

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Модель програмно-апаратної реалізації для обчислення спектру сигналу
за допомогою швидкого перетворення Фур'є
(тема)

Виконав:

здобувач II року навчання,
групи СКСм-23-1

Васильєв О.Ю.

(прізвище, ініціали)

Спеціальність 123–Комп'ютерна інженерія
(код і повна назва спеціальності)


Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма _____
Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник проф. Хаханова І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри


(підпис)

Чумаченко С.В.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проєктування обчислювальної техніки

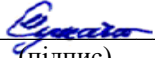
Рівень вищої освіти другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)

«___» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Васильєву Олександрю Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Модель програмно-апаратної реалізації для обчислення спектру сигналу за допомогою швидкого перетворення Фур'є

затверджена наказом по університету від 08 листопада 2024 р. № 1189 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 13 січня 2025 р.

3. Вихідні дані до роботи Швидке перетворення Фур'є

Програмовані логічні інтегральні схеми

Аналого-цифровий перетворювач

Протокол послідовної передачі даних UART

Протокол мережевої передачі даних Ethernet

UDP мережеві пакети

Графічний застосунок для керування

4. Перелік питань, що потрібно опрацювати в роботі _____

Огляд галузей використання швидкого перетворення Фур'є. Стан сучасних досліджень та актуальність теми. Математична база швидкого перетворення Фур'є.

Підбір та обґрунтування апаратних та програмних рішень.

Проектування інформаційних протоколів. Розробка апаратної реалізації.

Розробка програмного застосунку. Моделювання в САПР та імплементація на ПЛІС.

Натурні випробування.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)
21 комп'ютерна ілюстрація (слайди)

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

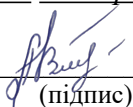
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз теми.	02.09.24 — 06.09.24	
2	Підготовка плану та структури роботи.	07.09.24 — 14.09.24	
3	Огляд літератури та збір теоретичних матеріалів.	15.09.24 — 01.10.24	
4	Розробка моделі та підходів до збору даних.	02.10.24 — 12.10.24	
5	Проведення експерименту. Обробка даних.	13.10.24 — 20.10.24	
6	Аналіз результатів і формулювання висновків.	21.10.24 — 25.10.24	
7	Написання основних розділів роботи.	26.10.24 — 24.11.24	
8	Оформлення рисунків, таблиць та посилань.	25.11.24 — 13.12.24	
9	Оформлення та редагування згідно вимог.	14.12.24 — 24.12.24	
10	Перевірка керівником та урахування зауважень.	22.12.24 — 30.12.24	
11	Завершення роботи. Підготовка до захисту.	31.12.24 — 12.01.25	
11	Захист роботи	13.01.25 — 24.01.25	

Дата видачі завдання 02 вересня 2024 р.

Здобувач _____


(підпис)

Васильєв О.Ю.

Керівник роботи _____


(підпис)

проф. Хаханова І.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Магістерська кваліфікаційна робота містить 164 сторінки, 17 рисунків, 4 таблиці, 23 джерела за переліком посилань та 4 додатки.

ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є, ПРОГРАМОВАНА ЛОГІЧНА ІНТЕГРАЛЬНА СХЕМА, PYTHON, UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER, ETHERNET

Метою роботи є розробка апаратно-програмної моделі для обчислення спектру сигналу за допомогою ШПФ на ПЛІС з урахуванням вимог до точності, продуктивності та ефективності використання ресурсів.

Проведено аналіз існуючих реалізацій і сформульовано вимоги до апаратної бази та програмного забезпечення, які забезпечують високу точність і обробку сигналів у реальному часі без втрат даних. Відповідно до цих вимог запропоновано апаратно-програмну модель ШПФ, реалізовану на ПЛІС. Розробка моделі включала проектування методів і форматів передачі даних, вибір відповідної апаратної платформи та програмних рішень. Також виконано моделювання, імплементацію, тестування та налагодження апаратно-програмного рішення.

Проведено оцінку використаних ресурсів ПЛІС, таких як логічні елементи, блоки пам'яті та споживана потужність, для визначення ефективності запропонованої моделі.

ABSTRACT

Master's qualification thesis contains: 164 pages, 17 figures, 4 tables, 23 sources according to the list of references and 4 appendices.

FAST FOURIER TRANSFORM, FIELD-PROGRAMMABLE GATE ARRAY, PYTHON, UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER, ETHERNET

The purpose of this study is to develop a hardware-software model for signal spectrum computation utilizing the Fast Fourier Transform (FFT) on an FPGA, with a focus on meeting requirements for precision, performance, and resource efficiency.

A comprehensive analysis of existing implementations was conducted, leading to the formulation of requirements for the hardware platform and software architecture to ensure high accuracy and real-time signal processing without data loss. Based on these requirements, a hardware-software FFT model was designed and implemented on an FPGA. The development process included the design of data transfer methods and formats, the selection of an appropriate hardware platform and software solutions, as well as modeling, implementation, testing, and debugging of the proposed system.

An evaluation of FPGA resource utilization, including logic elements, memory blocks, and power consumption, was conducted to assess the efficiency of the developed model.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ.....	7
ВСТУП.....	9
1 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ ЗАСТОСУВАННЯ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є В ЦИФРОВИХ СИСТЕМАХ.....	11
1.1 Сфери застосування швидкого перетворення Фур'є.....	11
1.2 Виклики при застосуванні швидкого перетворення Фур'є в цифрових системах реального часу.....	13
1.3 Огляд сучасних досліджень і розробок.....	15
2 ОБҐРУНТУВАННЯ МЕТИ ТА ВИЗНАЧЕННЯ ЗАДАЧ ДОСЛІДЖЕННЯ.....	24
3 МАТЕМАТИЧНИЙ АПАРАТ ПЕРЕТВОРЕННЯ ФУР'Є.....	27
3.1 Дискретне перетворення Фур'є.....	27
3.2 Швидке перетворення Фур'є.....	28
4 РЕАЛІЗАЦІЯ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є НА ПЛІС.....	35
4.1 Розробка апаратно-програмної моделі.....	35
4.2 Моделювання апаратної частини.....	42
4.3 Імплементация моделі та використані ресурси.....	46
4.4 Програма керування та відображення.....	48
5 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПРАКТИЧНОЇ ЦІННОСТІ.....	50
5.1 Натурні випробування реалізованої апаратно-програмної моделі.....	50
5.2 Аналіз науково-практичних результатів дослідження.....	53
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	58
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	61
ДОДАТОК Б Синтезована схема верхнього рівня на ПЛІС.....	69
ДОДАТОК В Лістинг апаратної реалізації та тестового оточення.....	70
ДОДАТОК Г Лістинг програмної реалізації.....	131

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ,
СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- АЦП – ADC – аналого-цифровий перетворювач – Analog-to-Digital Converter
ДПФ – DFT – дискретне перетворення Фур'є – Discrete Fourier Transform
ОШПФ – IFFT – обернене швидке перетворення Фур'є – Inverse Fast Fourier Transform
ПЛІС – FPGA – програмована логічна інтегральна схема – Field-Programmable Gate Array
САПР – система автоматизованого проектування
СКЩ – SCD – спектральна кореляційна щільність – Spectral Correlation Density
ШПФ – FFT – Швидке перетворення Фур'є – Fast Fourier Transform
ЦОС – DSP – Цифрова обробка сигналів – Digital Signal Processing
ANC – Active Noise Cancellation – активне шумозаглушення
ASIC – Application-Specific Integrated Circuit – спеціалізована інтегральна схема
BTMR – Block Triple Modular Redundancy – блокова потрійна модульна надмірність
CNN – Convolutional Neural Network – згорткова нейронна мережа
CPU – Central Processing Unit – центральний процесор
CORDIC – COordinate Rotation DIgital Computer – цифровий комп'ютер обертання координат
DDDWT – Double-Density Discrete Wavelet Transform – подвійно-щільне дискретне вейвлет-перетворення
DIT – Decimation In Time – проріджування у часі
DIF – Decimation In Frequency – проріджування у частоті
DL – Deep Learning – глибоке навчання

EAST – Experimental Advanced Superconducting Tokamak – експериментальний передовий надпровідний токамак

FIR-фільтр – Finite Impulse Response фільтр – фільтр зі скінченною імпульсною характеристикою

GPU – Graphic Processing Unit – графічний процесор

IA – Intellectual Analysis – інтелектуальний аналіз

IoT – Internet of Things – інтернет речей

MIMO – Multiple Input, Multiple Output – множинний вхід, множинний вихід

OFDM – Orthogonal Frequency-Division Multiplexing – ортогональне частотне мультиплексування

PHY – Physical Layer

RRE – resampling, retiming and equalizing – передискретизація, корекція часових інтервалів, нормалізація (компенсація) спотворень

UART – Universal Asynchronous Receiver-Transmitter – універсальний асинхронний приймач-передавач

WiFi – Wireless Fidelity – бездротова мережа

WT-RDO – Wallace Tree Multiplier Optimization Method – метод оптимізації множника дерева Воллеса

ВСТУП

В сучасному світі цифрова обробка сигналів є критично важливою складовою багатьох технологій, починаючи від масового використання в мобільних телефонах до вузькоспеціалізованих науково-дослідницьких приладів. Швидке перетворення Фур'є (ШПФ) — ключовий метод у цій галузі, що дозволяє швидко і точно перетворювати сигнали з часової області у частотну. Практичні задачі, вирішувані за допомогою ШПФ, охоплюють спектральний аналіз, радіолокацію, обробку звуку і зображень, телекомунікації, медичні технології та автономні системи управління в автомобільній і робототехнічній сферах.

Слід зазначити, що у сучасній практиці існують прогалини в галузі ефективної реалізації ШПФ. Серед основних проблем — висока енерговитратність, недостатня продуктивність для обробки великих обсягів даних у реальному часі та відсутність уніфікованих стандартів для інтеграції методів у промислові рішення. Такі недоліки ускладнюють впровадження ШПФ у передові системи, які потребують високої швидкості й точності обробки сигналів.

Світові тенденції у вирішенні цих задач спрямовані на вдосконалення алгоритмів та розробку апаратних прискорювачів. Особлива увага приділяється спеціалізованим обчислювальним пристроям (ASIC), що дають кращі результати при масовому використанні, та переконфігурованим обчислювальним пристроям (FPGA), які також забезпечують відносно високу продуктивність і гнучкість налаштування для конкретних завдань при зменшених ресурсах необхідних для впровадження. Крім того, дослідники активно працюють над зменшенням апаратних і енергетичних вимог, щоб зробити використання ШПФ ще більш доступним у широкому спектрі застосувань.

Актуальність цього дослідження полягає у зростанні потреби в оптимізації алгоритмів і реалізацій ШПФ для сучасних високотехнологічних систем. Метою роботи є розробка нових підходів до реалізації ШПФ втілених у вигляді апаратно-програмної моделі, що спрямовано на підвищення продуктивності і енергоефективності, а також зменшати вартість і складність інтеграції ЦОС у наявні системи. Основними галузями застосування результатів є телекомунікації, медична діагностика, радіолокація, системи автономного водіння та інші області, де аналіз даних у реальному часі має вирішальне значення.

Таким чином, вдосконалення методів реалізації ШПФ є важливим завданням сучасної науки і техніки. Робота в цьому напрямку сприятиме подоланню існуючих прогалин, забезпеченню ефективності обробки сигналів і відкриттю нових можливостей для інновацій у високотехнологічних системах.

1 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ ЗАСТОСУВАННЯ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є В ЦИФРОВИХ СИСТЕМАХ

1.1 Сфери застосування швидкого перетворення Фур'є

У сучасному світі цифрових технологій обробка сигналів є основою для багатьох критично важливих завдань, що виконують цифрові системи. Швидке перетворення Фур'є (ШПФ) є фундаментальним інструментом у цифровій обробці сигналів (ЦОС), який широко застосовується для перетворення сигналів з часової області в частотну. Це математичне перетворення є невід'ємною частиною багатьох сучасних технологій і відіграє ключову роль у широкому спектрі додатків, таких як обробка аудіо, зображень, радарних сигналів, телекомунікацій та багатьох інших галузей.

У сьогоднішньому світі споживання аудіо- та відеоінформації є значною частиною життя людини великого розповсюдження набули застосунки обробки графічної та звукової інформації. В застосунках обробки звуку ШПФ використовується для аналізу та зміни звукових сигналів, таких як фільтрація шуму або покращення якості звуку. Це включає в себе задачі вирівнювання звуку, так звані еквайзери, задачі очищення звуку від шумів, аналіз звуку для розділення його на складові та інше. У обробці зображень ШПФ дозволяє виконувати операції фільтрації та покращення зображення, такі як видалення шуму або розпізнавання об'єктів. Деякі застосунки не потребують роботи алгоритму ШПФ в реальному часі, а тому існує широкий спектр рішень задач, що не використовують спеціалізовані обчислювачі, а, як наприклад, імплементують програмну реалізацію з використанням GPU або CPU для обчислень.

Менш відомим широкій аудиторії є те, що перші практичні застосування ШПФ були пов'язані з радарними системами, зокрема цифровими приймачами. У типовому цифровому широкосмуговому приймачі сигнал, отриманий від аналогової підсистеми, оцифровується аналого-цифровим перетворювачем (АЦП). Після цього оцифровані дані передаються в спектральний оцінювач, наприклад, ШПФ, після чого параметричний кодер аналізує спектр і визначає параметри імпульсу, що описують властивості вхідного сигналу [1]. Рисунок 1.1 демонструє типовий широкосмуговий приймач.

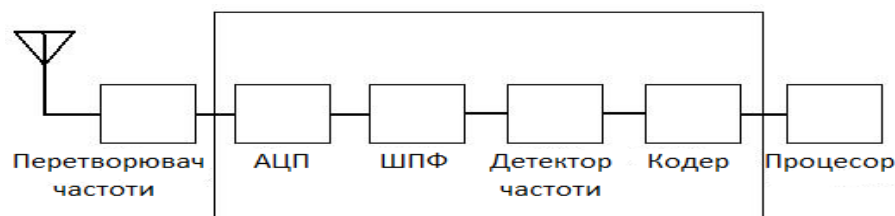


Рисунок 1.1 – Загальний вид цифрового широкосмугового приймача

Цифрові широкосмугові приймачі використовуються в задачах як цивільного так і військового спектру призначення. Такими системами є наприклад радарні системи для дослідження і моніторингу моря, повітря та космосу. Авжеж, функціонування сучасних аеропортів не можлива без радарних систем, а через те і без алгоритмів ШПФ. Сучасні автомобілі набувають все більшої автономності завдяки саме радарним системам, у яких знов таки використовується ШПФ. І не тільки автомобілі, а й так само усі розробки в напрямку автономної робототехніки. Це вже не кажучи про військові застосування для детекції будь якої активності на землі, в повітрі чи на морі. У радарних системах ШПФ допомагає в аналізі відбитих сигналів для визначення відстані до об'єктів і їх швидкості. Рішення подібних задач

здебільшого потребують вузькоспеціалізованих програмно апаратних комплексів до яких висуваються дуже високі вимоги. Ці вимоги включають точність, швидкодію, енергоефективність та інші специфічні до додатку вимоги.

