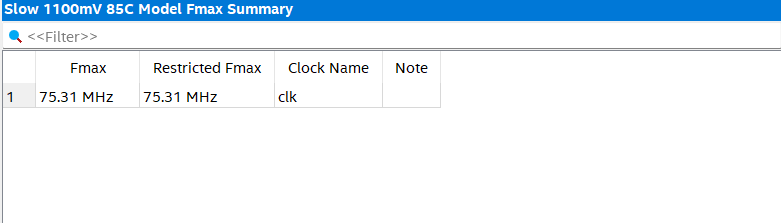
*Мал.*12 Отриманий результат.

Як бачимо, результати модуля, розрахованого за допомогою підсистеми (Magnitude), співпадають з еталонними значеннями (Magnitude REF) з невеликими відмінностями у тисячних, що є досить непоганим результатом.

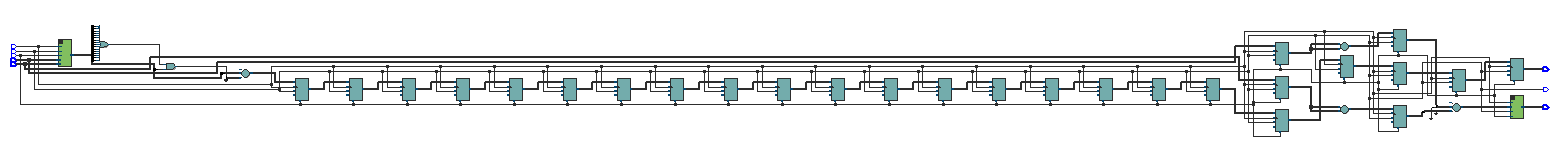
*Далі згенеруємо Verilog-код та тестбенч для нього і завантажимо у Quartus.*



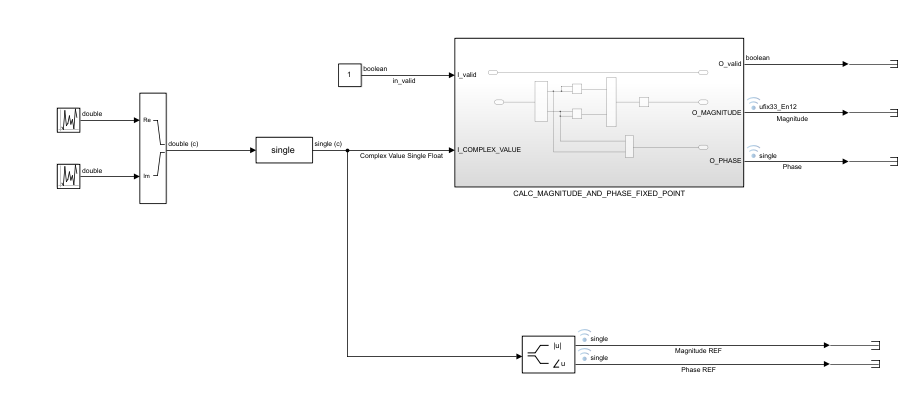
*Мал.* 13 Позитивний результат компіляції



*Мал.* 14 Максимальна частота при симуляції

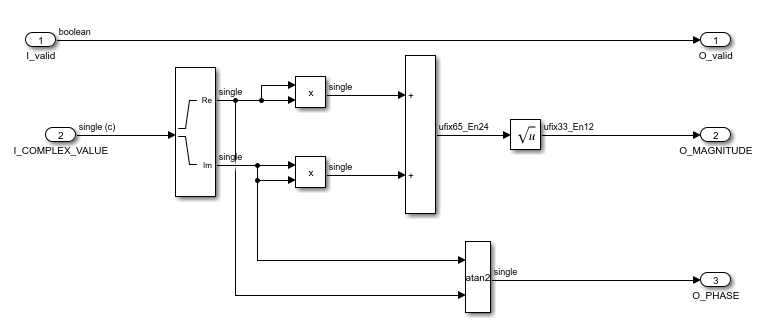


*Мал.* 15 RTL-viewer

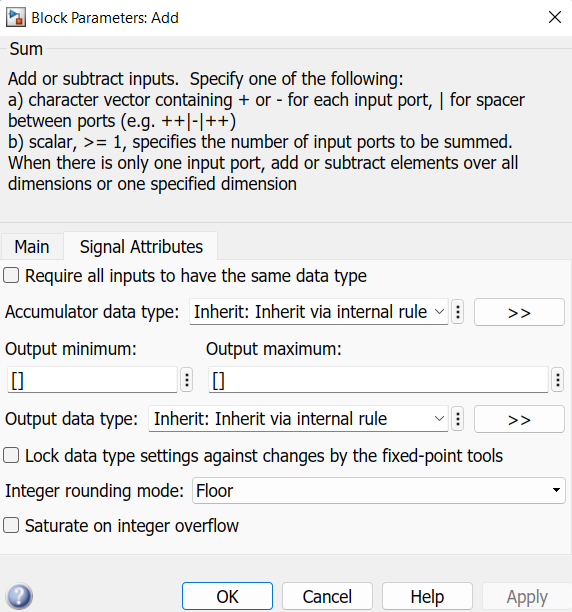


*Мал.*16 схема обчислювача з плаваючою комою.

