

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

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

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Бабкову Валерію Олексійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Цифровий фільтр на FPGA з використанням VHDL

затверджена наказом по університету від “ 26 ” травня 2025 р. № 426 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи _____

FPGA

VHDL

MATLAB

цифровий фільтр

4. Перелік питань, що потрібно опрацювати у роботі _____

Теоретичні основи розробки цифрових фільтрів на FPGA

Розробка та реалізація цифрового фільтра на FPGA

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 11 слайдів

6. Консультанти розділів роботи (заповнюється 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	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих фільтрів	31.05.2025–03.06.2025	
3	Вибір структур	04.06.2025–06.06.2025	
4	Вибір програмних засобів	07.06.2025–08.06.2025	
5	Програмна реалізація	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач


(підпис)

Керівник роботи

(підпис)

ас. Дмитро ДЯЧЕНКО

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 50 с., 8 рис., 2 дод., 10 джерел.

FIR-ФІЛЬТР, ЦИФРОВА ОБРОБКА СИГНАЛІВ, FPGA, VHDL, ЗСУВНИЙ РЕГІСТР, МНОЖНИК, АКУМУЛЯТОР, ФІКСОВАНА ТОЧНІСТЬ, MATLAB, ТЕСТБЕНЧ, СИНТЕЗ, МОДЕЛЮВАННЯ, АПАРАТНА АРХІТЕКТУРА, КОЕФІЦІЄНТИ ФІЛЬТРА, ACTIVE-HDL, HDL-ДИЗАЙН, ЦИФРОВИЙ ФІЛЬТР, МІКРОЕЛЕКТРОНІКА, ПЛІС, ЦИФРОВА СХЕМА.

Метою кваліфікаційної роботи є розробка, моделювання та реалізація цифрового фільтра із заданими характеристиками на FPGA за допомогою мови опису апаратури VHDL.

У ході виконання кваліфікаційної роботи розглянуто процес проєктування та апаратної реалізації цифрового фільтра з фіксованими коефіцієнтами (FIR-фільтра) на програмованій логічній інтегральній схемі (ПЛІС) із використанням мови опису апаратури VHDL. Основною метою дослідження є створення ефективної архітектури фільтра, здатної працювати в реальному часі при обмежених апаратних ресурсах. У роботі здійснено математичне обґрунтування структури фільтра, розраховано коефіцієнти в середовищі MATLAB, виконано перетворення значень у формат фіксованої точності для VHDL, реалізовано повний цикл розробки: від кодування апаратної архітектури до моделювання у програмному середовищі Active-HDL.

Архітектура фільтра реалізована у вигляді зсувного регістра, множників та акумулятора, що формує результат згортки, масштабований до 16-бітного виходу.

ABSTRACT

Bachelor's thesis: 50 pages, 8 figures, 2 appendices, 10 sources.

FIR FILTER, DIGITAL SIGNAL PROCESSING, FPGA, VHDL, SHIFT REGISTER, MULTIPLIER, ACCUMULATOR, FIXED-POINT ARITHMETIC, MATLAB, TESTBENCH, SYNTHESIS, SIMULATION, HARDWARE ARCHITECTURE, FILTER COEFFICIENTS, ACTIVE-HDL, HDL DESIGN, DIGITAL FILTER, MICROELECTRONICS, PLD, DIGITAL CIRCUIT.

The major goal of this thesis is the development, simulation, and implementation of a digital filter with specified characteristics on an FPGA using the VHDL hardware description language.

In order to the design and hardware implementation of a digital finite impulse response (FIR) filter with fixed coefficients on a field-programmable gate array (FPGA) was examined using VHDL. The primary goal of the study is to create an efficient filter architecture capable of real-time operation under constrained hardware resources. The work includes a mathematical justification of the filter structure, computation of coefficients in the MATLAB environment, conversion of values into a fixed-point format suitable for VHDL, and a complete development cycle from hardware architecture coding to simulation in the Active-HDL environment.

The filter architecture is implemented as a shift register, a set of multipliers, and an accumulator that computes the convolution result, scaled to a 16-bit output format.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	7
ВСТУП	8
1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ЦИФРОВИХ ФІЛЬТРІВ НА FPGA.....	10
1.1 Сигнальна обробка та місце цифрової фільтрації в ній	10
1.2 Класифікація та характеристики цифрових фільтрів	11
1.3 Алгоритмічні основи цифрової фільтрації	14
1.4 Особливості реалізації цифрових фільтрів на ПЛІС	16
1.5 Мова опису апаратури VHDL у контексті цифрових фільтрів	19
1.6 Огляд сучасних інструментів для проектування фільтрів на VHDL.....	21
2 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ЦИФРОВОГО ФІЛЬТРА НА FPGA.....	25
2.1 Постановка задачі цифрової фільтрації	25
2.2 Вибір структури фільтра	26
2.3 Розрахунок коефіцієнтів фільтра.....	28
2.4 Проектування архітектури фільтра	30
2.5 Реалізація фільтра	32
ВИСНОВКИ.....	38
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	39
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	40
ДОДАТОК Б Програмний код	47
Б.1 Лістинг коду MATLAB	47
Б.2 Лістинг коду VHDL	48

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AHDL – Active-HDL

AP – акумулятор результатів

AC – апаратна схема

АСЦ – апаратна схема цифрова

DSP – цифрова обробка сигналів

FIR – фільтр з кінцевою імпульсною характеристикою

FPGA – програмована логічна інтегральна схема

HDL – мова опису апаратури

PT – фіксована точність

CP – зсувний регістр

TB – тестбенч (тестове середовище)

VHDL – VHSIC Hardware Description Language

ВСТУП

Сучасний етап розвитку цифрових технологій характеризується стрімким зростанням вимог до швидкодії, надійності та гнучкості обчислювальних систем, що функціонують у режимі реального часу. Одним із ключових напрямів в електроніці та обробці сигналів є цифрова фільтрація, яка дозволяє ефективно виділяти необхідні компоненти сигналу, пригнічувати шум, зменшувати спотворення та підвищувати якість передавання інформації. Особливої актуальності ця задача набуває у сферах телекомунікацій, біомедицинської діагностики, обробки зображень, а також у системах керування вбудованого типу.