Алгоритм швидкого перетворення Фур'є знайшов широке застосування у телекомунікаціях, де відіграє важливу роль у процесах обробки сигналів у системах передачі даних. Ці системи включають модемні технології та бездротовий зв'язок, де необхідно забезпечити ефективне кодування та декодування сигналів для досягнення високої швидкості, ефективності та якості передачі інформації. Особливої уваги заслуговує застосування ШПФ у бездротових технологіях, особливо WiFi та стандарту зв'язку 5G. Сучасні стандарти WiFi, такі як 802.11ac та 802.11ax, використовують ШПФ для забезпечення високої продуктивності та ефективного використання частотного спектру. Технологія 5G використовує ШПФ для реалізації MIMO та OFDM систем. Це дозволяє 5G мережам забезпечувати високу пропускну здатність, низьку затримку та високу надійність зв'язку. Модифікації ШПФ для 5G включають оптимізації для зменшення затримок та підвищення енергоефективності.

1.2 Виклики при застосуванні швидкого перетворення Фур'є в цифрових системах реального часу

Швидке перетворення Фур'є широко використовується у різних галузях науки, техніки та промисловості. Завдяки своїй універсальності, його застосовують як для задач без суворих часових обмежень, так і в режимі реального часу. Особливо важливою є роль ШПФ у застосунках реального часу для оперативної обробки даних, що є критичними для прийняття рішень у таких системах, як управління, автономні транспортні засоби та інші. Тому забезпечення високої швидкості обробки даних алгоритмом ШПФ є

надзвичайно важливим. Реальні сигнали зазвичай містять великий обсяг інформації, яку потрібно обробляти швидко, адже навіть незначні затримки можуть призводити до серйозних наслідків.

Для досягнення необхідної продуктивності зазвичай використовуються спеціалізовані апаратні рішення, такі як ASIC і FPGA. Зазвичай вибір між FPGA і ASIC залежить від специфіки задачі та процесу розробки. Менш поширені програмні рішення на базі GPU і CPU, оскільки вони не завжди гарантують результат у заданий час без суттєвого резервування ресурсів, що робить їх менш придатними для критичних застосунків, чутливих до затримок.

Енергоефективність також є важливим аспектом, особливо для мобільних пристроїв та вбудованих систем із обмеженими енергоресурсами. Сучасні пристрої, як-от смартфони, носимі гаджети й IoT, мають виконувати складні обчислення, такі як ШПФ, з мінімальним споживанням енергії, забезпечуючи при цьому високу точність і швидкодію. Для цього можуть використовуватись модифіковані алгоритми, що мінімізують кількість обчислень, або апаратні прискорювачі, які забезпечують виконання обчислень з меншою витратою енергії.

Ще однією важливою проблемою є точність обчислень, особливо у реальному часі. Числові похибки, які виникають під час обчислень, можуть суттєво впливати на результати. В алгоритмі ШПФ виконується велика кількість операцій з дійсними та комплексними числами, що в цифрових системах супроводжується накопиченням похибок. Це вирішується шляхом покращення параметрів аналого-цифрових перетворювачів або оптимізацією алгоритмів для зменшення похибок і ефективного зберігання проміжних результатів.

Оптимізація використання пам'яті є ще одним викликом, особливо для вбудованих систем з обмеженими ресурсами. У ПЛІС кількість доступної внутрішньої пам'яті часто визначає максимальну довжину ШПФ.

Використання зовнішньої пам'яті можливе, але пов'язане зі значно нижчою швидкістю доступу, що підкреслює необхідність ефективного управління пам'яттю для зменшення затримок і підвищення продуктивності.

Окрім загальних викликів, кожне застосування ШПФ має свої специфічні проблеми. Наприклад, для збільшення частоти обчислень за збереження точності можуть використовуватись методи горизонтального масштабування, які підвищують потреби у ресурсах, таких як пам'ять та енергія. Таким чином, планування та реалізація алгоритму ШПФ вимагають ретельного аналізу й оптимізації для кожного конкретного випадку.

1.3 Огляд сучасних досліджень і розробок

Дослідження в напрямку ШПФ мають відносно довгу історію і почалися в 1963 році з роботи Кулі-Тьюклі[2], що представила перший підхід до розрахунку перетворення Фур'є в цифрових системах. Сучасні застосунки з ШПФ досі використовують результати цієї роботи і включають обчислення, які потребують великої кількості арифметичних операцій. Обчислювач ШПФ зазвичай побудовано з використанням модулів типу «метелик» дискретного перетворення Фур'є (ДПФ), які охоплюють переважну більшість розрахунків загального блоку. Комплексний помножувач, який при цьому використовується блоками типу «метелик», сприяє високій складності у використанні апаратного забезпечення та енергоспоживанні. Підхід, який був реалізований для виконання цих операцій, представив у 1979 році Елвін Деспейн [3]. У своїй роботі він створив методику заміни складної арифметики множення простими операціями зсуву. Ці дві роботи заклали основу для подальшого розвитку усіх підходів до спектрального аналізу сигналів.

У свою чергу у сучасних дослідженнях і розробках у галузі обчислення ШПФ значна увага приділяється розробці нових алгоритмів і архітектур, які

можуть забезпечити високу продуктивність і енергоефективність. Використання спеціалізованих апаратних засобів, таких як спеціалізовані інтегральні схеми (ASIC) і програмовані логічні матриці (FPGA), дозволяє досягти значного прискорення обчислень ШПФ завдяки можливості паралельної обробки і оптимізації під конкретні задачі. Програмні засоби, такі як графічні процесори (GPU) та центральні процесори (CPU), також активно використовуються для обчислення ШПФ завдяки їхній гнучкості та здатності до виконання складних обчислювальних завдань. Однак, ефективність цих засобів значно залежить від оптимізації алгоритмів та архітектури програмного забезпечення.

Дослідження дизайнів цифрових приймачів і ШПФ [4][5], показали значне покращення динамічного діапазону та використання апаратного забезпечення. Запропонована в дослідженні техніка ШПФ з використанням 20 опорних відліків продемонструвала помітне покращення порівняно з попередньою технікою з використанням 12 опорних відліків. Подібно до ранніх схем проектування, ряд функцій ядра обчислювача ШПФ використовується для представлення складного множення в мережі «метелика».

Як було сказано раніше, основою апаратної реалізації ШПФ є так званий «метелик». Це radix-2 операція яка розділяє вхідну послідовність точок на пари доки не залишиться дві точки для яких розраховується ДПФ. Так, у дослідженні [6] було запропоновано нову архітектуру ШПФ з імплементацією P-parallel radix-2 на основі пам'яті, яка має за мету зменшити кількість мультиплексорів і досягнути ефективного використання пам'яті. Було запропоновано використовувати схеми перестановки після пам'яті, що зменшує використання мультиплексора до одного на одну паралельну гілку. Автори також використовують ідеальне перетасування для обчислення тієї самої перестановки на кожній ітерації, що спрощує керування та дозволяє об'єднувати пам'ять. В статті описана реалізація такої архітектури на ПЛІС

яка потребує малу кількість мультиплексорів і демонструє низьке використання площі.

Автори [7] представляють нову 64-паралельну 4096-точкову архітектуру ШПФ на основі пам'яті radix-2 для 6G. Цей підхід є першим, який використовує 64 паралельні гілки в архітектурах на основі пам'яті. Була вирішена проблема великої кількості пам'яті, яка потрібна для такого високого розпаралелювання, використовуючи однакову адресу читання та запису для всіх пам'ятей та організовуючи їх у групи для усунення надлишкових регістрів. Автори також використовують нову конструкцію пам'яті обертання, яка дозволяє повторно використовувати коефіцієнти обертання серед паралельних ротаторів, а також новий дизайн для лічильника фаз, який контролює архітектуру. Ця архітектура ШПФ була реалізована на ПЛІС Virtex7. Експериментальні результати показують, що запропонована архітектура забезпечує відносно низьку затримку виміряну в тактах і відносно високу пропускну здатність у вибірках за такт при порівнянні з іншими ШПФ на основі пам'яті.

Натомість, автори [8] презентують гнучку та переконфігуровану архітектуру ШПФ на основі підходу ШПФ з змішаними «метеликами», яка підтримує багато режимів та довжин перетворення для систем високопродуктивного зв'язку, таких як системи 5G. Автори використовують алгоритм ШПФ з децимацією за часом, який дозволяє зменшити невідповідності між швидкістю обчислення обчислювальних елементів та пропускну здатністю пам'яті. Вони також використовують ШПФ з прорідженням у часі оснований на багато-паралельних блока одиничних перестановок «метелика», щоб покращити використання апаратних ресурсів. Крім того, автори пропонують структуру об'єднання та стиснення множників для зменшення області і затримки блоків генерації множників. Архітектура ШПФ була реалізована в технології CMOS 40 нм від Semiconductor Manufacturing International Corporation (SMIC) з робочою частотою 483 МГц,

демонструючи продуктивність на рівні 654 Мб/с для ШПФ розміром 2048 і 1536 точок.

Ще одним рішенням для оптимізації є використання блокової потрійної модульної надлишковості (Block Triple Modular Redundancy - BTMR) для зменшення впливу помилок на модуль цифрової обробки, такий як ШПФ. BTMR - це техніка, яка використовує три копії логічних блоків та мажоритарне голосування для виявлення та виправлення помилок. Автори [9] дослідили вплив архітектурного вибору на частоту помилок, частоту відмов загального режиму та співвідношення сигнал/шум для різних послідовних і конвеєрних архітектур ШПФ з використанням BTMR. Вони використали підхід ін'єкції помилок для аналізу цілісності ШПФ на ПЛІС. Авторські експериментальні результати показали, що конвеєрні архітектури мають більшу частоту відмов загального режиму, але нижче співвідношення сигнал/шум, ніж послідовні архітектури, а також зауважили що частота виникнення помилок сильно залежить від використання ресурсів.

Важливим ресурсом для додатків з обчислення ШПФ є витрачана енергія. Дослідження спрямовані на зменшення напруги, а тому зменшення споживаної енергії, описані в статті [10]. Автори презентують метод використання тотожності Парсеваля, яка є внутрішньою властивістю перетворення Фур'є, для виявлення помилок у прискорювачах ШПФ, які працюють при зниженій нарузі. Автори не вносять жодних змін до схеми, а лише додають алгоритмічний модуль, який порівнює енергію вхідного та вихідного сигналів ШПФ і виявляє помилки, якщо вони не рівні. Автори реалізують свій метод на системі на ПЛІС, призначеній для роботи при зниженій нарузі. Вони також створили демонстраційний випробувальний стенд, який складається з програмного та апаратного забезпечення, для профілювання здатності виявлення несправностей. Автори показують, що їх метод дозволяє знизити споживання енергії на 43% без втрати пропускну здатності та надійності. Вони також вказують, що накладні витрати їх методу

виявлення несправностей сублінійно масштабуються відносно розміру ШПФ і становлять 10% для 1024-точкового ШПФ.

Для розрахунку ШПФ часто використовують поворотні кутові коефіцієнти які можуть зберігатись у пам'яті, а можуть вираховуватись в реальному часі, тобто для економії пам'яті можуть проводитись тригонометричні розрахунки. Одним з найпоширеніших алгоритмів для таких розрахунків в апаратурі є алгоритм CORDIC. Основна ідея алгоритму CORDIC полягає в тому, щоб представити необхідні обчислення у вигляді послідовності обертань навколо координатних осей. Цей процес може бути представлений у вигляді матричних множень і може бути ефективно реалізований за допомогою операцій зсуву бітів та додавань [3]. Стаття [11] пропонує класифікацію та діагностику несправностей для блоків CORDIC, які використовуються для виконання комплексного множення в ШПФ на ПЛІС. Автори використовують метод ін'єкції помилок для аналізу впливу несправностей на продуктивність та якість ШПФ. Вони також застосовують ШПФ для вилучення ознак з сигналів несправностей та використовують машинне навчання для класифікації та прогнозування несправностей. Показано, що авторський метод має високу точність прогнозування несправностей та може покращити надійність та продуктивність блоків CORDIC на ПЛІС.

Альтернативним методом оптимізації використовуваних ресурсів ПЛІС є використання алгоритму radix-16 CORDIC для виконання комплексного множення коефіцієнтів обертання. Автори [12] запропонували високопродуктивний ротатор на основі radix-16 CORDIC, який зменшує кількість ітерацій та потребу у великій пам'яті для зберігання коефіцієнтів обертання. Вони також реалізували алгоритм radix-2 проріджування по частоті на ПЛІС Virtex 7 та порівняли його з іншими реалізаціями ШПФ. Їхні результати показали, що запропонований ротатор на основі radix-16 CORDIC

має високу продуктивність та ефективність, у порівнянні з іншими методами комплексного множення.

Стаття [13] пропонує інший паралельний ротатор для ШПФ і оберненого ШПФ (ОШПФ), які застосовуються в системах радіозв'язку з кількома несучими. Автори покращують обчислювальну складність, затримку обробки та точність обертання за допомогою свого алгоритму, який також значно зменшує споживання апаратних і енергетичних ресурсів. Вони також розробили апаратну архітектуру ротатора фази та реалізують її на ПЛІС. Автори демонструють, що їх ротатор має кращі характеристики порівняно з деякими сучасними алгоритмами.

Цифрові системи формування сигналів спектрального детектора вимагають високої енергетичної роздільної здатності, низького балістичного дефіциту та високої швидкості розрахунків. Трапецієподібні алгоритми формування є одним з ефективних способів досягнення цих вимог, оскільки вони дозволяють легко регулювати час підйому та плоскої вершини трапеції, а також зберігати два складені експоненціальні імпульси з різними піками. Однак, трапецієподібні алгоритми формування також мають деякі недоліки, такі як спотворення сигналу через старіння детектора, температурний дрейф, сигнальний шум та коливання. Розглянемо рішення, які були запропоновані в науковій літературі, для подолання цих проблем за допомогою швидкого перетворення Фур'є на ПЛІС. Наприклад, стаття [14] презентує метод використання ШПФ для обчислення константи затухання експоненціального імпульсу, який виводиться детектором енергетичного спектру. Автори використовують цифровий синтез форм імпульсу, щоб адаптивно розробляти нові формувачі відповідно до різних характеристик детектора. Вони також підтримують швидку процедуру калібрування для відповідності різним спектрам шуму та вирішення непередбачуваних коливань сигналу. Автори реалізували свою адаптивну трапецієподібну систему формування на ПЛІС і підтвердили її відносну похибку коефіцієнтів, яка становить менше 0,8%.