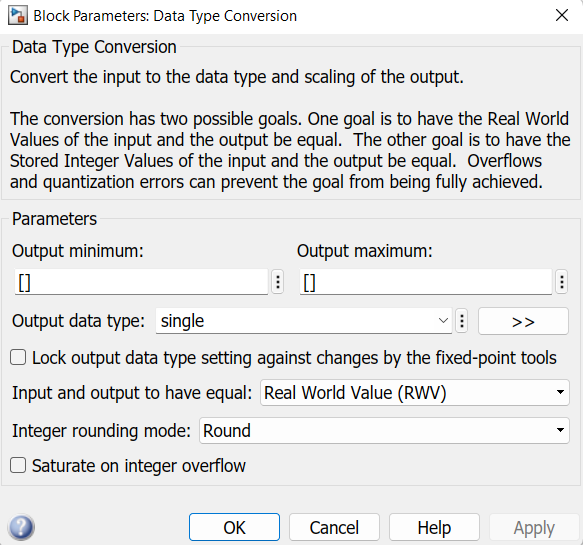
Загалом налаштування схеми обчислювача з фіксованою комою відрізняється від схеми з плаваючою комою налаштуванням ADD, Single та додатковим пунктом Floating point у параметрах.



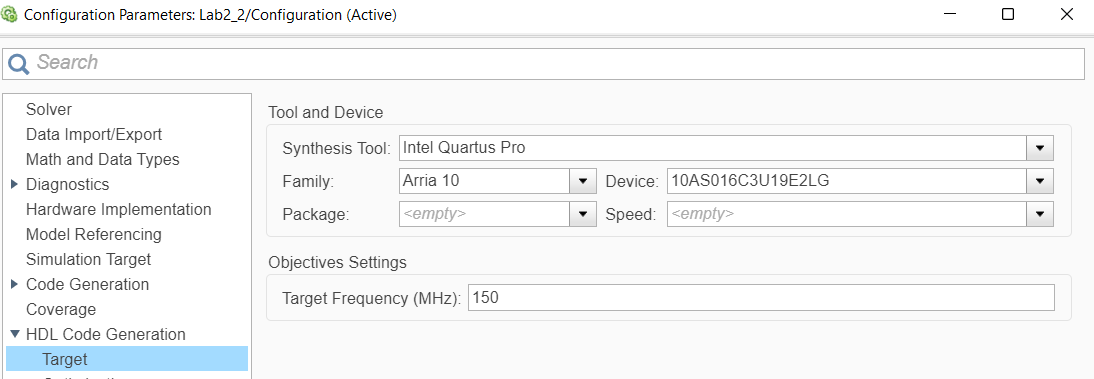
*Мал.*17 Підсистема обчислювача.