З огляду на потребу в обробці великих обсягів даних із мінімальними часовими затримками, зростає інтерес до апаратної реалізації цифрових фільтрів на програмованих логічних інтегральних схемах (ПЛІС). ПЛІС, або FPGA (Field-Programmable Gate Array), забезпечують можливість високошвидкісного паралельного виконання операцій із мінімальними енергетичними затратами, що вигідно вирізняє їх серед традиційних мікропроцесорних рішень. Їхня архітектура дозволяє реалізовувати як прості фільтри з обмеженим числом коефіцієнтів, так і складні адаптивні структури, які динамічно змінюють свої параметри залежно від вхідного сигналу.

Одним із основних інструментів проектування цифрових пристроїв на FPGA є мова опису апаратури VHDL (VHSIC Hardware Description Language). Саме вона дозволяє описати логіку обробки сигналів, а також задати архітектурні особливості, що враховують паралелізм, глибину конвеєризації, використання апаратних ресурсів та часові характеристики. VHDL дає змогу реалізовувати пристрої будь-якого рівня складності – від простих арифметичних модулів до повноцінних цифрових систем обробки сигналів. Її використання дозволяє не лише задати функціональність, але й проконтролювати часові затримки, ресурсну ефективність та сумісність із

конкретними мікросхемами.

Вибір саме цифрового фільтра як об'єкта дослідження обумовлений його універсальністю та широким спектром практичного застосування. У процесі роботи передбачається дослідити різні типи цифрових фільтрів, розглянути їхні алгоритмічні основи, обґрунтувати доцільність вибору певної структури, реалізувати її мовою VHDL, провести моделювання та верифікацію, а також виконати синтез для FPGA-платформи з подальшим тестуванням на практичному стенді. Особливу увагу буде приділено аналізу ефективності реалізованої апаратної моделі у порівнянні з теоретичними характеристиками.

Таким чином, мета цієї кваліфікаційної роботи полягає у створенні працездатного цифрового фільтра із заданими параметрами, який реалізується на ПЛІС з використанням мови опису апаратури VHDL, що дозволяє забезпечити високопродуктивну та надійну обробку сигналів у реальному часі.

Метою кваліфікаційної роботи є розробка, реалізація та експериментальне дослідження програмних засобів обробки зображень на основі штучних нейронних мереж у середовищі Google Colab для підвищення точності автоматичного розпізнавання візуальних об'єктів.

Завдання:

- провести аналіз цифрових фільтрів та їх реалізації на FPGA;
- визначити оптимальну структуру фільтра;
- розробити математичну модель обраного фільтра;
- реалізувати архітектуру фільтра на VHDL та в MATLAB;
- провести моделювання у середовищі Active HDL.

1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ЦИФРОВИХ ФІЛЬТРІВ НА FPGA

1.1 Сигнальна обробка та місце цифрової фільтрації в ній

Обробка сигналів є фундаментальною складовою сучасної електроніки, телекомунікацій, систем автоматизованого управління, мультимедійних застосунків, а також широкого кола інженерних рішень, що працюють із часовими або просторовими рядами даних. У її межах цифрова обробка сигналів (Digital Signal Processing, DSP) [1,2] посідає особливе місце завдяки здатності до точного математичного моделювання, гнучкої зміни параметрів в реальному часі та інтеграції зі складними обчислювальними платформами. На відміну від аналогових методів, цифрова обробка дозволяє формалізувати й програмно реалізувати складні фільтраційні процедури, які унеможливлено або надто дорого реалізувати апаратно в аналоговому середовищі.

Цифрова фільтрація в цьому контексті виконує ключову роль, забезпечуючи ізоляцію корисних компонентів сигналу, видалення перешкод, компенсацію викривлень, згладжування або виділення характерних особливостей даних. Вона базується на чисельних алгоритмах, які виконують операції над вибірками сигналу, обчислюючи нові значення відповідно до заданих функцій перетворення. Основною перевагою таких фільтрів є можливість реалізації широкого класу функцій передавання, що описуються математичними моделями, з гарантією стабільності та відтворюваності результату. У зв'язку з цим цифрові фільтри набули широкого поширення у пристроях зв'язку, системах цифрового телебачення, радіоелектронній розвідці, аудіообробці, спектральному аналізі, біомедичних пристроях, цифрових приладах контролю та діагностики.

Істотною перевагою цифрової обробки сигналів є її модульність та масштабованість. Це дозволяє створювати як елементарні алгоритми для енергообмежених пристроїв, так і високопродуктивні паралельні

обчислювальні структури, що реалізуються на апаратному рівні. При цьому можливість швидкої перебудови конфігурації та адаптації до нових умов середовища забезпечується за допомогою мов опису апаратури, таких як VHDL, та середовищ проєктування, зокрема на основі FPGA [3].

Застосування цифрових фільтрів на ПЛІС створює нові перспективи для високошвидкісної обробки вхідних потоків даних. FPGA-платформи дозволяють реалізовувати паралельні обчислення, конвеєрні структури та апаратну логіку, що працює незалежно від основного програмного забезпечення, що істотно знижує затримку обробки. У порівнянні з мікропроцесорними рішеннями, ПЛІС забезпечують вищу продуктивність при значно меншому енергоспоживанні, що критично для застосувань у вбудованих і мобільних системах.

Таким чином, цифрова фільтрація є невід'ємною складовою сучасних інформаційно-технологічних систем. Її апаратна реалізація на базі ПЛІС у поєднанні з мовою опису VHDL [4] відкриває широкі можливості для створення ефективних, масштабованих і надійних обчислювальних пристроїв, здатних до обробки сигналів у реальному часі з високим ступенем точності та адаптивності.