Висока проблематика також з'являється при дискретизації з високою частотою. Одним з можливих рішень запропонованим в статті [15] є використання нової схеми з'єднання каналів для швидкісного АЦП, яка не відповідає розповсюдженому стандарту послідовного АЦП JESD204. Це дозволяє отримати плату та мікросхему з аналоговою смугою пропускання 16 ГГц, що полегшує вибірку сигналу в другій зоні Найквіста. За допомогою ШПФ, розрахованого у Матлаб (Matlab), автори перевірили спектр оцифрованого сигналу та отримали наступні результати: вільний від паразитних сигналів динамічний діапазон 18–33 дБ та ефективна кількість бітів 2,1–3,7 біта для першої та другої зон Найквіста відповідно. Це показує, що така схема з'єднання каналів може покращити продуктивність ШПФ реалізованого на ПЛІС.

Стаття [16] пропонує новий комбінований алгоритм передискретизації, повторної синхронізації та вирівнювання (RRE) для когерентних оптичних приймачів, які використовують ШПФ та ОШПФ для обробки сигналів. Автори використовують один фільтр з кінцевою відповіддю на імпульс (FIR-фільтр) для виконання всіх трьох функцій, що зменшує складність. Вони також розробляють апаратну архітектуру ротатора фази та реалізують її на ПЛІС. Автори демонструють, що їх алгоритм знижує величину помилки та споживання апаратних ресурсів порівняно з традиційними схемами у часовій та частотній областях. Вони також реалізували реально-часовий когерентний оптичний приймач зі швидкістю 106,24 Гбіт/с (26,56 ГБод) на основі однієї мікросхеми ПЛІС з використанням чотирьох 6-розрядних АЦП з частотою дискретизації $\sim 31,48$ MSPS. Автори стверджують, що на момент публікації це найвища зареєстрована бітова швидкість когерентного приймача на основі однієї мікросхеми ПЛІС.

Як бачимо при розвитку додатків що використовують ШПФ вчені та інженери стикаються з багатьма викликами. Це значно уповільнює розробку та введення в експлуатацію нової апаратури. Стаття [17] описує

встановлення та інтеграцію двох нових інтерферометрів для експерименту EAST, який використовує ШПФ для вимірювання густини плазми. Автори розробляють єдиний апаратний шаблон для обробки сигналів в реальному часі з трьома основними частинами: диджиталізація, цифрова обробка сигналу та вихідні модулі. Автори використовують ПЛІС як апаратну основу для реалізації цього шаблону, який може працювати з різними типами інтерферометрів. Були використані два різні методи для вилучення та розгортання фази з сигналів інтерферометрів: метод відношення інтенсивності та метод ШПФ або фазової демодуляції. Автори також надають інтерфейси параметрів для налаштування їхнього шаблону під різні сценарії застосування.

Стаття [18] пропонує метод аналізу відношення між довжиною слова та шумом сигналу до квантування в реалізаціях спектральної кореляційної щільності (СКЩ, SCD) із фіксованою комою. СКЩ є важливим інструментом для виявлення та класифікації циклостаціонарних сигналів, але вона має високу обчислювальну складність. Автори досліджують класичний алгоритм оцінки СКЩ, а саме метод накопичення ШПФ з використанням арифметики з фіксованою комою. Вони отримують вирази закритої форми для відношення сигнал/шум і порівнюють їх із довжиною слова від 14 до 26 біт. Вони також розробляють дизайн ПЛІС і показують, що він має високу енергоефективність та пропускну здатність порівняно з графічним процесором (GPU).

Цифрові приймачі не єдине поле для використання ШПФ на ПЛІС. Стаття [19] презентує новий підхід на основі глибокого навчання (DL) та інтелектуального аналізу (IA) для покращення роздільної здатності та якості зображень, які використовують ШПФ. Автори використовують згорткову нейронну мережу (CNN) для відновлення втраченої інформації під час збільшення роздільної здатності зображень. Вони також застосовують техніку дискретного вейвлет-перетворення подвійної щільності (DDDWT)

для видалення шуму та покращення візуальної якості. Крім того, вони використовують метод оптимізації помножувача дерева Уоллеса (WT-RDO) з перекриттям і додаванням ШПФ для досягнення високої роздільної здатності та якості зображення. Продемонстровано, що такий підхід досягає високої точності, чутливості, специфічності, часової складності, потужності, затримки та пропускну здатності.

Стаття [20] презентує систему активного шумозаглушення (ANC) з фіксованим фільтром із алгоритмом класифікації згорткової нейронної мережі (CNN) для аудіопристроїв. Активне шумозаглушення є важливою функцією сучасного аудіопристрою, оскільки воно усуває небажані навколишні шуми, але воно має коротку вторинну затримку в звичайних алгоритмах найменшого середнього квадрата, що ускладнює реалізацію високоякісної апаратури. Автори використовують нейронні мережі для врахування короткої затримки вторинного шляху та зменшення коефіцієнта шуму. Вони також вводять класифікацію шуму в частотній області та алгоритм вибору коефіцієнтів для усунення шуму для систем, що змінюються в часі. В статті описується реалізація архітектури шумозаглушення на ПЛІС з використанням парно-непарного буферу, який ефективно обчислює ШПФ і згортку з перекриттям. Також було продемонстровано, що запропонована система покращує співвідношення сигнал/шум та знижує час обробки та динамічне енергоспоживання.

З розвитком технологій постійно вдосконалюються алгоритми ШПФ. Модифікації включають оптимізації для специфічних умов використання, таких як робота в реальному часі, обмеження енергоспоживання та підвищення точності обробки сигналів. Були розглянуті деякі проблеми та рішення, пов'язані з реалізацією ШПФ на ПЛІС. Видно, як оптимізуючи алгоритми та підходи, використовуючи різні схеми з'єднання каналів, алгоритми комплексного множення та техніки надлишковості можна покращити продуктивність, ефективність та надійність ШПФ.

2 ОБҐРУНТУВАННЯ МЕТИ ТА ВИЗНАЧЕННЯ ЗАДАЧ ДОСЛІДЖЕННЯ

Швидке перетворення Фур'є є актуальним алгоритмом, який використовується у широкому спектрі додатків, що підтверджується великою кількістю досліджень. Високої уваги заслуговують імплементації ШПФ на програмованих логічних інтегральних схемах, оскільки вони дозволяють поєднати швидкодію, енергоефективність, ресурсоемність та гнучкість такого рішення.

Існує багато досліджень в області покращення імплементації алгоритму ШПФ, однак часто інформація з різних досліджень є розрізною, не є вичерпною і не дозволяє порівняти запропоновані рішення між собою. Деякі дослідження є загальними і теоретичними, а інші спрямовані на конкретні архітектури ПЛІС та оптимізовані під специфічні задачі. Тому впровадження моделі ШПФ, що дозволить об'єктивно порівняти ідеї запропоновані у розрізних дослідженнях, дуже необхідне.

Окремо слід зазначити, що порівняння можна провести теоретично, без апаратної бази, але розробка для ПЛІС є комплексним процесом і існує велика ймовірність пропустити деталі, які вплинуть на результати оцінки та висновки. Тому найкращим рішенням є апаратно-програмна реалізація запропонованої моделі з можливістю подальшого використання підходів описаних в різноманітних дослідженнях ШПФ для подальшого порівняння на одній і тій самій апаратній базі. Це дозволить виявити сильні і слабкі сторони означених пропозицій і прискорити пошук оптимальних імплементацій для застосунків, що використовують ШПФ. Додатково результуюче впровадження можна використовувати як завершений прилад, наприклад аналізатор спектру, або інтегрувати в існуючий додаток що потребує ЦОС в реальному часі.

Для якісної оцінки результатів роботи ШФП у процесі випробувань потрібно збирати вичерпну інформацію, що дасть можливість порівнювати імплементації ШПФ. Як еталонну імплементацію доцільно використовувати ту, що запропонована розробником ПЛІС. При цьому слід означити аспекти, які необхідно оцінювати, а саме: точність, використовувані ресурси ПЛІС, швидкодія та енергоефективність.

Для оцінки точності необхідно мати повну інформацію про вхідні та вихідні дані алгоритму ШПФ. Зрозуміло, що для порівняння двох алгоритмів можна реалізувати схему, де два ШПФ реалізовані на різних апаратних базах та проводити порівняння в реальному часі, але це не є ефективним та раціональним. Краще використати перевагу ПЛІС, а саме можливість перепрограмування, і проводити експерименти окремо, порівнюючи результати пізніше. Для такого порівняння необхідно реалізувати швидкий та точний спосіб запису усіх вхідних та вихідних даних у реальному часі.

Використовувані ресурси кожної імплементації оцінити відносно легко, оскільки більшість сучасних середовищ розробки для ПЛІС надають вичерпну інформацію про внутрішню структуру і залучені ресурси кожної імплементації. При умові впровадження подібних алгоритмів з однаковими параметрами довжини та розрядності усі відмінності у використовуваних ресурсах наочно продемонструють усі відмінності.

При оцінці параметрів швидкодії слід розділити підходи до обчислення ШПФ на два класи: потокові та паралельні. Поточкові реалізації дозволяють гарантовано обробляти потік даних без втрат. Паралельні реалізації використовують менше ресурсів для свого обчислення, але вони не підходять для роботи з потоками даних. Тому доцільно сфокусуватись на поточкових алгоритмах як на тих, що пропонують більшу повноту. Апаратна база ПЛІС дозволить імплементувати і паралельні реалізації, але як відправну точку оберемо саме поточкові алгоритми. В такому випадку алгоритм ШПФ слід розглядати як конвеєр. Припустимо, що алгоритм з поточною реалізацією

працює на частоті дискретизації, тоді швидкодію можна визначити як час від надходження даних до блоку ШПФ до виходу результату. У випадку постійної апаратної бази можлива також оцінка затримки від надходження у ПЛІС до детектування у системі моніторингу чи запису. У такому випадку механізми передачі даних в ПЛІС та їх зчитування стають також частиною конвеєра, але при постійній апаратній базі ці частини будуть незмінними і не завадять оцінці швидкодії.

Оцінка енергоефективності також спрощується при постійній апаратній базі. По-перше, однакова апаратна база спрощує вимірювання споживання енергії як таке. По-друге, сучасні середовища розробки для ПЛІС надають відносно надійні результати оцінки енергоспоживання, що дозволяють проводити оцінку споживання на етапі проєктування. Але для найбільшої точності слід підтвердити розраховані значення в процесі натурних експериментів.

3 МАТЕМАТИЧНИЙ АПАРАТ ПЕРЕТВОРЕННЯ ФУР'Є

3.1 Дискретне перетворення Фур'є

Перетворення Фур'є є одним із найпоширеніших інструментів, які використовуються для перенесення функції з часової області на частотну. В цифровій обробці сигналів алгоритм дискретного перетворення Фур'є (ДПФ) широко використовується для аналізу сигналів при перенесенні даних в частотну область. ДПФ для N точок виконує перетворення даних часової області в дані частотної області. ДПФ працює з N -точковою послідовністю чисел, яку зазвичай називають $x(n)$, і отримують послідовною вибіркою фіксованого періоду деякої безперервної функції $f(x)$ [21]. Функція ДПФ для $X(k)$, яка є N -точковою послідовністю $x(n)$, визначена в рівнянні 3.1:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}kn}, \text{ де } 0 \leq k \leq N-1 \quad (3.1)$$

Але більш поширене та спрощене позначення для ДПФ можна побачити в рівнянні 3.2:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{kn}, \text{ де } 0 \leq k \leq N-1 \quad (3.2)$$

де W_N представляє коефіцієнт обертання або “корінь N -го з одиниці” комплексного множника. Математично, W_N описує комплексний множник, який реалізує обертання векторів на одиничному колі на кут. Геометрично це відповідає точкам, рівномірно розподіленим на одиничному колі в комплексній площині. Для подальшого використання, слід формалізувати визначення змінної W_N в рівняннях 3.3 і 2.4:

$$W_N = e^{-j\frac{2\pi}{N}} \quad (3.3)$$

$$W_N = \cos\left(\frac{2\pi}{N}\right) - j \cdot \sin\left(\frac{2\pi}{N}\right) \quad (3.4)$$

Термін «N-ний корінь з одиниці» часто використовується для опису фактора обертання, оскільки рівняння 3.3 можна змінити, щоб дати наступне визначення сформульоване в 3.5:

$$(W_N)^N = e^{-j2\pi} = 1 \quad (3.5)$$

Важливою проблемою реалізації ДПФ для N-точкової послідовності є чисельні складні обчислення, які викликають проблеми для високошвидкісної обробки сигналу. Для обчислення послідовності дискретне перетворення Фур'є вимагає $2N^2$ або $O(N^2)$ операцій довжиною N [21]. Це означає, що для N-точкового ДПФ знадобиться $N \times (N - 1)$ комплексних додавань і N^2 комплексних множень, що потребуватиме надмірної кількості апаратного забезпечення [21].

3.2 Швидке перетворення Фур'є

Як було зазначено раніше, ДПФ складний та ресурсоємний алгоритм, тому через його складність він ніколи не обчислюється безпосередньо. Натомість, Кулі та Тьюклі був розроблений набір ефективних алгоритмів [3] для значного прискорення обчислень ДПФ, що отримав назву швидке перетворення Фур'є. ШПФ забезпечує метод «розділяй і володарюй» для покращення швидкодії відносно складних алгоритмів ДПФ. ШПФ здатне ефективніше розкласти та обчислювати ДПФ, використовуючи властивість

симетрії та періодичності комплексної послідовності W_N [23]. Визначимо наступні властивості коефіцієнту обертаня визначені в рівняннях 3.6 і 3.7.

$$W_N^{k+N/2} = -W_N^k \quad (3.6)$$

$$W_N^{k+N} = W_N^k \quad (3.7)$$

де рівняння 3.6 відоме як властивість симетрії, а рівняння 3.7 відоме як властивість періодичності.