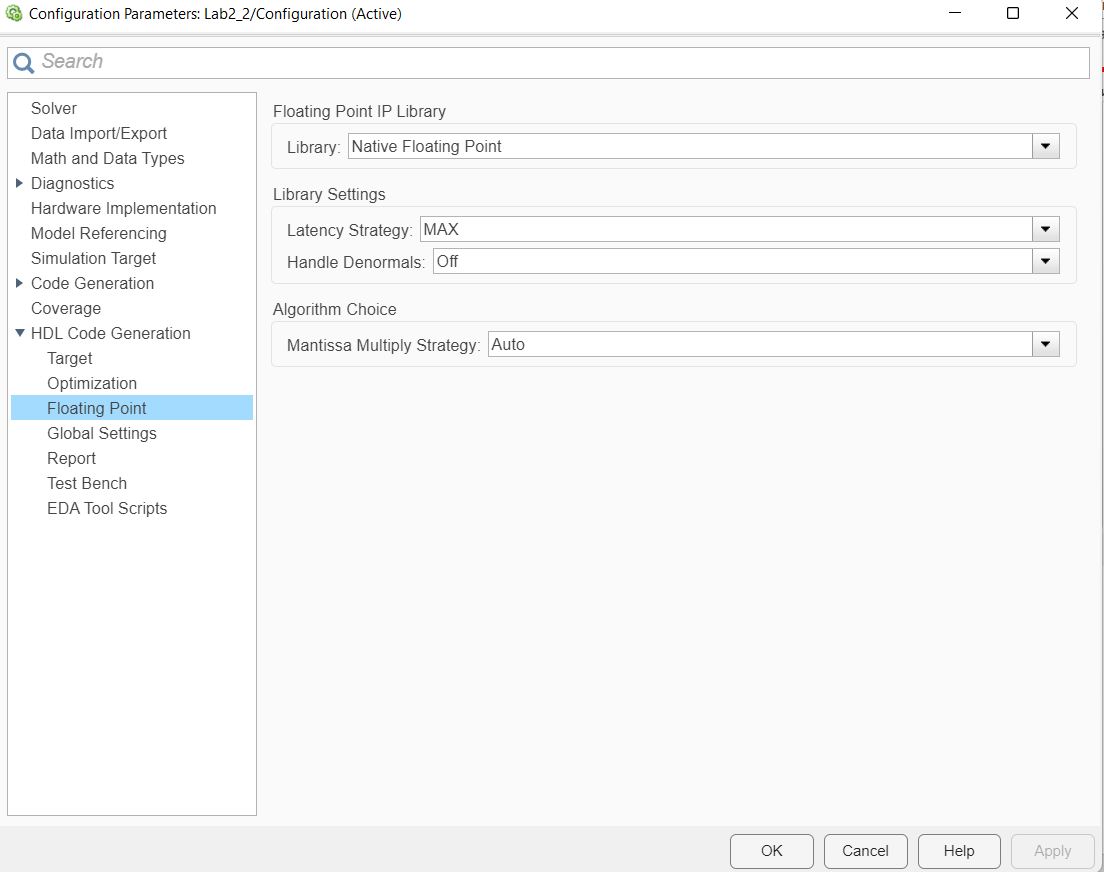


*Мал.*18 Налаштування блоку додавання.

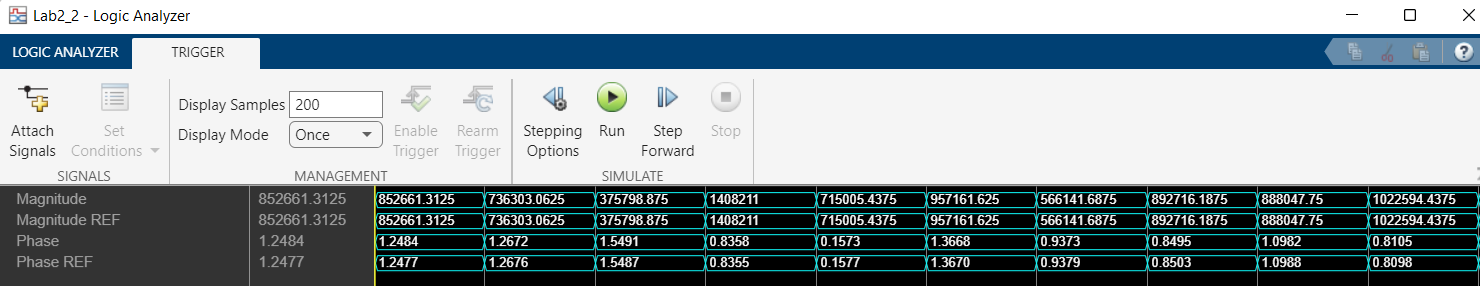


*Мал.*19 Налаштування convertor.