1.2 Класифікація та характеристики цифрових фільтрів

Цифрові фільтри є невід'ємними елементами у структурі обробки дискретних сигналів, що функціонують у часовій або частотній області. Їх математична природа дозволяє з високою точністю моделювати бажані властивості передавання сигналів, що стало основою для створення великої кількості структур та алгоритмів. Загальноприйнятою є класифікація цифрових фільтрів за типом їх імпульсної характеристики, яка поділяє всі фільтри на дві фундаментальні групи – фільтри скінченної імпульсної характеристики (FIR) та фільтри нескінченної імпульсної характеристики (IIR) [5].

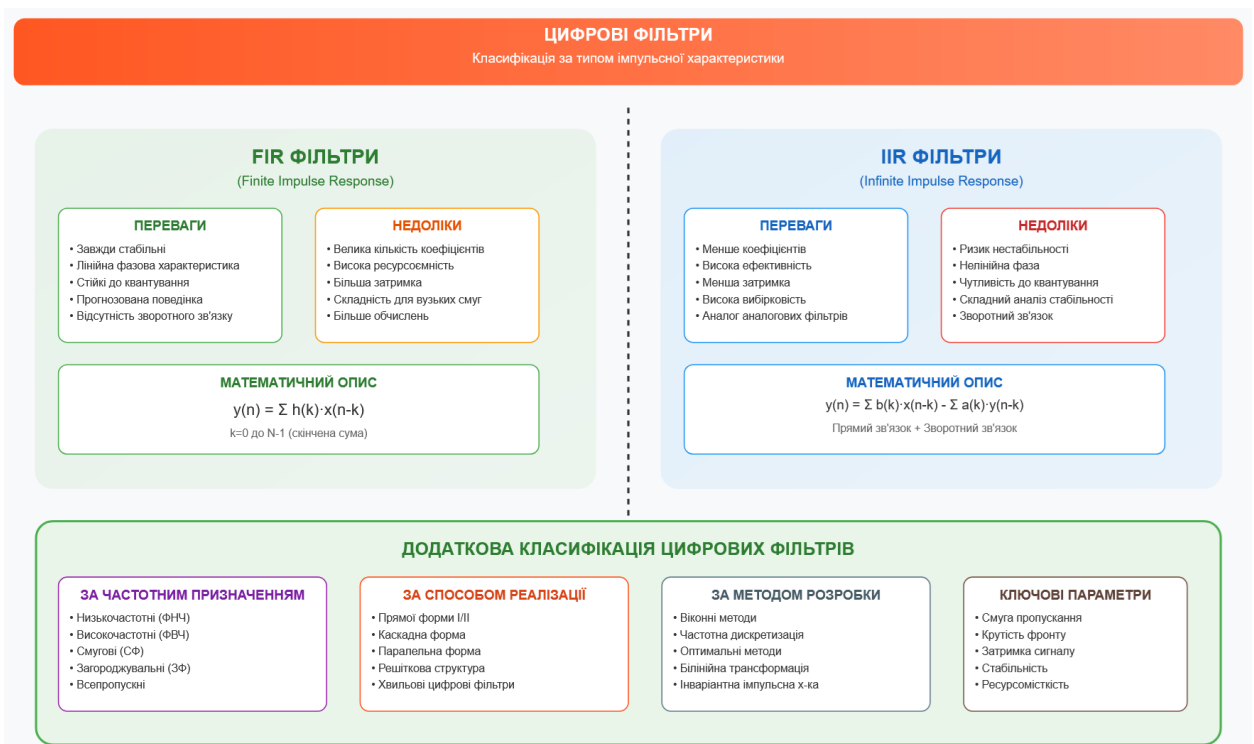


Рисунок 1.1 – Цифрові фільтри

На рисунку 1.1 представлені цифрові фільтри є невід'ємними елементами у структурі обробки дискретних сигналів, що функціонують у часовій або частотній області. Їх математична природа дозволяє з високою точністю моделювати бажані властивості передавання сигналів, що стало основою для створення великої кількості структур та алгоритмів. Загальноприйнятою є класифікація цифрових фільтрів за типом їх імпульсної характеристики, яка поділяє всі фільтри на дві фундаментальні групи – фільтри скінченної імпульсної характеристики (FIR) та фільтри нескінченної імпульсної характеристики (IIR).

FIR-фільтри характеризуються обмеженою тривалістю імпульсної характеристики, тобто їх вихідний сигнал залежить лише від кінцевого числа попередніх значень вхідного сигналу. Такі фільтри завжди є стабільними, навіть у випадку реалізації з обмеженою точністю.

FIR-фільтри характеризуються обмеженою тривалістю імпульсної характеристики, тобто їх вихідний сигнал залежить лише від кінцевого числа попередніх значень вхідного сигналу. Такі фільтри завжди є стабільними,

навіть у випадку реалізації з обмеженою точністю. Їхньою важливою перевагою є можливість забезпечення точної лінійності фазової характеристики, що є критично важливим для обробки аудіосигналів, зображень та інших застосувань, де спотворення фази може призвести до небажаних артефактів. Однак їх головним недоліком вважається потреба у великій кількості множень і додавань, особливо для фільтрів з вузькою смугою пропускання, що підвищує ресурсоємність апаратної реалізації.

IIR-фільтри, натомість, мають нескінченну імпульсну характеристику, оскільки їх вихід залежить не лише від вхідного сигналу, але й від попередніх значень самого вихідного сигналу. Це забезпечує вищу ефективність у досягненні бажаних частотних характеристик при меншій кількості коефіцієнтів, що робить такі фільтри привабливими для застосувань, де обмежені ресурси і важлива висока вибірковість. Водночас, через рекурсивну природу обчислень, реалізація IIR-фільтрів супроводжується ризиком втрати стабільності, особливо в умовах обмеженої розрядності, а також складністю досягнення лінійної фази.