Скориставшись перевагами ДПФ, можна отримати набагато вищу ефективність та знизити алгоритмічну складність операцій нижче $O(N^2)$. Щоб продемонструвати факторизацію ШПФ, розглянемо обчислення ДПФ для $N=2m$ точок, де m є додатним цілим значенням. N -точкову послідовність $x(n)$ можна розділити на дві окремі послідовності довжиною $N/2$. Перша група складається з парних відліків, а друга складається з непарних. Результиуюче проріджене ДПФ у N -точках виражається у рівнянні 3.8:

$$X(k) = \sum_{n=0,2,4,\dots}^{N-2} x(n) \cdot W_N^{kn} + \sum_{n=1,3,5,\dots}^{N-1} x(n) \cdot W_N^{kn} \quad (3.8)$$

Набір парних і непарних вибірок представлено як $2m$ та $2m+1$ у рівнянні 3.9, де $m=0, 1, \dots, N/2 - 1$.

$$X(k) = \sum_{m=0}^{N/2-1} x(2m) \cdot (W_N^2)^{mk} + \sum_{m=0}^{N/2-1} x(2m+1) \cdot (W_N^2)^{mk} W_N^k \quad (3.9)$$

Цей підхід є основою для методу декомпозиції ШПФ, де сигнал розбивається на дві частини: одну, що відповідає парним індексам, і другу —

непарним. Такий поділ дозволяє зменшити розмір задачі вдвічі, забезпечуючи рекурсивне виконання алгоритму.

Використовуючи властивість сформульовану раніше у рівнянні 3.5, та обравши $N=2$, отримаємо рівняння 3.10 та 3.11.

$$W_N^2 = e^{-j\frac{2\pi}{N/2}} \quad (3.10)$$

$$W_N^2 = W_{N/2} \quad (3.11)$$

В свою чергу рівняння 3.9 можна переписати наступним чином:

$$X(k) = \sum_{m=0}^{N/2-1} x(2m) \cdot W_{N/2}^{mk} + W_N^k \sum_{m=0}^{N/2-1} x(2m+1) \cdot W_{N/2}^{mk} \quad (3.12)$$

де $x(2m)$ послідовність, що складається з парних вибірок, а $x(2m+1)$ послідовність, що складається з непарних вибірок $x(n)$. Оскільки ДПФ є періодичним, непарні та парні сегменти потрібно обчислювати лише за $1/2 N$ значень k , або $N/2$ разів. Результатом є децимація великої кількості необхідних операцій, зазвичай пов'язаних із прямим обчисленням ДПФ, викладену у рівнянні 3.1.

Оскільки кожен із етапів ДПФ розбивається на дві менші парні та непарні послідовності, вважається, що він відноситься до класу ШПФ, який називається ШПФ з проріджуванням (децимацією) у часі (DIT) [21][22]. Потік даних представленого алгоритму проріджування в часі ШПФ можна побачити на рисунку 3.1 для $N = 8$ вхідних точок. Згідно представленого графу, процедура децимації повторюється для $\log_2(N)-1$ разів, що дає $\log_2(N)$ етапів, доки послідовність на останньому етапі не буде скорочена до $N/4$ -точкового ДПФ [21]. Крім того, представлений підхід дозволяє оптимізувати порядок обчислень, організовуючи їх так, щоб забезпечити зручну апаратну чи програмну реалізацію. Зокрема, у випадку $N = 8$ видно, як алгоритм

покроково обчислює часткові результати, поступово інтегруючи їх у фінальну частотну картину сигналу.

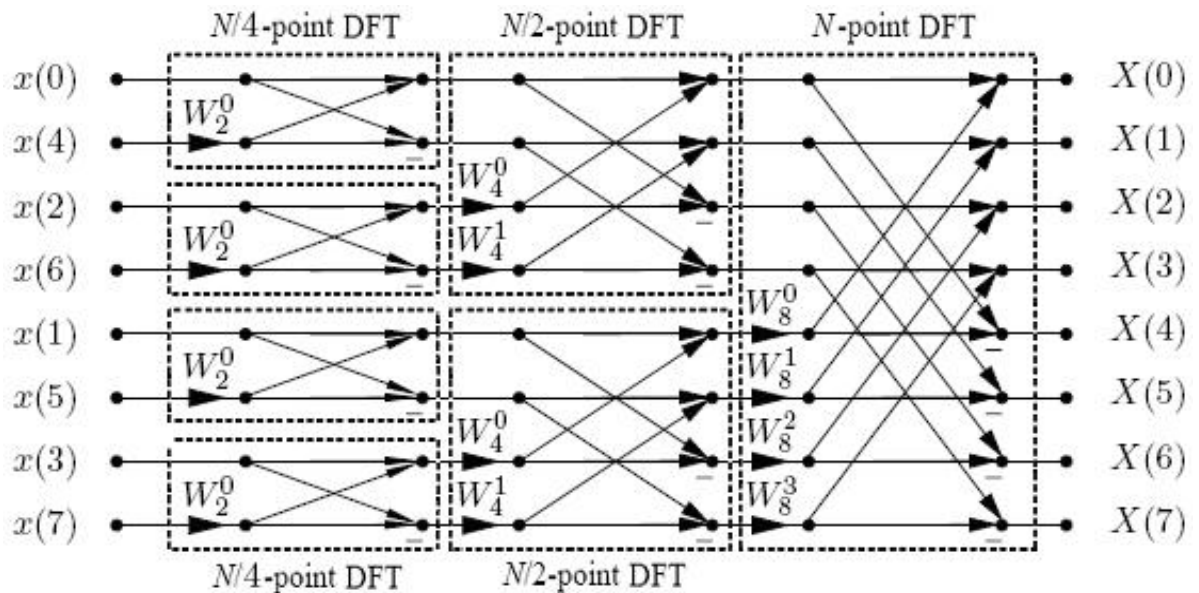


Рисунок 3.1 – Поточковий граф 8-точкового Radix-2 DIT ШПФ

Отримане N -точкове ШПФ вимагатиме $N/2$ комплексних множень на етап на деяку ступінь W_N , за винятком останнього 2-точкового етапу ДПФ, де множення не потрібне. Для спрощення розрахунку використовується обчислення «метелик» radix-2 DIT, показане на рисунку 3.2.

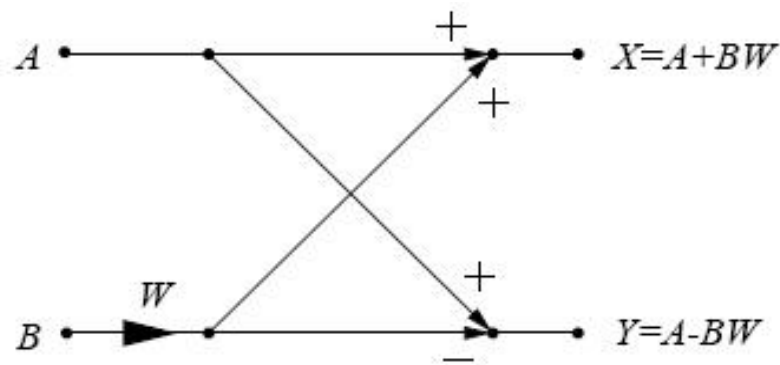


Рисунок 3.2 – Поточковий «метелик» Radix-2 DIT [4]

Існує інший варіант алгоритму швидкого перетворення Фур'є radix-2, а саме алгоритм з проріджуванням за частотою (DIF), що є незначною модифікацією алгоритму проріджування за часом, що був розглянутий раніше. Цей алгоритм також реалізує підхід подібний до «розділяй і володарюй», при цьому вхідна послідовність також розділена на два масиви розміру $N/2$ відліків, а відмінність складається в тому що відліки обираються інакше. В цьому випадку, замість парних і непарних відліків обирають першу та другу половини відліків. Тобто, один оператор суми складається з перших $N/2$ відліків, тоді як інший складається з останніх $N/2$ відліків. Виконавши перетворення згідно рівняння 3.8, отримаємо нову модифіковану послідовність 3.13, а властивість симетрії 3.6, можна відобразити у рівняння 3.14.

$$X(k) = \sum_{n=0}^{N/2-1} x(n) \cdot W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{N/2-1} x\left(n + \frac{N}{2}\right) \cdot W_N^{kn} \quad (3.13)$$

$$X(k) = \sum_{n=0}^{N/2-1} \left(x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right) W_N^{kn} \quad (3.14)$$

Тепер $X(k)$ можна розбити на парні та непарні вибірки. Відповідно до виведемо залежності 3.15 і 3.16:

$$X(k) = \sum_{n=0}^{N/2-1} \left(x(n) + x\left(n + \frac{N}{2}\right) \right), \text{ де } 0 \leq k \leq \frac{N}{2} - 1 \quad (3.15)$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} \left(x(n) - x\left(n + \frac{N}{2}\right) \right), \text{ де } 0 \leq k \leq \frac{N}{2} - 1 \quad (3.16)$$

Використовуючи $N/2$ -точкове ДПФ з проріджуванням по частоті, $X(2k)$ і $X(2k+1)$, процес можна скоротити на $\log_2(N)$ етапів за допомогою $N/2$ обчислень типу метелик radix2 DIF. Ця модифікація алгоритму, так само, як і

алгоритм проріджування в часі, вимагатиме $(N/2)\log_2(N)$ комплексних множень та $N\log_2(N)$ комплексних додавань. Крім того, структура метелика DIF дозволяє легко реалізувати алгоритм як у програмному забезпеченні, так і на апаратному рівні, що робить його універсальним для широкого спектра застосувань у цифровій обробці сигналів. Потік 8-точкового DIF ШПФ і метелик обчислення radix-2 DIF можна побачити на рисунках 3.3 і 3.4.

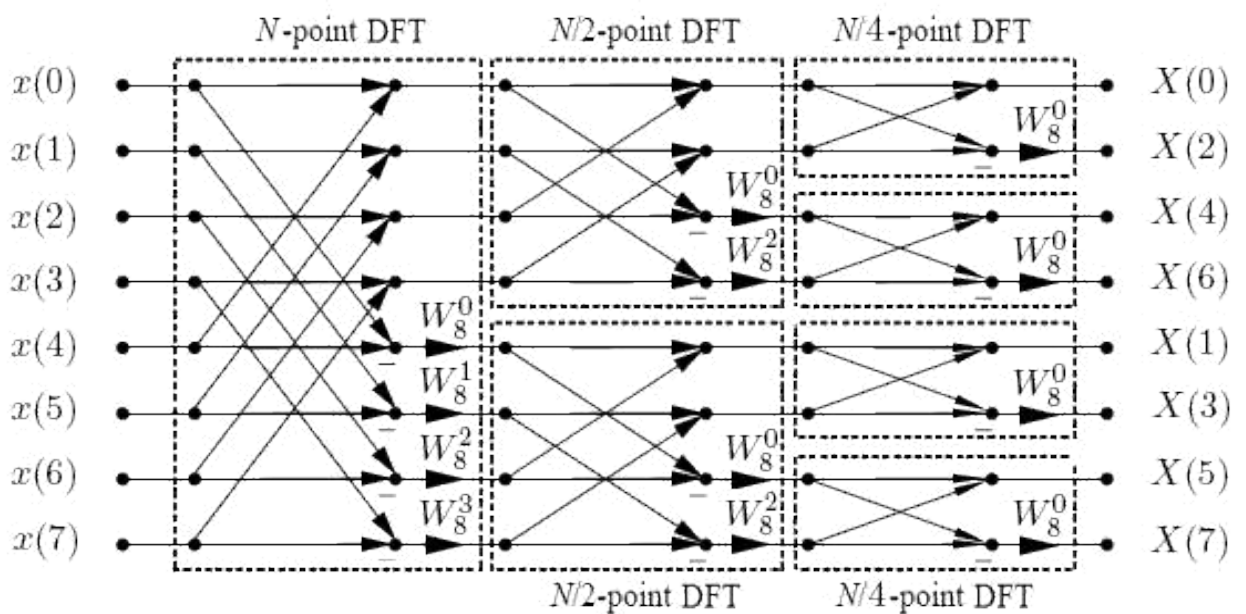


Рисунок 3.3 – Поточковий граф 8-точкового Radix-2 DIF ШПФ

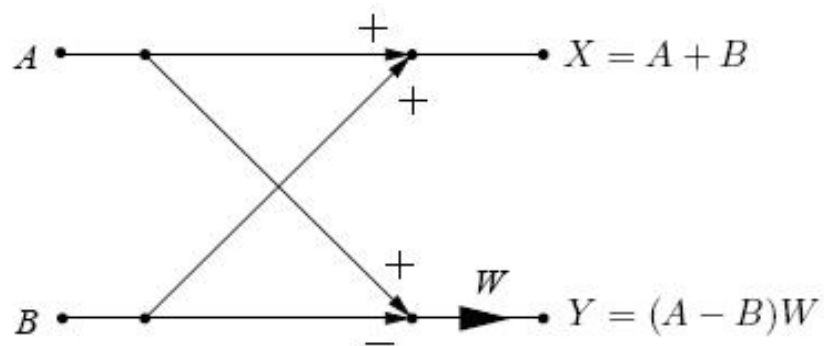


Рисунок 3.4 – Поточковий «метелик» Radix-2 DIF

Можна спостерігати, що вхідна послідовність $x(n)$ йде в природному порядку, тоді як порядок вихідних відліків після децимації змінюється на біт-зворотній порядок. Функція алгоритму залишається такою ж, як у DIT ШПФ, а вхідна послідовність довжини N проріджується $\log_2(N)-1$ разів, поки не зведеться до двоточкового DFT на фінальному етапі, що показує алгоритмічну складність $O(N/\log N)$ операцій. У таблиці 3.1 показано порівняння ефективності обчислення ДПФ і продемонстрованих алгоритмів ШПФ довжини N .

Таблиця 3.1 – Порівняння ефективності ДПФ та ШПФ за кількістю операцій

Довжина перетворення	Кількість ДПФ операцій	Кількість ШПФ операцій	Кількість ДПФ операцій на одну ШПФ операцію
16	256	64	4
128	16400	896	18
1024	1.05×10^6	10240	102
32768	1.07×10^9	4.92×10^5	2185
1048676	1.10×10^{12}	2.10×10^7	52429

Таким чином за допомогою швидкого перетворення Фур'є можливо визначення частоти, а таким чином і виявлення високих піків сигналу серед зашумленого спектру. Вихід ШПФ відображає місце та висоту частотного каналу шляхом визначення максимальної амплітуди кожного спектрального відліку. В такому разі спектральні відліки представлені як частотні інтервали fs/N , де fs — частота дискретизації, а N — вхідне значення ШПФ.