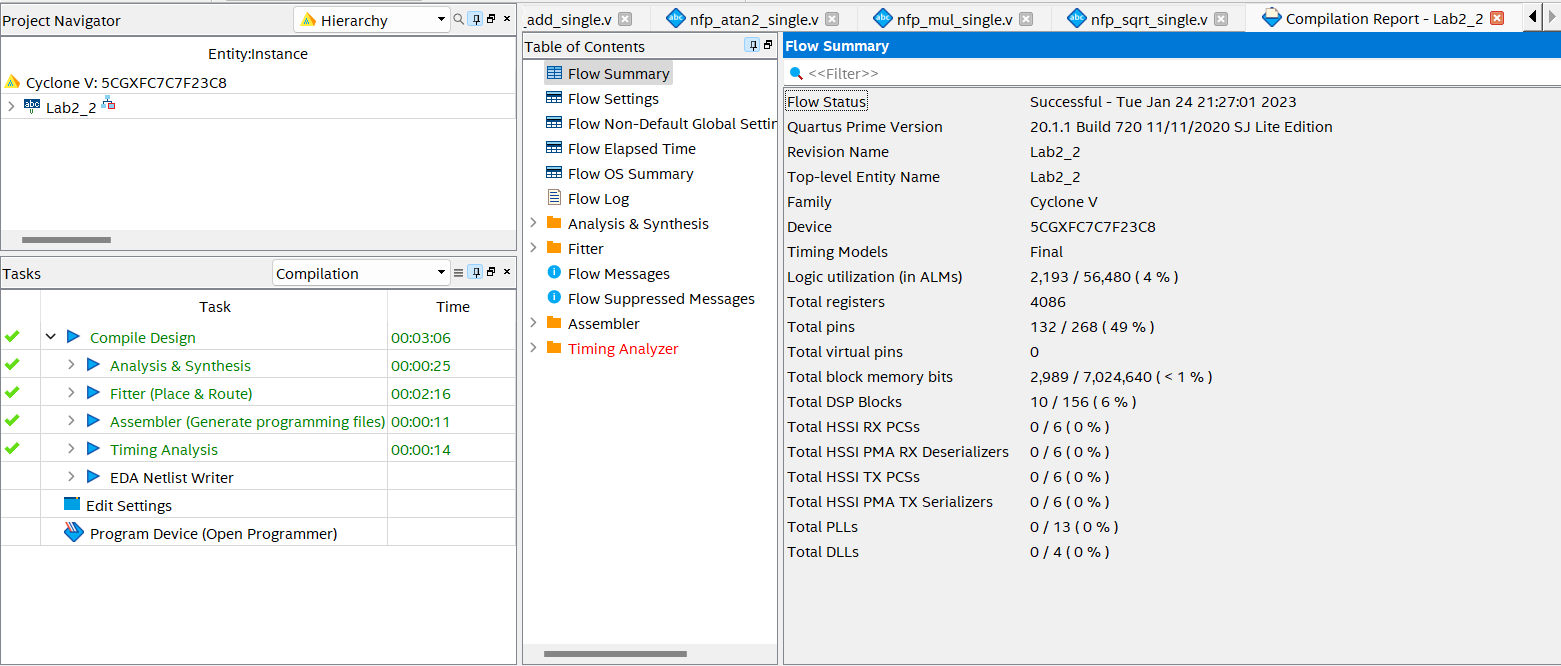
*Мал.*20 Налаштування параметрів.



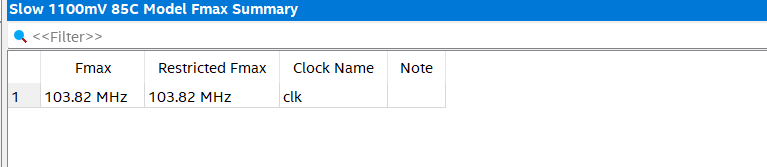
*Мал.*21 Отриманий результат.

Бачимо, що дані Phase від Phase REF, як і Magnitude від Magnitude REF відрізняються, починаючи з тисячних, що є досить непоганим результатом.

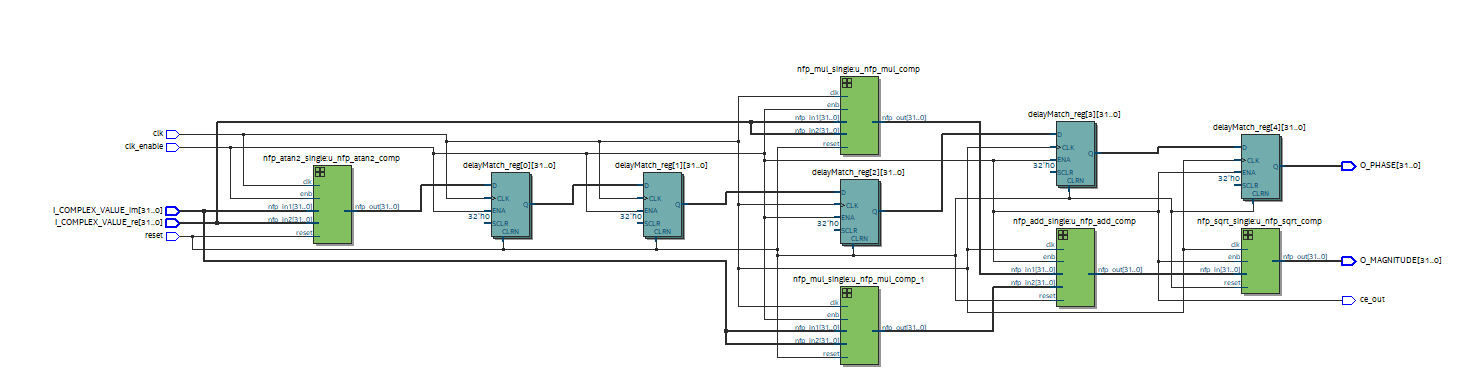
*Далі згенеруємо Verilog-код та тестбенч для нього і завантажимо у Quartus.*



*Мал.*22 Позитивний результат компіляції та апаратні витрати.



*Мал.*23 Максимальна частота



*Мал.*24 Результат RTL-viewer.