Крім фундаментального поділу на FIR та IIR, цифрові фільтри також класифікуються за частотним призначенням: низькочастотні, високочастотні, смугові та загороджувальні; за способом реалізації – прямої форми, каскадної або решіткової; за методом розробки – віконні методи, методи оптимізації частотної характеристики, трансформаційні підходи тощо. Вибір конкретної структури залежить від технічного завдання, допустимого спотворення амплітуди та фази, обмежень на затримку сигналу, продуктивності системи та доступних апаратних ресурсів.

У процесі цифрової фільтрації важливу роль відіграють характеристики фільтра, зокрема смуга пропускання, крутість фронту зрізу, величина затримки, стабільність, ресурсомісткість реалізації, а також чутливість до квантування коефіцієнтів. Такі параметри визначають як якість обробки сигналу, так і ефективність реалізації фільтра в конкретному середовищі, особливо при апаратному втіленні на ПЛІС.

У сукупності, розуміння переваг і обмежень різних типів цифрових фільтрів, а також характеристик, що впливають на їх поведінку, є необхідним етапом при розробці систем цифрової обробки сигналів. Це дозволяє приймати обґрунтовані рішення щодо вибору структури, алгоритму та методів реалізації з урахуванням цільових параметрів і архітектурних обмежень програмованої логіки.

1.3 Алгоритмічні основи цифрової фільтрації

Процес цифрової фільтрації ґрунтується на математичній обробці дискретного сигналу, що полягає в застосуванні певної фільтраційної функції, заданої у вигляді різницевого рівняння або згортки. В основі цього підходу лежить уявлення сигналу як послідовності числових вибірок, над якими виконуються лінійні або нелінійні операції, спрямовані на пригнічення небажаних частотних компонентів або виділення корисного спектру. Алгоритм цифрової фільтрації визначається типом фільтра, структурою передавальної функції та методом обчислення його вихідного значення.

Для фільтрів скінченної імпульсної характеристики (FIR) основним математичним виразом є операція дискретної згортки між вхідним сигналом та імпульсною характеристикою. Це рівняння має вигляд:

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n - k], \quad (2.1)$$

де $y[n]$ – вихідний сигнал у момент часу n , $x[n-k]$ – вхідні значення з часовим зсувом, $h[k]$ – коефіцієнти фільтра, що визначають його частотну характеристику, а N – порядок фільтра. Цей вираз передбачає виконання N множень і $(N-1)$ додавань для кожного нового вибіркового значення, що зумовлює значне навантаження при реалізації на загальнопроцесорних

архітектурах. Водночас, така структура є детермінованою, лінійною, стійкою і зручною для апаратного втілення завдяки відсутності зворотного зв'язку.

Реалізація фільтрів нескінченної імпульсної характеристики (ІІР) ґрунтується на використанні рекурсивного рівняння, у якому вихідний сигнал залежить як від вхідного, так і від попередніх значень вихідного сигналу:

$$y[n] = \sum_{i=0}^M b_i \cdot x[n - i] - \sum_{j=1}^N a_j \cdot y[n - j], \quad (2.2)$$

де коефіцієнти b_i відповідають нульовим значенням, а a_j – полюсам фільтра. Така форма дозволяє досягти бажаної частотної характеристики при меншій кількості коефіцієнтів, що сприяє зменшенню обчислювального навантаження, однак ускладнює забезпечення стабільності системи, особливо при реалізації з обмеженою розрядністю.

Розрахунок коефіцієнтів фільтра є невід'ємною складовою етапу проектування. Для FIR-фільтрів часто використовуються віконні методи, зокрема застосування прямого перетворення ідеальної імпульсної характеристики із подальшим згладжуванням за допомогою функцій згортки (вікон Геммінга, Блекмана, Кайзера). Такий підхід дозволяє контролювати ширину перехідної смуги, рівень бічних пелюсток та амплітудні викривлення. Альтернативою є оптимізаційні методи, які мінімізують середньоквадратичну похибку або максимальне відхилення в заданій частотній області.

У випадку ІІР-фільтрів найпоширенішими є підходи на основі аналогових прототипів, зокрема методи Бесселя, Баттерворта, Чебишева, які трансформуються в дискретну область за допомогою алгоритму білатерального Z-перетворення або методу імпульсної інваріантності. Ці методи забезпечують контрольовану поведінку системи у частотній області,

однак часто вимагають складної арифметики для реалізації в цифровій логіці.

Особливу увагу при апаратній реалізації цифрових фільтрів слід приділяти фіксованій точності, квантуванню коефіцієнтів, оптимізації розрядності та захисту від переповнення. Саме алгоритмічний рівень, що поєднує вибір формули, порядок фільтра та метод розрахунку коефіцієнтів, визначає успішність подальшого перенесення моделі на апаратний рівень із використанням мов опису апаратури, зокрема VHDL.

1.4 Особливості реалізації цифрових фільтрів на ПЛІС

Реалізація цифрових фільтрів на програмованих логічних інтегральних схемах (ПЛІС), або FPGA, має низку суттєвих відмінностей від програмної реалізації на процесорах загального призначення. Ці відмінності зумовлені архітектурною природою ПЛІС, яка базується на гнучко конфігурованих логічних блоках, що дозволяють створювати апаратно орієнтовані обчислювальні структури з високим ступенем паралелізму. Такий підхід відкриває можливість обробки сигналів у реальному часі без необхідності використання центрального процесора, що особливо важливо для систем із жорсткими вимогами до затримки, енергоспоживання й швидкодії.

Основною перевагою використання ПЛІС у задачах цифрової фільтрації є можливість реалізації паралельних обчислень. На відміну від послідовного виконання інструкцій у традиційних мікропроцесорах, логічні блоки FPGA можуть працювати одночасно над різними частинами фільтраційного алгоритму. Це дає змогу досягати високої пропускної здатності при збереженні детермінованості системи. Наприклад, при реалізації FIR-фільтра кожне множення та додавання може бути реалізоване як окремий логічний елемент, що виконується в одному або декількох тактах, залежно від глибини конвеєризації. Це дозволяє масштабувати фільтр за розрядністю і кількістю коефіцієнтів практично без втрати продуктивності.