4 РЕАЛІЗАЦІЯ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є НА ПЛІС

4.1 Розробка апаратно-програмної моделі

Апаратно-програмні моделі відіграють центральну роль у забезпеченні високопродуктивної обробки даних, особливо в системах реального часу, де важливі швидкодія, надійність та можливість обробки великих обсягів інформації. Для успішної реалізації задачі необхідно детально спроектувати деталі майбутнього застосунку та означити їх в моделі. У рамках дослідження була запропонована апаратно-програмна модель конвеєрного ШПФ, яка здатна працювати в режимі реального часу і значно підвищувати ефективність обчислень. Така модель поєднує переваги апаратних ресурсів, програмного керування та зручної візуалізації, дозволяючи знизити затримки та підвищити точність обробки сигналів. Для успішного виконання своїх функцій до моделі були висунуті наступні вимоги:

- апаратна платформа має бути обладнана якісним сучасним АЦП для подачі тестових сигналів
- модель має включати блок генерації еталонного сигналу для подачі сигналу без зовнішніх спотворень
- модель має бути спроможним виконувати задачі з аналізу спектру або його частини для проведення оцінення роботи, а тому необхідні методи прорідження та пост обробки результатів ШПФ
- має бути налагоджений надійний зв'язок для керування апаратною платформою без затримок та помилок
- мають бути надані системи виводу даних, їх стиснення та запису
- для виконання натурального експерименту зручно мати систему відображення результатів роботи приладу в реальному часі

Запропонована модель та її інтерфейси зв'язку зображені на рисунку 4.1. За своїм функціоналом реалізація такої моделі буде наближена до аналізатору спектру. В якості апаратної платформи пропонується використовувати ПЛІС з під'єднаним АЦП та зв'язком з комп'ютером двома інтерфейсами: UART та Ethernet. Інтерфейс UART пропонується використовувати у двох напрямках для керування та демонстрації проріджених даних в процесі налагодження експериментальної установки. Інтерфейс Ethernet пропонується використовувати для передачі повного обсягу результатів обчислення ШПФ. При цьому мережеве з'єднання рекомендується використовувати в режимі точка-точка для унеможливлення впливу сторонніх приладів на перебіг експерименту.

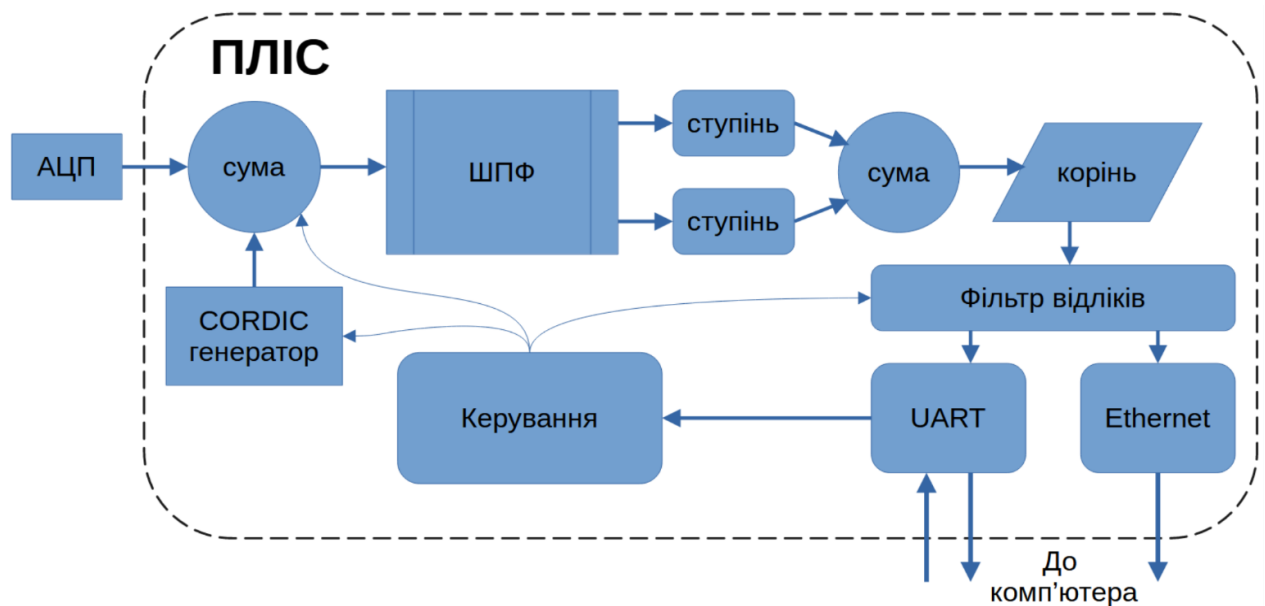


Рисунок 4.1 – Запропонована апаратна структура

В якості ПЛІС обрано Xilinx Artix7. Це актуальна ПЛІС що підтримує усі сучасні технології, має достатній набір портів та добру економічну доступність. ПЛІС фірми Xilinx мають довгу історію та якісну підтримку. Середа розробки Xilinx Vivado реалізує усі етапи проєктування, такі як проєктування, розробка, моделювання, синтез, імплементація, запис коду на

ПЛІС, налагодження та інші. Важливою перевагою Xilinx є розвинена бібліотека ІР-блоків, що призначені для оптимальної реалізації та дозволяють значно пришвидшити розробку дизайну.

Плата носій ПЛІС має бути обладнана всіма необхідними інтерфейсами, зокрема, важливо мати мережеве з'єднання достатньої швидкості, інтерфейс налагодження, можливість реалізувати послідовний інтерфейс передачі даних, доступність достатньої кількості портів для підключення додаткових модулів, зокрема АЦП та пам'яті.

При виборі стандарту мережевого з'єднання слід враховувати можливість передачі всіх даних від ПЛІС без втрат, а також доступність відповідного мережевого обладнання. Таким чином, обсяг даних з АЦП повинен узгоджуватись із пропускнуою здатністю мережевого інтерфейсу. Наприклад, при швидкості передачі даних 100 Мбіт/с і 8-бітному АЦП тактова частота може досягати 12,5 МГц, що відповідає смузі пропускання 6,25 МГц за теоремою Котельникова. Однак, такі параметри є недостатніми для більшості сучасних застосувань. Мережеве обладнання зі швидкістю 10Гбіт/с дозволяє передавати більше даних, із вищою частотою та розрядністю, але воно не є широко розповсюдженим і економічно не вигідне. Крім того, таке обладнання висуває високі вимоги до якості тактових сигналів, що генеруються в ПЛІС, і до кабельного обладнання для мережевого зв'язку. Оптимальним варіантом буде використання мережевого інтерфейсу зі швидкістю 1 Гбіт/с. Більшість сучасного користувацького обладнання підтримує цей стандарт, що сприятиме економічній ефективності проєкту. Крім того, цей стандарт забезпечує достатній запас пропускнуої здатності для роботи з багатьма сучасними АЦП, що дозволяє зменшити технічні ризики та забезпечити надійність передачі даних.

Передача даних по мережевому інтерфейсу Ethernet відбувається за допомогою протоколу UDP без підтвердження цілісності. Таке рішення прийнято для полегшення розробки як програмної так і апаратної частини, бо

не потребує встановлення та підтримки сесії мережевого зв'язку. Дані від апаратної частини об'єднані в слова розміру три байта, а так як довжина перетворення не може бути передана одним пакетом, то вона розділена на пакети по 256 слів в кожній. Кожний пакет має мережеві заголовки та префікс, формат яких описаний у таблиці 4.1.

Таблиця 4.1 – Формат префіксу для передачі даних по протоколу Ethernet

Байт	Функціональний опис	Значення
0	Номер пакету	0-255
1	Номер фрейму	0-31
2-3	Номер відліку з якого починаються дані в пакеті	0-8191
4-5	Номер відліку яким закінчуються дані в пакеті	0-8191

Таким чином відомо скільки додаткових даних необхідно буде передавати процесі роботи протоколу. Наприклад, для передачі 256 слів реальних даних і можна розрахувати відсоток пропускної спроможності інтерфейсу, що буде дійсно використаний. Максимальний можливий потік даних складає $12\text{bit} \times 65\text{MSPS} = 780\text{Mbit/s}$. Для передачі 256 слів по 3 байти, необхідно передати 7 байт фізичної преамбули, 1 байт завершення преамбули (SFD), 14 байт Ethernet заголовку, 20 байт IPv4 заголовку, 8 байт UDP заголовку та 6 байт префіксу. Таким чином, для передачі $256 \times 3 = 768$ байт корисних даних має бути передано пакет сумарного розміру $7+1+14+20+8+6+768=824$ байти, що означає що додаткове навантаження на мережу буде $(824/768-1)/100 \approx 7,3\%$, або $\sim 836\text{Mbit/s}$ загального навантаження при теоретичній пропускній спроможності в 1Gbit/s , тобто задовольняє заданим вимогам. Цей розрахунок не включає використання пропускної спроможності для передачі пакетів що необхідні для функціонування мережевого з'єднання, таких як ARP та SNMP.

Послідовний інтерфейс передбачається для первинного налаштування мережевого інтерфейсу та надійного керування експериментальною установкою. Тобто важливо забезпечити простоту та надійність керування. Стандарт UART повністю відповідає цим потребам, а його надійність зворотнопропорційна його швидкості. Таким чином при виявленні помилок передачі можна зменшити швидкість передачі що не є критичною, а підвищити критичний параметр надійності.

Дані що передаються від програмної частини до апаратної передаються лише по інтерфейсу UART і призначені для керування апаратною частиною. Керування відбувається при записі по послідовному інтерфейсу керуючого слова. Для простоти керування кожний керуючий вплив передає керуюче слово цілком. Це спростить задачу приймача UART реалізованого в апаратурі до простого підтвердження цілісності переданих керуючих впливів та їх своєчасне зберігання. Таким чином, відсоток використання пропускну спроможності в напрямку передачі від комп'ютера до ПЛІС дуже низький, а тому є сенс не економити пропускну спроможність та не ускладнювати систему декодування на приймальній стороні. Натомість необхідно підвищити надійність передачі для чого впроваджено формат керуючого слова з можливістю подальшої інтеграції додаткових функцій. Формат керуючого слова наведено у таблиці 4.2.

Таблиця 4.2 – Формат керуючого слова для передачі по протоколу UART

Біт	Функціональний опис	Значення
0	Старт роботи апаратного автомату передачі даних по протоколу UART та протоколу Ethernet, якщо він активований у біті 7, якщо біт 1 встановлено в нуль то буде передана лише одна вибірка заданого розміру	1 – старт 0 – стоп
1	Старт безперервної передачі даних, передачу можливо зупинити лише командою стоп передану у бітах 1 та 0	1 – старт 0 – стоп

2	Використання зовнішнього для ПЛІС джерела даних (АЦП), якщо біти 2 та 3 активні одночасно то дані з обох джерел діляться навпіл та підсумовуються	1 – активно 0 – не активно
3	Використання внутрішнього для ПЛІС джерела даних (генератор еталонної синусоїди), якщо біти 2 та 3 активні одночасно то дані з обох джерел діляться навпіл та підсумовуються	1 – активно 0 – не активно
6-4	Керування набором даних що передаються	111 – дані спектру (корінь суми квадратів) 101 – дійсна складова результату ШПФ 001 - уявна складова результату ШПФ 000 – дані що поступають в блок ШПФ
7	Передача даних по мережевому інтерфейсу Ethernet для їх запису на носій	1 – активно 0 – не активно
11-8	Зарезервовано для керування амплітудою внутрішнього еталонного генератора синусоїди	-
15-12	Зарезервовано без спеціального призначення	-
31-16	Крок фази синусоїди за один такт що контролює частоту вихідного сигналу еталонного генератора; повний оборот фази складає 51470 кроків, значення за замовчуванням - 800	1 - 25735
45-32	Номер відліку з якого починається накопичення даних для передачі, відліки з порядковими номерами нижче будуть відфільтровані, значення за замовчуванням - 0	0 - 8191
47-78	Зарезервовано для вирівнювання	-
61-48	Номер відліку до якого відбувається накопичення даних для передачі, відліки з порядковими номерами вище будуть відфільтровані, значення за замовчуванням - 8191	0 - 8191
63-62	Зарезервовано для вирівнювання	-

UART інтерфейс передає дані безперервно. У разі відсутності даних передаються нульові значення. Для виділення керуючого слова використовуються 8-бітні контрольні слова (префікс і постфікс), що допомагають виділити пакет у потоці байт і підтвердити цілісність даних. У протоколі, реалізованому через UART, передаються пакети з префіксом і даними. Формат префіксу наведено в таблиці 4.3.

Таблиця 4.3 – Формат префіксу для передачі даних по протоколу UART

Байт	Функціональний опис	Значення
0-1	Початкові байти	0x9B, 0xFA
2-3	Номер відліку з якого починаються дані в пакеті	0-8191
4-5	Номер відліку яким закінчуються дані в пакеті	0-8191
6-8	Байти завершення	0x0E, 0xBF, 0x9A

Якщо будь-який з байтів префіксу не відповідає поточними налаштуваннями, встановленими в застосунку, то такий пакет розцінюється як пошкоджений та ігнорується. Префікс складається з непарної кількості байт, а саме дев'яти. Ця особливість була закладена при розробці протоколу даних оптимізованого під розрядність АЦП 12 біт. Розмір слова в даному протоколі становить 3 байта, тобто 24 біти. В залежності від даних що передаються три байта відображають або два відліки АЦП (12+12 біт), або 24 біти результату перетворення Фур'є (дійсної або уявної складової), або 24 біта магнітуди спектра (корінь суми квадратів). Це не впливає на точність обчислень але дозволяє значно спростити алгоритм автомата в ПЛІС, що керує передаванням даних по послідовному протоколу.

Для запису передбачається використовувати комп'ютер з авторським програмним забезпеченням. Означений застосунок надасть можливість легко передавати керуючі команди до апаратної частини, забезпечувати

візуалізацію при налаштуванні апаратної частини в процесі підготовки програмно-апаратного комплексу до експерименту, та збереження отриманих даних. Пріоритетнішими є платформонезалежні рішення що нададуть гнучкості експериментальній установці. На основі означених вимог, для реалізації застосунку доцільно обрати мову Python. Ця мова забезпечить гнучкість, багатоплатформеність та достатню швидкодію. Мова Python є актуальною мовою, що невинно розвивається. Великий набір відкритих бібліотек, а також доступні та зручні середовища розробки забезпечують швидкість, простоту та надійність результуючого застосунку.

В якості вхідних даних дозволяється використовувати будь який сигнал, електричні параметри котрого не виходять за параметри АЦП. Для повторюваності і простоти оцінки доцільніше використовувати генератор сигналів що має достатню стабільність, точність і надійність, і опціонально має можливість модулювання сигналу або зміни його форми.

4.2 Моделювання апаратної частини

Моделювання виконувалось в середі Modelsim 10.7, що була налаштована для безпосередньої роботи в парі з САПР Vivado. Таким чином була досягнута простота використання для розробника, були спрощені процеси компіляції, втілена інкрементальна компіляція, що зменшило час виконання кожної ітерації. Таким чином було пришвидшено кожен етап створення реалізації моделі та скорочено загальний час розробки.

Модель апаратної частини була виконана згідно попередньо спроєктованої структури. Середою проєктування обрано Xilinx Vivado 2023.2. Також були використані декілька Xilinx IP-core для пришвидшення проєктування, а обв'язка та тестове оточення були виконані мовою SystemVerilog. В додатку Б можна ознайомитись з загальним виглядом

синтезованої схеми верхнього рівня. Верхній рівень проєкту був розбитий на декілька блоків, головні з яких:

- main_pll – блок генерації тактових частот
- sine_gen – генератор тестової синусоїди
- fft_ctrl – блок керування блоком ШПФ
- input_inst – блок перетворення форматів вхідних даних та підсумовування даних від різних джерел
- dsp_unit – блок обробки, що включає блок ШПФ, блоки піднесення у ступінь, блок кореню та фільтрування вихідних відліків
- uart_inst – блок буферизації та взаємодії за протоколом UART
- incoming_dec – блок декодування вхідних команд
- eth_inst – блок генерації та буферизації мережевих пакетів, а також контролю мережевого інтерфейсу

Програмна частина не використовувалась в процесі моделювання, а її керуюча складова була частково імітована в тестовому оточенні. Тестове оточення було виконано мовою SystemVerilog і виконувало наступні функції:

- генерація зовнішнього для дизайну тактового сигналу 50МГц у відповідності до обраної платформи
- підтримка послідовного протоколу даних UART
- передача керуючого слова у відповідності до спроектованих протоколів передачі даних
- передача послідовності відліків, що імітують дані з АЦП

Моделювання мало на меті продемонструвати цілісність потоку даних, можливість досягнення заданих параметрів та надійність протоколів керування та передачі даних. Функціональні блоки, що відповідають за контролери мережевого зв'язку (Ethernet PHY) потребують значного модельного часу для своєї ініціалізації, тому моделювання блоку передачі по мережі Ethernet не відбувалось. Натомість був перевірений конвєр формування мережевих пакетів та їх буферизація.

Наведено приклади моделювання найважливіших частин дизайну. На рисунку 4.2 зображено моделювання пристрою з початку його роботи, що видно по активному сигналу скиду (`rst_n`). Тактовий сигнал генерується незалежно від сигналу скиду, бо в апаратній реалізації він надходить від зовнішнього для ПЛІС джерела розташованого на платі і генерується постійно. Після ввімкнення починається передача керуючого слова по інтерфейсу UART. Керуюче слово було успішно декодовано, що видно по тому що надійшли відліки з внутрішнього генератора еталонного сигналу. Керуюче слово налаштовує еталонний генератор на генерацію синусоїди частотою 1,01 МГц.

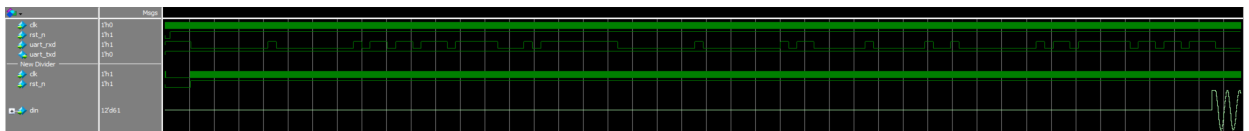


Рисунок 4.2 — передача керуючого слова

Згенерований сигнал надходить у блок перетворення Фур'є. Моделювання розрахунку одного ШПФ зображено на рисунку 4.3.

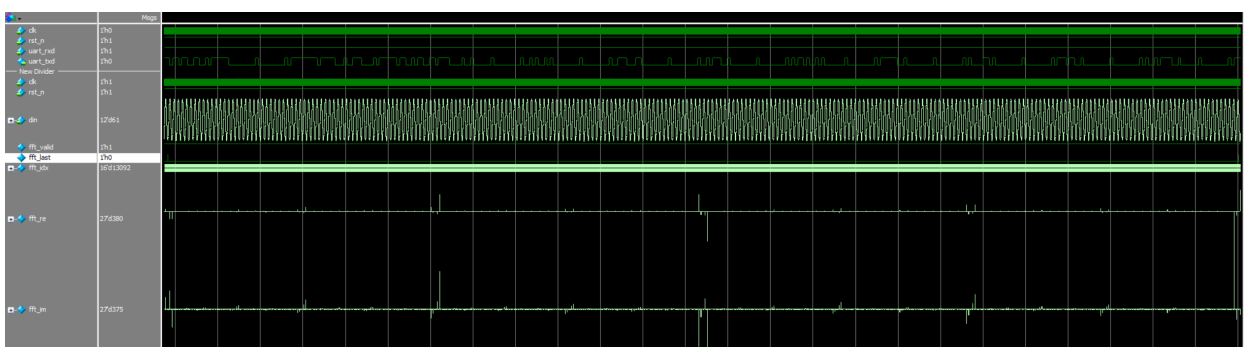


Рисунок 4.3 — моделювання розрахунку однієї вибірки ШПФ

Одна вибірка перетворення починається і закінчується сигналом `fft_last`, також вхідний сигнал синусоїди `din`, вихідні результати дійсної (`fft_re`) та уявної (`fft_im`) складових, а також порядковий номер вхідного

відліку (`fft_idx`). Порядковий номер вихідного відліку дуже важливий, бо для економії пам'яті результати надходять в біт-зворотньому порядку.

При детальному вивченні результатів обчислення ШПФ бачимо що найбільші по модулю значення результатів відповідають частотному каналу номер 255, що зображено на рисунку 4.4. Це відображає максимум амплітуди сигналу в частотній області після ШПФ. Для підтвердження коректності моделювання було проведено розрахунок для цього максимуму амплітуди. Зокрема, було перевірено відповідність частоти синусоїдального сигналу, що генерується внутрішнім генератором згідно з керуючим словом. Ширина одного частотного каналу розраховується як $65 \text{ МГц} / 16384 = 3,97 \text{ кГц}$. Таким чином, частота для пік-каналу, який відповідає 255-му каналу, становить $3,97 \text{ кГц} \times 255 = 1,01167 \text{ МГц}$. Це значення близьке до частоти еталонного генератора, яка складає $1,01 \text{ МГц}$.

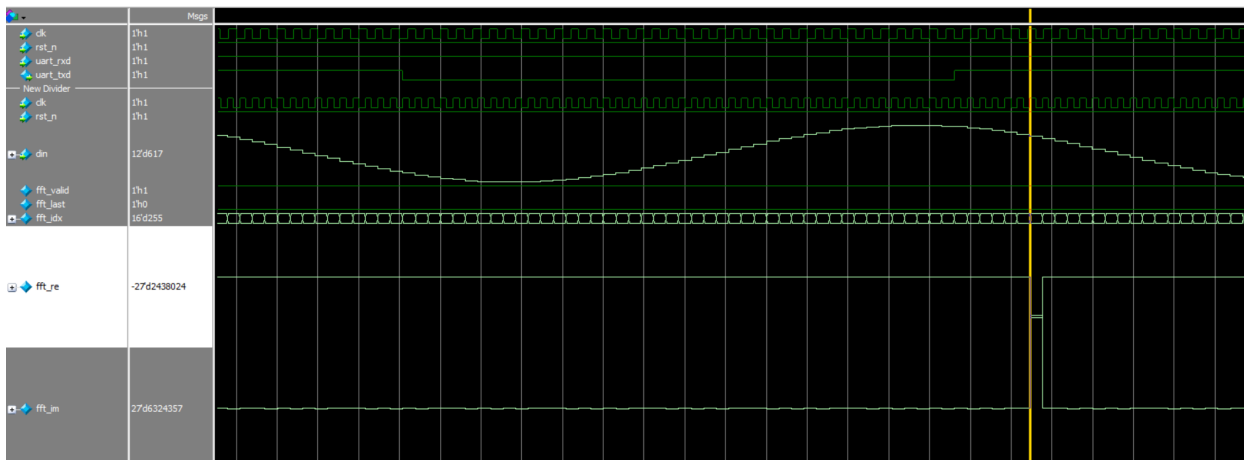


Рисунок 4.4 — максимум по модулю вибірки ШПФ

Таким чином, результати моделювання підтверджують правильність роботи апаратної реалізації, зокрема коректність функціонування інтерфейсів та протоколів передачі, механізмів керування, алгоритму ШПФ та обчислень після самого перетворення. Моделювання демонструє високу точність та відповідність до теоретичних розрахунків, що свідчить про правильну інтеграцію всіх компонентів системи.

4.3 Імплементация моделі та використані ресурси

Імплементация — це невід’ємна частина циклу розробки в ПЛІС. За допомогою засобів, що надаються САПР Xilinx Vivado була виконана імплементация дизайну моделі в ПЛІС. На цьому етапі в дизайн були зроблені, що не вплинули на алгоритм обробки впровадженої моделі, однак дозволили оптимізувати дизайн для стабільної роботи дизайну та досягнення заданих часових параметрів схеми. Опорна тактова частота плати становить 50МГц, але більшість дизайну працює на частоті дискретизації, а саме 65МГц для спрощення виконання конвеєрної обробки. Частина дизайну що відповідає за роботу з мережею Ethernet працює на частоті 125МГц, що відповідає частоті 1Gbt ethernet РНУ. Під час ітеративної розробки було проведено ретельний аналіз та оптимізацію використаних апаратних ресурсів, що дозволило досягти ефективного використання доступних блоків та зменшити їх загальне споживання (Рисунок 4.5).

Resource	Utilization	Available	Utilization %
LUT	5104	20800	24.54
LUTRAM	1009	9600	10.51
FF	8864	41600	21.31
BRAM	25	50	50.00
DSP	63	90	70.00
IO	33	250	13.20
BUFG	6	32	18.75
MMCM	2	5	40.00

Рисунок 4.5 — використані ресурси ПЛІС

Внутрішня блочна пам’ять BRAM використана на 50%, тобто є важливим ресурсом для реалізованої моделі. Блочна пам’ять використовується як для збереження проміжкових результатів розрахунку

ШПФ, так і для буферизації даних для передачі інтерфейсами UART та Ethernet. Збільшення довжини перетворення Фур'є у цьому проєкті можливе, але зіштовхнеться з ускладненими розташуванням та маршрутизацією (place and route) дизайну ПЛІС. Ще одним ресурсом цієї реалізації що привертає увагу є блоки DSP. Вони були використані як в блоці ШПФ так і в блоках піднесення ступеню. Їх можна замінити логічними комірками, тому не слід зважати на високий відсоток використання. Їх використання обумовлено спрощенням маршрутизації дизайну, що використовує такі блоки.

Додатково середа розробки надає звіт з очікуваної використовуваної потужності та очікувану робочу температуру кристалу ПЛІС. Звіти приведені на рисунках 4.6 та 4.7.

Total On-Chip Power:	0.535 W
Junction Temperature:	26,5 °C
Thermal Margin:	58,5 °C (20,7 W)
Effective θ_{JA} :	2,8 °C/W

Рисунок 4.6 — очікувана використовувана потужність ПЛІС

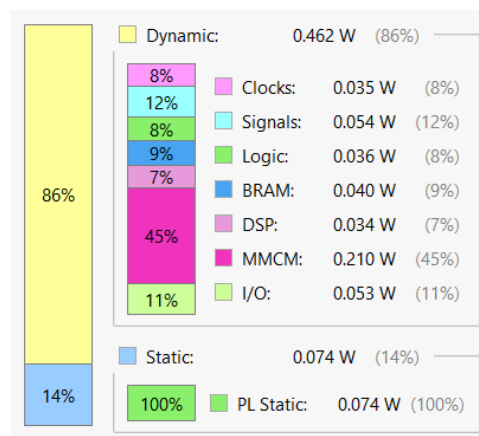


Рисунок 4.7 — деталізація потреб потужності частинами ПЛІС

Згідно детального звіту споживання наведеного на рисунку 4.7 можемо оцінити зміну споживання при переході дизайну ПЛІС від режиму

очікування до активного використання. У зв'язку з тим, що стан живлення вхідних/вихідних портів (I/O), системи генерації (MMCM) та розподілу тактових сигналів (Clocks) не вимикаються в режимі очікування, то можна оцінити зміну споживання як максимум споживання сигналів (Signals), логічних блоків (Logic), блоків пам'яті (BRAM) та множення (DSP). Тобто при переході з одного режиму в інший споживання має змінюватись не більше ніж на $54+36+40+34=164$ мВт. У цій реалізації потужність не є критичною, але надає можливість оцінити ефективність використання ресурсів ПЛІС при впровадженні моделі.

4.4 Програма керування та відображення

Програмна частина запропонованої моделі була виконана мовою Python. Вона відповідає вимогам висунутим до програмного забезпечення на етапі проєктування та забезпечує простоту розробки, надійність, багатоплатформенність та швидкодію.

Додаток було розбито на декілька частин відповідно до їх функціонального призначення, а саме:

- взаємодія з графічний інтерфейсом
- взаємодія з послідовним інтерфейсом UART
- відображення графіків в реальному часі
- взаємодія з мережевим інтерфейсом Ethernet
- компресія та збереження мережевих даних

Для розробки використовувались лише відкриті бібліотеки такі як `numpy`, `matplotlib`, `threading`, `gzip`. Графічний дизайн додатку виконано за допомогою бібліотеки `PyQt6` за допомогою додатка `PyQt6 UI code generator`, в який було вбудовано вікно для побудови графіку та елементи керування, як зображено на рисунку 4.8.