РЕАЛІЗАЦІЯ ЦИФРОВИХ ФІЛЬТРІВ НА FPGA

s ПРОЦЕСОРИ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

ПРОЦЕСОР (CPU)
<ul style="list-style-type: none"> • Висока швидкість виконання інструкцій • Висока пропускна здатність • Ефективна тримка даних • Висока якість ОС та драйверів • Енергоспоживання • Висока якість real-time обробки • Висока якість архітектури процесора • Висока якість використання ресурсів


FPGA (ПЛІС)
<ul style="list-style-type: none"> • Паралельні обчислення • Апаратно орієнтовані структури • Висока пропускна здатність • Детермінованість системи • Мінімальні затримки (latency) • Прямий доступ до апаратури • Низьке енергоспоживання • Гарантована real-time обробка • Гнучкість архітектури • Виділені ресурси для кожної задачі

СПЕЦІАЛІЗОВАНІ РЕСУРСИ FPGA

DSP SLICES (БЛОКИ DSP)
<ul style="list-style-type: none"> • MAC блоки (Multiply-Accumulate) • 25x18 біт множення (Xilinx) • 48-бітні суматори/акумулятори • Pipeline глибина 3-4 такти

БЛОКИ ПАМ'ЯТІ
<ul style="list-style-type: none"> • BRAM: 36 Кбіт блоки (Xilinx) • Розподілена пам'ять: LUT як RAM • Зсувні регістри (SRL) • Буферизація коефіцієнтів та даних

КТУРИ РЕАЛІЗАЦІЇ FIR-ФІЛЬТРІВ

ПОСЛІДОВНА АРХІТЕКТУРА
<p>Принципова схема:</p>  <p>Ресурси:</p> <ul style="list-style-type: none"> • SP slice • N на вибірку (N - порядок фільтра) • N ресурсів • семплована = fclk/N

ПОВНІСТЮ ПАРАЛЕЛЬНА АРХІТЕКТУРА
<p>Структурна схема:</p>  <p>Характеристики:</p> <ul style="list-style-type: none"> • N DSP slices (N множиликів) • Один такт на вибірку

КОНВЕБЕРНА АРХІТЕКТУРА
<p>Pipeline стадії:</p>  <p>Характеристики:</p> <ul style="list-style-type: none"> • Частота семплована = fclk • Затримка = кількість стадій • Балансування ресурсів/швидкості • Вища тактова частота

ІНІ ПАРАМЕТРИ ТА ОБМЕЖЕННЯ

ЧАСОВІ ХАРАКТЕРИСТИКИ
<ul style="list-style-type: none"> • $t_{\text{nsd}} (T_{\text{clk}}) = 1/f_{\text{clk}}$ • $t_{\text{nsu}} = 0.5-2 \text{ ns}$ • $t_{\text{ns}} = 0.1-0.5 \text{ ns}$ • $t_{\text{nsd}} \text{ delay} (T_{\text{pd}})$ • $t_{\text{nsu}} < 0.5 \text{ ns}$ • $t_{\text{nsd}} > 0$

АРИФМЕТИЧНА ТОЧНІСТЬ
<ul style="list-style-type: none"> • Фіксована точність (Qm.n формат) • Розрядність входу: 8-16 біт • Розрядність коефіцієнтів: 12-18 біт • Внутрішня розрядність: 48 біт • Двійковий додатковий код • Контроль переповнення

ВИКОРИСТАННЯ РЕСУРСІВ
<ul style="list-style-type: none"> • DSP48E slices: 1 на множилик • BRAM: коефіцієнти та буфери • LUT: керує логікою • FF: pipeline регістри • I/O: AXI-Stream інтерфейси • Clock: глобальні мережі

ІНТЕРФЕЙСИ
<ul style="list-style-type: none"> • AXI4-Stream • Avalon-ST • FIFO інтерфейси • Handshaking протоколи • Clock domain crossing • Backpressure підтримка

Рисунок 1.2 – Реалізація цифрових фільтрів

Реалізація цифрових фільтрів на програмованих логічних інтегральних схемах або FPGA (рисунок 1.2), має низку суттєвих відмінностей від програмної реалізації на процесорах загального призначення. Ці відмінності зумовлені архітектурною природою ПЛІС, яка базується на гнучко конфігурованих логічних блоках, що дозволяють створювати апаратно орієнтовані обчислювальні структури з високим ступенем паралелізму. Такий підхід відкриває можливість обробки сигналів у реальному часі без необхідності використання центрального процесора, що особливо важливо для систем із жорсткими вимогами до затримки, енергоспоживання й швидкодії.

Основною перевагою використання ПЛІС у задачах цифрової фільтрації є можливість реалізації паралельних обчислень. На відміну від послідовного виконання інструкцій у традиційних мікропроцесорах, логічні блоки FPGA можуть працювати одночасно над різними частинами фільтраційного алгоритму. Це дає змогу досягати високої пропускної

здатності при збереженні детермінованості системи

Архітектура сучасних FPGA містить спеціалізовані ресурси для реалізації арифметичних операцій, зокрема блоки множення-накопичення (MAC), апаратні суматори, блоки цифрової сигнальної обробки (DSP slices) та внутрішню оперативну пам'ять (BRAM), що можна використовувати для збереження коефіцієнтів або буферизації даних [6]. Вибір відповідної архітектурної топології фільтра залежить від бажаного балансу між швидкістю, ресурсоспоживанням та складністю проектування. Найпоширенішими підходами є послідовна реалізація з використанням зсувних регістрів, повністю паралельна реалізація з одночасною обробкою всіх вибірок, а також частково паралельні архітектури з використанням конвеєрних структур.