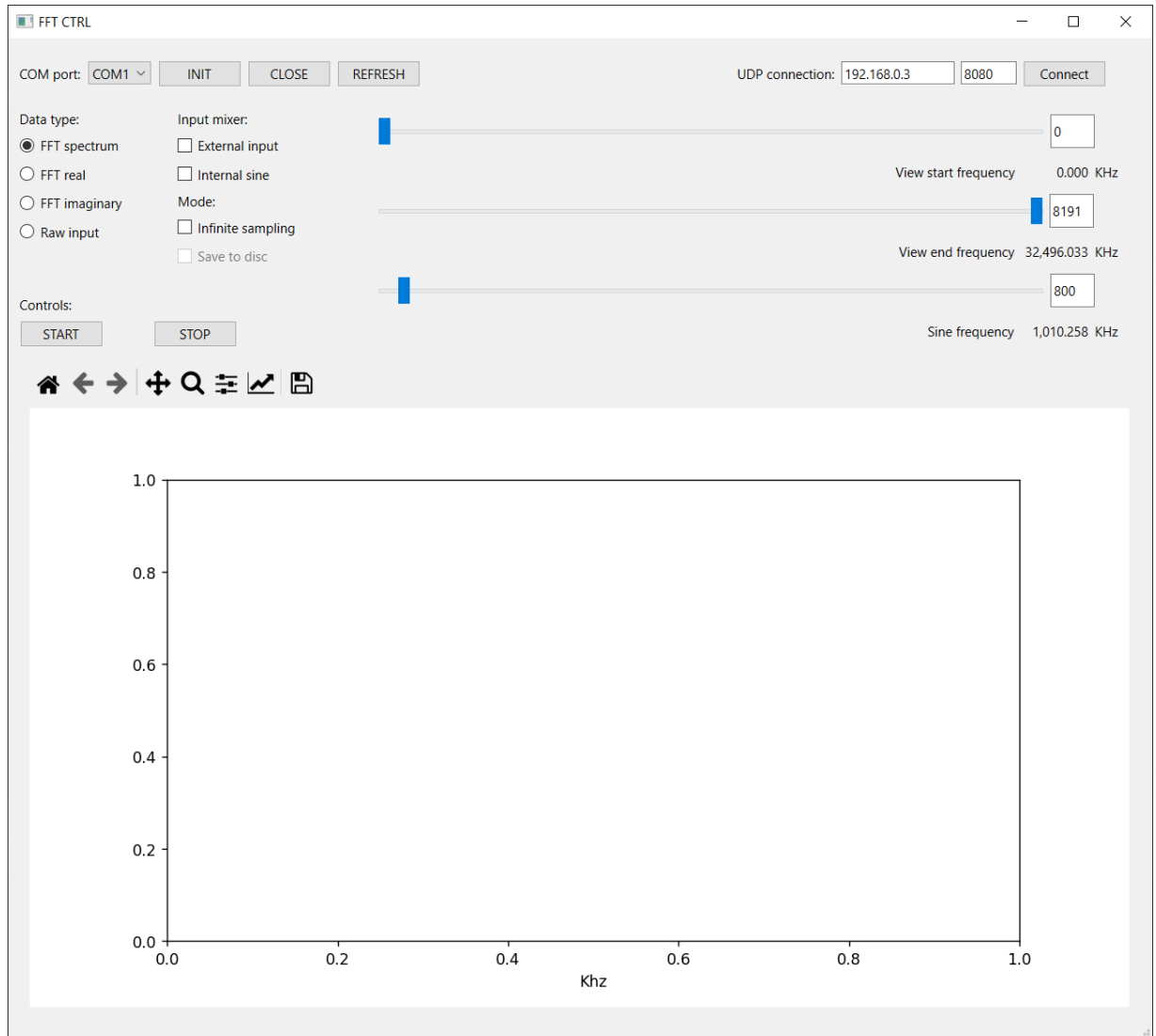


Рисунок 4.8 — зовнішній вигляд застосунку на Python

Для підвищення швидкодії додатку на багатопроцесорних комп'ютерах вищезначені процеси виконуються у незалежних потоках за допомогою бібліотеки `threading`.

Так як відліки з блоку ШПФ надходять в біт-зворотньому порядку у додатку виконано сортування переданих даних у натуральний порядок. Відповідно тому що у модель впроваджена фільтрація вихідних даних ШПФ для передачі обмеженої частини частотних каналів, то в додатку інтегрована система адаптивного відображення відповідної частини спектру. Ця частина також безпосередньо пов'язана з підтвердженням цілісності отриманих даних на основі префіксів та постфіксів які описані у розділі 4.1.

5 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПРАКТИЧНОЇ ЦІННОСТІ

5.1 Натурні випробування реалізованої апаратно-програмної моделі

Для проведення випробувань був створений тестовий стенд, що складається з наступних частин:

- ПЛІС Artix7-35t з приєднаною до неї АЦП AD9226 (Рисунок 5.1)
- генератор безперервних сигналів різної форми FG-100, що подає сигнали на вхід АЦП (Рисунок 5.2)
- користувацький комп'ютер, що використаний для запуску розробленого програмного забезпечення (Розділ 4.4)

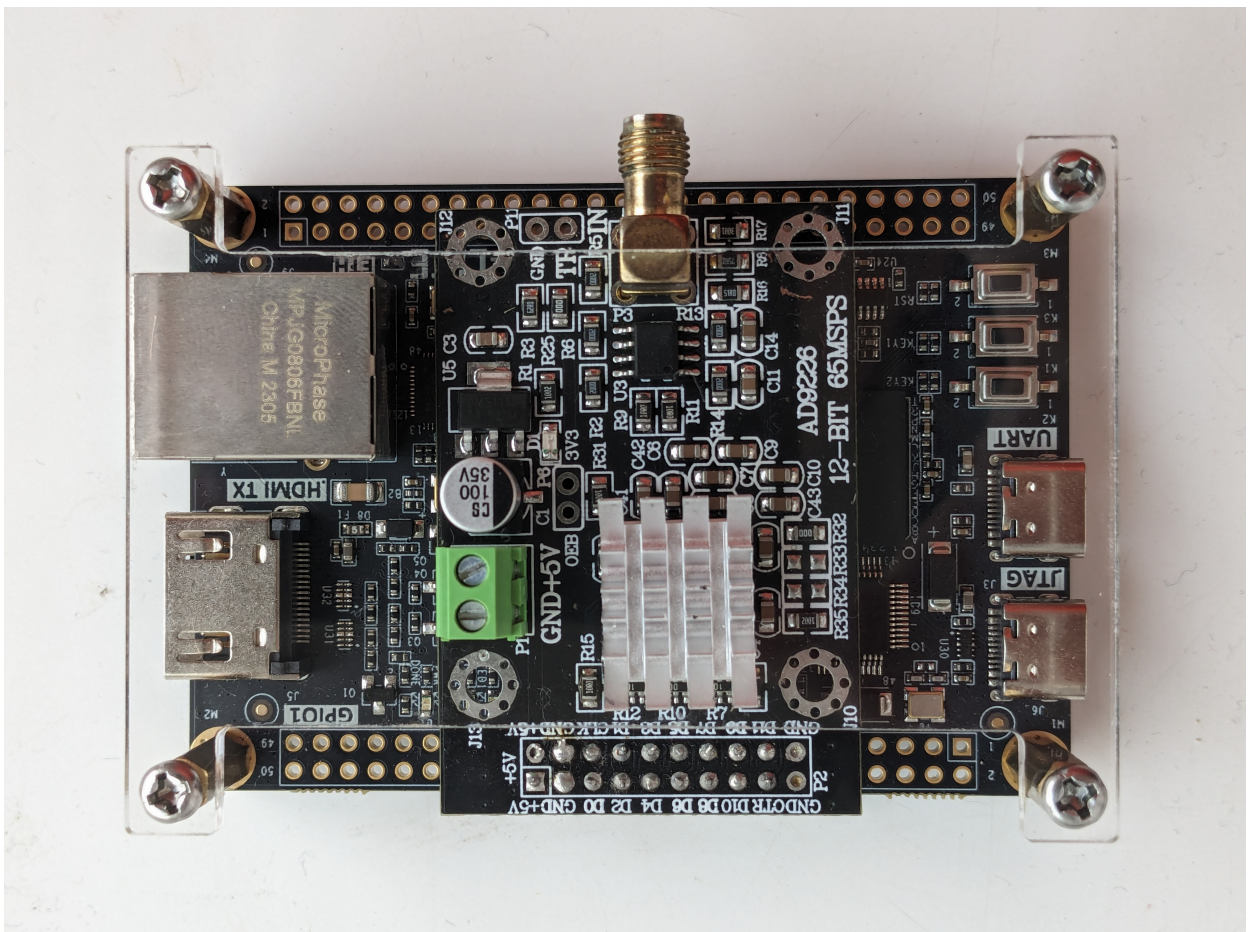


Рисунок 5.1 — Плата носій з АЦП

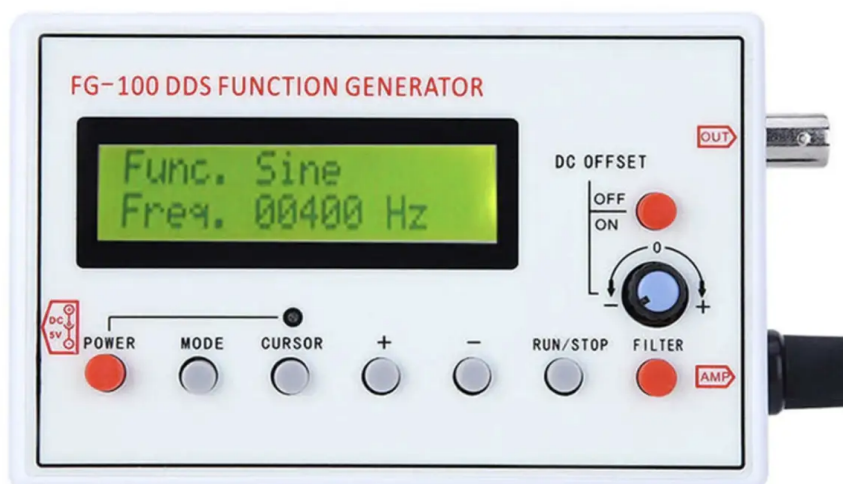


Рисунок 5.2 — Зовнішній генератор сигналів

Загальний вигляд на етапі експерименту зображено на рисунку 5.3, що представляє знімок екрану користувацького комп'ютера в процесі проведення одного з випробувань.

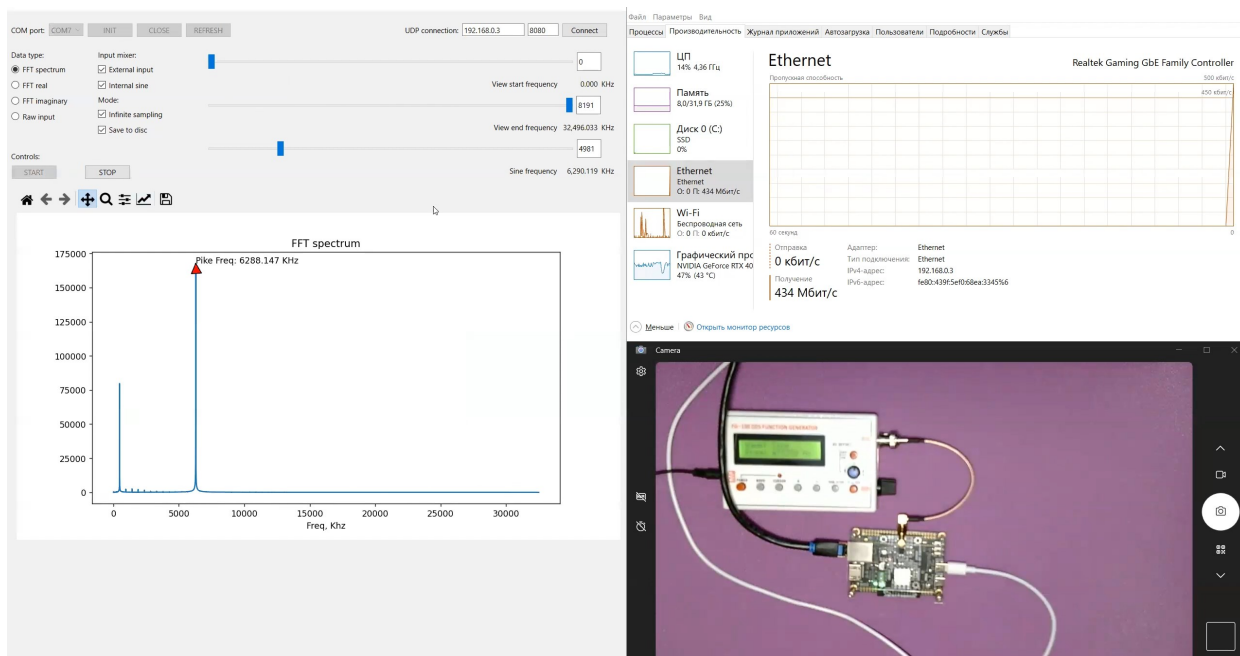


Рисунок 5.3 — знімок екрану під час випробування

На рисунку 5.3 зліва видно вікно програми керування в процесі роботи; справа вгорі видно графіки використання ресурсів користувацького

комп'ютера, зокрема швидкість передачі даних по мережевому інтерфейсу Ethernet; справа внизу передається зображення з камери, що направлена безпосередньо на ПЛІС та під'єданого до неї генератору сигналів.

В програмі керування обрано відображення спектру сигналу (Data type - FFT spectrum), що отримано від ПЛІС по послідовному інтерфейсу (COM7). Внутрішній генератор синусоїди реалізований в ПЛІС налаштовано на 6290,119 кГц (Sine frequency), а відповідний йому пік позначено на діаграмі зі значенням 6288,147 кГц. Внутрішній суматор сигналів (Input mixer) налаштовано на використання даних як з внутрішнього генератора синусоїди (Internal sine) так і даних отриманих з АЦП (External input). На діаграмі спектру пік меншої амплітуди близько 457 кГц відповідає сигналу від зовнішнього генератора, а його амплітуда зменшена з ціллю більш простої ідентифікації піків відносно їх джерела.

У програмному забезпеченні було реалізовано керування режимами роботи (Mode), що включає два основних режими: режим постійного оновлення діаграми в реальному часі (Infinite sampling), де дані передаються по послідовному інтерфейсу, та режим запису даних від ПЛІС без втрат (Save to disc), де дані передаються по мережевому інтерфейсу. У режимі Infinite sampling програма постійно оновлює діаграму, відображаючи поточні значення в реальному часі, що дозволяє здійснювати моніторинг процесу обробки сигналу без затримок. Для забезпечення передачі даних без втрат було виконано з'єднання з ПЛІС через мережевий інтерфейс Ethernet (UDP connection). Це дозволяє здійснювати стабільну та швидку передачу великих обсягів даних без ризику їх втрати. У верхньому правому куті інтерфейсу програми відображається початок передачі даних, а значення початкової швидкості передачі ~ 450 Мбіт/с, що підтверджує активну та ефективну передачу даних від ПЛІС до користувачького комп'ютера. Ці дані проходять через процес стиснення, що значно зменшує обсяг передаваних даних, а потім записуються на диск без втрат в реальному часі.

Протягом експериментальних сеансів не було виявлено жодних проблем із надійністю, швидкодією або функціональністю системи. Більше того, ці тести підтвердили коректність спроектованого алгоритму цифрової обробки сигналів, а також надійність систем керування, передачі, збереження та відображення даних. Тестування показало, що система здатна стабільно працювати при високих швидкостях передачі даних та забезпечувати безперервну обробку сигналів без втрат інформації.

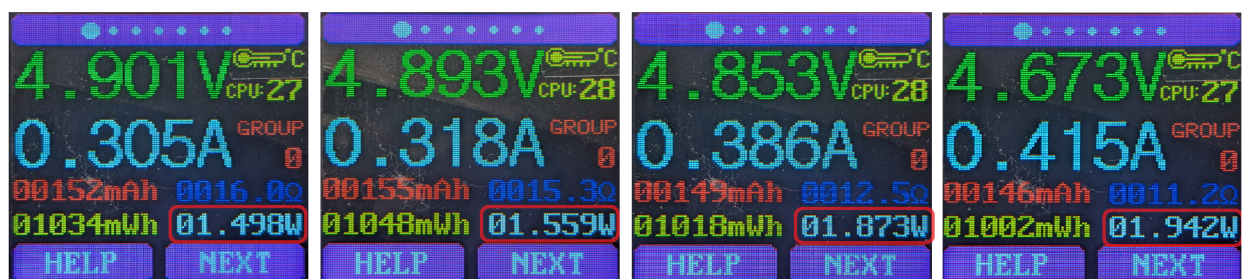
5.2 Аналіз науково-практичних результатів дослідження

На прикладі проведених випробувань було продемонстровано функціональність впровадженої апаратно-програмної моделі. У демонстраційному прикладі була помічена різниця частоти згенерованого сигналу та його відображення на спектрі, а саме 6290,119 кГц проти 6288,147 кГц, тобто неточність склала $6290,119 \text{ кГц} - 6288,147 \text{ кГц} = 1,972 \text{ кГц}$. В розділі 4.2 була розрахована ширина одного частотного каналу, що склала 3,97 кГц. Тобто похибка складає половину розподільної здатності впровадженої системи, що відповідає допустимій похибці вимірювальної техніки.

Було проведено оцінення системи з точки зору відповідності роботи в реальному часі. Так як усі дані обробляються потоковим ШПФ без втрат, то для оцінки роботи в реальному часі слід вирізняти затримку на відображення, що залежить від швидкості послідовного інтерфейсу, та затримку на передачу повних даних без втрат, на яку впливає швидкість передачі по мережевому інтерфейсу. Швидкість передачі по послідовному інтерфейсу становить 1,5 Мбод/с або 150кБ/с. Розмір однієї повної вибірки 16384, тобто затримка її захвату в АЦП, обробки та накопичення для передачі становить $16384 \times 2 = 32768$, що при частоті дискретизації 65MSPS становить $5,04 \times 10^{-4} \text{ с}$. При цьому на вихід подається лише перша зона Найквіста, тобто половина

цієї вибірки, де кожен відлік має розмір 3 байти. Таким чином, повна передача складає $8192 \times 3 = 24576$ байт, а передача по послідовному інтерфейсу займає $24576 / 150000 = 0,164$ с. Тобто час обробки в ПЛІС та програмному забезпеченні не вносить значної затримки, а сумарна затримка відображення становить 0,17с. Швидкість передачі по мережевому інтерфейсу складає 1Гбіт/с, розмір повної вибірки не змінюється і становить 24576 байт, а згідно з розділом 4.1 додаткові дані, що необхідно передати по мережевому інтерфейсу складають 7.3%. Тоді час передачі по мережевому інтерфейсу складає $24576 \times 1,073 / 125000000 = 2,1 \times 10^{-4}$ с, а сумарна затримка на збереження даних становить $2,1 + 5,04 = 7,14 \times 10^{-4}$ с. Розраховані параметри затримки дійсно малі і демонструють спроможність роботи впровадженої апаратно-програмної моделі в реальному часі.

Окремо проведено оцінку споживаної потужності апаратної платформи за допомогою проміжного вимірювача потужності АТОРCH А3. При цьому неможливо вирізнити споживання ПЛІС, АЦП, контролера мережі чи втрати на дротах живлення. Тому проведено відносну оцінку споживання в стані очікування та в робочому стані, а додатково при використанні мережі Ethernet чи без такої. Результати вимірювань на рисунку 5.4а-г.



а)

б)

в)

г)

Рисунок 5.4 — результати вимірювання споживаної потужності

На рисунку 5.4 результати відповідають наступним станам апаратної реалізації. а,б — без використання Ethernet, в,г — з використанням Ethernet,

а,в — режим очікування, б,г — в активному режимі використання. Значення споживаної потужності наведено в правому нижньому куті екрану приладу. З вимірювань зроблено висновок, що при активній роботі ПЛІС використовується на 59-69мВт більше потужності. В розділі 4.3 було розраховано, що активна робота ПЛІС може збільшити споживання до 164мВт, тобто вимірні значення входять в розрахункові допуски.

Реалізація апаратно-програмної моделі відповідає закладеним принципам і використовує менше половини доступних ресурсів ПЛІС. Це забезпечує можливість подальших модифікацій, які потребуватимуть додаткових ресурсів, без зміни довжини перетворення чи протоколів зв'язку та передачі даних. Дана реалізація може застосовуватись для аналізу сигналів у реальному часі в одно-канальному режимі з можливістю збереження даних для подальшої обробки. Також вона підходить для впровадження алгоритмів цифрової обробки сигналів у проєктах, де критично важливий час обробки даних.

Реалізована модель має значний потенціал розвитку, зокрема як базова платформа для вирішення завдань підвищення точності, ефективності та зменшення ресурсоємності. Вона придатна для простого горизонтального масштабування як за кількістю каналів, так і за розмірністю даних і частотами обробки даних.

ВИСНОВКИ

У цій магістерській атестаційній роботі досліджено алгоритм швидкого перетворення Фур'є, спроектовано та реалізовано модель апаратно-програмної системи для обчислення спектру сигналу в реальному часі.

В дослідженні окреслені головні виклики застосування ШПФ, зокрема забезпечення високої швидкості обробки, енергоефективності, точності обчислень та оптимального використання ресурсів. Було запропоновано апаратно-програмну модель з конвеєрною архітектурою, яка сприяє підвищенню ефективності та швидкості обробки цифрової інформації незалежно від галузі застосування.

Для реалізації запропонованої моделі виконано аналіз і планування апаратної бази, обрано мову програмування для створення зручного програмного забезпечення, а також розроблено надійні протоколи обміну даними між програмною та апаратною частинами.

Апаратну частину моделі реалізовано мовою SystemVerilog. Для перевірки функціональності проведено моделювання за допомогою тестового середовища. В обраній САПР виконано імплементацію моделі в ПЛІС, а також оцінено використання ресурсів обраної платформи. Для взаємодії з апаратною частиною створено застосунок мовою Python, який відповідає вимогам висунутим на етапі проєктування.

Експериментальні випробування підтвердили надійність розроблених протоколів зв'язку та можливість роботи системи в реальному часі. Проведено оцінку точності реалізованої апаратно-програмної моделі, а також використаних апаратних ресурсів і енергоспоживання.

Використання спеціалізованої апаратної платформи ПЛІС дозволило досягти високої швидкості обробки та передачі даних, що забезпечило можливість виконання ШПФ у реальному часі. Цього вдалося досягти

завдяки конвеєрній архітектурі, розробці та оптимізації дизайну під конкретні задачі та платформу.

На етапі проєктування були закладені можливості для масштабування моделі, як то збільшення довжини перетворення, додавання додаткових каналів даних або обробки крос-спектрів. Результати дослідження можуть бути використані для аналізу спектру сигналу, можуть бути інтегровані в існуючі проєкти або стати основою для подальшого дослідження з метою перевірки та розвитку алгоритмів цифрової обробки сигналів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tsui, James Bao-Yen. Digital Techniques for Wideband Receivers. Second Edition. Norwood, MA: Artech House, Inc., 2001.
2. Cooley, James W., Tukey, John W. An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics Of Computation. 1965, Vol. 19, p. 297-301.
3. Despain, Alvin M. Very Fast Fourier Transform Algorithms Hardware for Implementation. IEEE Trans. On Computers. 1979, 28(5), p. 333-341.
4. Patli, Sudhir. Hardware Implementation of a Radix-4 Fast Fourier Transform Technique with Improved Two-Tone Resolution for Electronic Warfare Applications. MS Thesis. Wright State University, Dayton, OH, August 2004.
5. Tsui, James Bao-Yen. Digital Techniques for Wideband Receivers. IEEE Trans. on Microwave Theory and Techniques. 1997, Vol. 45, No. 12.
6. Kaya Z., Garrido M., Takala J. Memory-Based FFT Architecture With Optimized Number of Multiplexers and Memory Usage. IEEE Transactions on Circuits and Systems II: Express Briefs. 2023, 70(8), p. 3084-3088. DOI: <https://doi.org/10.1109/TCSII.2023.3245823>
7. Kaya Z., Garrido M. Low-Latency 64-Parallel 4096-Point Memory-Based FFT for 6G. IEEE Transactions on Circuits and Systems I: Regular Papers. 2023, 70(10), p. 4004-4014. DOI: <https://doi.org/10.1109/TCSI.2023.3298227>
8. Yang C., Wu J., Xiang S., Liang L., Geng L. A High-Throughput and Flexible Architecture Based on a Reconfigurable Mixed-Radix FFT with Twiddle Factor Compression and Conflict-Free Access. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2023, 31(10), p. 1472-1485. DOI: <https://doi.org/10.1109/TVLSI.2023.3298943>

9. García-Astudillo, L.A., Lindoso, A., Entrena, L., Martín, H., García-Valderas, M. Error sensitivity study of FFT architectures implemented in FPGA. *Microelectronics Reliability*. 2021, Vol. 126, 114298. DOI: <https://doi.org/10.1016/j.microrel.2021.114298>
10. Safarpour M., Silven O. LoFFT: Low-Voltage FFT Using Lightweight Fault Detection for Energy Efficiency. *IEEE Embedded Systems Letters*. 2023, 15(3), p. 125-128. DOI: <https://doi.org/10.1109/LES.2022.3212776>
11. Xie Y., Chen H., Zhuang Y., Xie Y. Fault Classification and Diagnosis Approach Using FFT-CNN for FPGA-Based CORDIC Processor. *Electronics (Switzerland)*. 2024, 13(1), art. no. 72. DOI: <https://doi.org/10.3390/electronics13010072>
12. Changela A., Zaveri M., Verma D. FPGA implementation of high-performance, resource-efficient Radix-16 CORDIC rotator based FFT algorithm. *Integration*. 2020, Vol. 73, p. 89-100. DOI: <https://doi.org/10.1016/j.vlsi.2020.03.008>
13. Dang T.-H., Tran V.-N., Nguyen L.-C. A parallel rotator for FFT/IFFT applied in multi-carrier wireless communication systems. *Digital Signal Processing: A Review Journal*. 2023, Vol. 141, art. no. 104190. DOI: <https://doi.org/10.1016/j.dsp.2023.104190>
14. He J., Bao Z., Li H., Li Q., Li Z., Liu P., Yang Z., Liu M. Implementation of an Adaptive Trapezoidal Shaping Method Based on FFT. *IEEE Transactions on Nuclear Science*. 2023, 70(9), p. 2234-2239. DOI: <https://doi.org/10.1109/TNS.2023.3307475>
15. Jiang, H., Yu, C.-Y., Chen, M., Liu, M. A Low Cost 4 bit 16 Giga-sample-per-second Analog-to-digital Converter for Radio Astronomy. *Publications of the Astronomical Society of the Pacific*. 2020, 132(1014), p. 1-7. DOI: <https://www.jstor.org/stable/26973300>
16. Song J., Li Y., Qiu J., Hong X., Guo H., Yang Z., Wu J. Low-Complexity FPGA Implementation of 106.24Gbps DP-QPSK Coherent Optical Receiver

- With Fractional Oversampling Rate Based on One FIR Filter for Resampling, Retiming and Equalizing. *Journal of Lightwave Technology*. 2023, 41(16), p. 5244-5251. DOI: <https://doi.org/10.1109/JLT.2023.3258072>
17. Yao Y., Zhang J., Liu Y., Ruan T., Li W., Lian H., Wang S., Zhang Y., Lin B., Jie Y., Liu H. Development of a multifunctional real-time data processing system for interferometers on EAST. *Journal of Instrumentation*. 2023, 11(3), art. no. C11013. DOI: <https://doi.org/10.1088/1748-0221/18/11/C11013>
18. Li C.J., Li X., Lou B., Jin C.T., Boland D., Leong P.H.W. Fixed-point FPGA Implementation of the FFT Accumulation Method for Real-time Cyclostationary Analysis. *ACM Transactions on Reconfigurable Technology and Systems*. 2023, 16(3), art. no. 41. DOI: <https://doi.org/10.1145/3567429>
19. Malathi L., Bharathi A., Jayanthi A.N. FPGA design of FFT based intelligent accelerator with optimized Wallace tree multiplier for image super resolution and quality enhancement. *Biomedical Signal Processing and Control*. 2024, Vol. 88, Part B, 105599. DOI: <https://doi.org/10.1016/j.bspc.2023.105599>
20. Park S., Park D. Low-Power FPGA Realization of Lightweight Active Noise Cancellation with CNN Noise Classification. *Electronics (Switzerland)*. 2023, 12(11), art. no. 2511. DOI: <https://doi.org/10.3390/electronics12112511>
21. Baas, Bevan M. An Approach to Low-Power, High-Performance, Fast Fourier Transform Processor Design. PhD. Dissertation. Stanford University, February 1999.
22. Lyons, R.G. (2022). *Understanding Digital Signal Processing* (3rd ed.). Pearson. ISBN 9780137582228.
23. Oppenheim, A. V., Schaffer, R. W., Buck, J. R. *Discrete-Time Signal Processing*. Second Edition. Upper Saddle River, NJ: Prentice Hall, Inc., 1999.