Реалізація цифрових фільтрів на ПЛІС вимагає ретельного врахування часових характеристик, зокрема затримок на поширення сигналу, затримок синхронізації та часу встановлення даних. Ці аспекти визначають максимальну тактову частоту, з якою система може функціонувати без порушення стабільності. Оптимізація тактової частоти досягається через використання pipeline-архітектур, в яких обчислення розбиваються на кілька стадій із проміжним збереженням результатів. Це дозволяє збільшити обчислювальну глибину без втрати продуктивності.

Особливу увагу слід приділяти обробці чисел з фіксованою точністю, оскільки реалізація операцій з плаваючою комою на ПЛІС є ресурсоемною і потребує значних логічних площ. Тому в більшості випадків застосовують фіксовану арифметику з попереднім масштабуванням коефіцієнтів, що дозволяє реалізувати обчислення на основі цілих чисел із заданою розрядністю. Такий підхід потребує додаткового аналізу щодо переповнення, втрати точності, а також дотримання знаковості чисел, що реалізується через відповідне кодування, наприклад, у форматі двійкового додаткового коду.

У процесі апаратної реалізації необхідно також враховувати питання синхронізації та керування потоками даних. Як правило, цифрові фільтри

реалізуються як автономні модулі, які взаємодіють із периферією через стандартизовані інтерфейси (наприклад, AXI-Stream, Avalon), що дозволяє їх вбудовувати в більші проєкти систем на кристалі (SoC). Модулі керування, генерації тактових сигналів та контролю станів також мають бути реалізовані засобами HDL, що забезпечує повну інтеграцію з іншими компонентами цифрової системи.

Таким чином, використання ПЛІС для реалізації цифрових фільтрів відкриває можливості створення високошвидкісних, масштабованих і адаптивних систем обробки сигналів, які суттєво перевершують традиційні процесорні рішення як за продуктивністю, так і за гнучкістю. Ефективне використання апаратних ресурсів, правильний вибір архітектури, ретельна арифметична оптимізація та дотримання часових обмежень є ключовими факторами успішного проєктування фільтрів на базі FPGA.

1.5 Мова опису апаратури VHDL у контексті цифрових фільтрів

Мова опису апаратури VHDL посідає ключове місце в сучасному проєктуванні цифрових пристроїв, зокрема у контексті реалізації систем цифрової обробки сигналів на ПЛІС. Її використання дозволяє формалізовано описувати апаратну логіку на різних рівнях абстракції – від поведінкового до рівня регістрів і фізичного розміщення. Особливо важливим є те, що VHDL не є мовою програмування у класичному сенсі, а слугує засобом моделювання, синтезу та перевірки цифрових систем, які реалізуються як апаратні об'єкти в інтегральній схемі.

У реалізації цифрових фільтрів мова VHDL виступає як універсальний інструмент для опису логічної структури фільтра, обробки вхідних вибірок, реалізації арифметичних операцій множення та додавання, а також забезпечення синхронного керування даними в межах заданої тактової частоти. Її використання забезпечує незалежність від конкретної FPGA-платформи, дозволяючи описувати логіку у вигляді абстрактних конструкцій,

які потім адаптуються до конкретного синтезатора чи архітектури ПЛІС.

У розробці фільтрів найчастіше застосовується рівень опису RTL (Register Transfer Level), на якому формалізовано задаються регістри зберігання, шляхи передачі даних і логіка керування. Це дозволяє детально контролювати час проходження сигналу через модуль, реалізовувати pipeline-структури та забезпечувати узгодження із зовнішніми модулями. VHDL забезпечує можливість створення ієрархічних структур, де цифровий фільтр може бути оформлений як окремий компонент, що взаємодіє з іншими блоками через визначені порти.

Однією з особливостей мови VHDL є її строгий синтаксис і суворі типізація, що, з одного боку, ускладнює початкове навчання, але з іншого – підвищує надійність коду та зменшує ймовірність помилок при синтезі. У контексті цифрової фільтрації ключову роль відіграє підтримка типів з фіксованою точністю, які дозволяють реалізовувати обчислення у цілочисельній формі. Для цього широко використовуються пакети `numeric_std` або `fixed_pkg`, що забезпечують підтримку операцій над знаковими та беззнаковими типами, з урахуванням розрядності, масштабування та обмеження переповнень.

Арифметика у VHDL вимагає явного визначення ширини бітових полів для кожної змінної, що особливо важливо при реалізації фільтрів, де точність і переповнення мають критичне значення. Наприклад, при множенні двох 16-бітних значень необхідно передбачити розширений результат щонайменше до 32 біт, після чого виконати масштабування або усічення для збереження розрядності виходу. Для цього в VHDL передбачено механізми приведення типів та зрізу бітів, що дозволяє керувати точністю й адаптувати код до обмежень ПЛІС.

Створення цифрового фільтра у VHDL передбачає також розробку тестового середовища (`testbench`), що дає змогу верифікувати правильність роботи модуля до його синтезу. У `testbench`-модулі задається імпульсний, синусоїдальний або шумовий вхідний сигнал, а також записуються вихідні

вибірки для порівняння з еталонними результатами, отриманими, наприклад, у MATLAB. Такий підхід дає змогу перевірити функціональну правильність фільтра, визначити затримку, виявити проблеми переповнення або похибки квантування ще до етапу апаратної реалізації.

Загалом, мова VHDL є незамінним інструментом при створенні цифрових фільтрів на ПЛІС. Вона забезпечує детальний контроль над архітектурою пристрою, дозволяє реалізовувати ефективну обробку сигналів у реальному часі, забезпечує можливості моделювання, відлагодження та повторного використання модулів, що робить її ідеальною платформою для розробки високопродуктивних і надійних систем цифрової фільтрації.

1.6 Огляд сучасних інструментів для проектування фільтрів на VHDL

Сучасна практика проектування цифрових фільтрів на мові VHDL передбачає використання комплексного програмного інструментарію, який охоплює всі етапи життєвого циклу розробки цифрової системи – від математичного моделювання до синтезу та апаратної реалізації на кристалі ПЛІС. Різноманітність доступних програмних рішень, засобів автоматизації та бібліотек значною мірою визначає ефективність, точність і надійність кінцевої реалізації. Для створення цифрових фільтрів доцільним є інтегроване використання таких інструментів, як MATLAB, Vivado Design Suite, Quartus Prime, ModelSim та інших.

На етапі розробки математичної моделі фільтра провідним середовищем є MATLAB з модулем Signal Processing Toolbox [7,8]. Цей інструмент дає змогу проектувати фільтри різних типів (FIR, IIR), розраховувати їх частотні характеристики, візуалізувати імпульсну відповідь, фазовий зсув, затримку та інші метрики, важливі для аналізу та валідації. MATLAB також дозволяє автоматизовано згенерувати коефіцієнти фільтра з високою точністю, які в подальшому можуть бути адаптовані для реалізації з фіксованою точністю в VHDL. Крім того, існують плагіни та інструменти

(наприклад, HDL Coder), які дають змогу автоматично згенерувати VHDL-код на основі моделі MATLAB/Simulink.

МЕТОДИ СИНТЕЗУ FIR-ФІЛЬТРІВ

<p>МЕТОД ПРЯМОГО ОБРИЗАННЯ Frequency Domain Truncation</p> <p>Математичний опис: $H(\omega) \rightarrow \text{FFT} \rightarrow h(n) \rightarrow \text{truncate} \rightarrow h$ $h(n) = (1/2\pi) \int_{-\pi}^{\pi} H(e^{j\omega}) e^{jn\omega} d\omega$</p> <p>Характеристики: • Інформація реалізації • Ефект Гіббса на краях • Основа для віконних методів • Нескінченна ІХ → об'єднана</p>	<p>МЕТОД ВІКОН Windowing Method</p> <p>Типи вікон: • Rectangular: $w(n) = 1$ • Hamming: $w(n) = 0.54 - 0.46\cos(2\pi n/N)$ • Hanning: $w(n) = 0.5(1 - \cos(2\pi n/N))$ • Blackman: 3-term cosine series</p> <p>Порівняння параметрів: Rectangular: $\Delta\omega = 9\pi/N$, $A_s = 21\text{дБ}$ Hamming: $\Delta\omega = 3.3\pi/N$, $A_s = 53\text{дБ}$ Blackman: $\Delta\omega = 5.5\pi/N$, $A_s = 74\text{дБ}$</p>	<p>ОПТИМАЛЬНИЙ СИНТЕЗ Parks-McClellan / Chebyshev</p> <p>Алгоритм Ремеза: • Minimax-оптимізація • Рівномірне наближення • Максимальна селективність • Equiripple характеристика</p> <p>Функція помилки: $E(\omega) = W(\omega)H(\omega) - H_d(\omega)$ $\omega \in \text{bands}$</p>
<p>МЕТОД НАЙМЕНШИХ КВАДРАТІВ Least Squares Method</p> <p>Цільова функція: $J = W(\omega)H(\omega) - H_d(\omega) ^2$</p> <p>Переваги: • Гладка характеристика • Низькі пульсації • Стабільність синтезу • М'який фронт</p>	<p>ЛІНІЙНЕ ПРОГРАМУВАННЯ Linear Programming Method</p> <p>Формулювання задачі: Minimize: $c^T x$ Subject to: $Ax \leq b$ де x - коефіцієнти фільтра</p> <p>Особливості: • Універсальність • Гнучі обмеження • Висока складність • Рідко використовується</p>	<p>СПЕКТРАЛЬНА ФАКТОРИЗАЦІЯ Spectral Factorization Method</p> <p>Принцип: $S(\omega) = H(\omega)H^*(\omega)$ $S(z) = H(z)H^*(z^*)$</p> <p>Застосування: • Адаптивні фільтри • Стохастичні процеси • Задана PSD • Noise shaping</p>

ПОРІВНЯЛЬНА ТАБЛИЦЯ МЕТОДІВ

МЕТОД	СКЛАДНІСТЬ	СЕЛЕКТИВНІСТЬ	ПУЛЬСАЦІЇ	ФРОНТ ЗРІЗУ	СТАБІЛЬНІСТЬ	ГНУЧІСТЬ	ЗАСТОСУВАННЯ	FPGA
Віконний	Низька	Середня	Залепить від вікна	Широкий	Висока	Середня	Загальне	Відмінно
Parks-McClellan	Висока	Максимальна	Equiripple	Найефективніший	Висока	Середня	Критичні системи	Добре
Найм. квадрати	Середня	Середня	Низькі	Широкий	Дуже висока	Висока	Аудіо, зображення	Добре
Лінійне програм.	Дуже висока	Налаштується	Контрольовані	Налаштується	Середня	Максимальна	Спеціальні задачі	Складно
Спектр. факториз.	Висока	Спеціальна	Залепить від PSD	Спеціальний	Середня	Низька	Адаптація, PSD	Середньо
Пряме обрізання	Найменша	Низька	Ефект Гіббса	Широкий	Висока	Низька	Швидке прототипування	Відмінно

ПОСЛІДОВНІСТЬ ПРОЕКТУВАННЯ FIR-ФІЛЬТРА

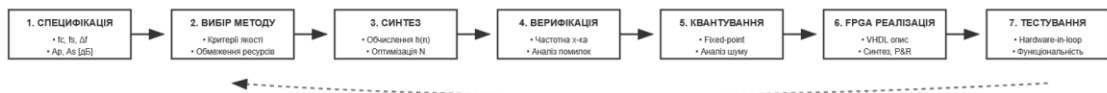
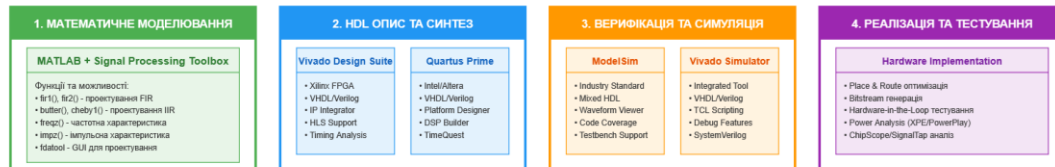


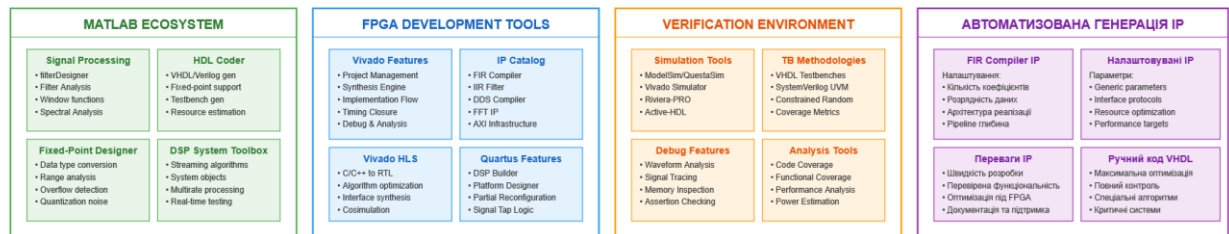
Рисунок 1.3 – Методи синтезу FIR-фільтрів

ІНСТРУМЕНТАРІЙ ДЛЯ ПРОЕКТУВАННЯ ЦИФРОВИХ ФІЛЬТРІВ НА VHDL

ЕТАПИ ЖИТТЄВОГО ЦИКЛУ РОЗРОБКИ



ДЕТАЛЬНЕ ПОРІВНЯННЯ ІНСТРУМЕНТІВ



ІНТЕГРОВАННИЙ РОБОЧИЙ ПРОЦЕС



Рисунок 1.4 – Огляд сучасних інструментів для проектування фільтрів на VHDL

На етапі опису та верифікації логіки фільтра використовується середовище розробки мовою VHDL, зокрема Vivado Design Suite (для FPGA Xilinx) [9] або Active HDL. У цих системах забезпечено підтримку проєктного керування, ієрархічної побудови, модулів автоматичного синтезу, аналізу затримок, генерації програмованих образів (bitstream) та їх завантаження на цільову платформу. Вони також містять зручні редактори коду з перевіркою синтаксису, інструменти генерації IP-ядра та бібліотеки стандартних цифрових компонентів, що значно прискорює процес проєктування.

На рисунку 1.3 представлені методи синтезу цифрових фільтрів, а на рисунку 1.4

Особливе значення має моделювання та верифікація розроблених VHDL-модулів, яке зазвичай виконується в середовищі Active HDL. Верифікація дозволяє перевірити логіку роботи цифрового фільтра, виявити помилки на етапі обчислень, затримки в синхронізації сигналів, некоректну обробку переповнення, а також відповідність отриманих вихідних даних очікуваним значенням. Для цього створюються тестові модулі (testbench), які подають на вхід моделювання заздалегідь визначені сигнали, фіксують результати та дозволяють здійснювати порівняння з еталоном, отриманим у MATLAB або іншому середовищі математичного моделювання.

Крім того, у сучасних FPGA-середовищах широко використовуються засоби автоматичної генерації фільтрів. Наприклад, у Vivado наявний FIR Compiler IP, який дозволяє проєктувати FIR-фільтри із заданими параметрами, автоматично виконуючи генерацію VHDL-коду, оптимізованого під конкретну архітектуру FPGA. Аналогічні інструменти пропонує й Quartus Prime через IP Catalog, що забезпечує створення цифрових фільтрів із можливістю налаштування розрядності, типу арифметики, кількості коефіцієнтів, глибини pipeline тощо. Хоча такі автоматизовані рішення скорочують час розробки, вони не завжди забезпечують максимальну оптимізацію, тому ручне написання коду

VHDL[10] залишається актуальним для критичних систем.

Інтеграція усіх зазначених інструментів у єдиний робочий процес дозволяє досягти високої точності проектування, гнучкості у виборі параметрів, можливості ранньої перевірки результатів і, зрештою, підвищення якості розробленого апаратного фільтра. У контексті даної кваліфікаційної роботи застосування таких інструментів є критично важливим для забезпечення достовірності моделі, її ефективності в реальному часі та відповідності технічним характеристикам цільової платформи.

2 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ЦИФРОВОГО ФІЛЬТРА НА FPGA

2.1 Постановка задачі цифрової фільтрації

У рамках кваліфікаційної роботи передбачено розробку цифрового фільтра, який реалізується на основі ПЛІС за допомогою мови опису апаратури VHDL. Завданням є створення працездатного фільтра, здатного ефективно обробляти вхідний сигнал у реальному часі, із заданими характеристиками частотної селективності, затримки, точності обчислень та апаратної ефективності. Згідно з технічними вимогами, фільтр має забезпечувати пригнічення високочастотних шумів у спектрі цифрового сигналу при збереженні корисних частот у межах допустимої смуги пропускання.

Параметри проєктованого фільтра визначаються виходячи з моделюваного прикладного сценарію, у якому сигнал підлягає цифровій обробці після оцифрування. Частота дискретизації сигналу встановлюється на рівні 48 кГц, що є типовим значенням для задач аудіообробки, медичної діагностики та сенсорного моніторингу. Виходячи з цього, завдання полягає у розробці фільтра нижніх частот із частотою зрізу на рівні 8 кГц. Такий фільтр повинен відсікати компоненти сигналу, що перевищують даний поріг, водночас мінімізуючи фазові викривлення у смузі пропускання.

Додатковими умовами є обмеження щодо розрядності сигналу та коефіцієнтів. Зокрема, вхідний сигнал представлено у вигляді 16-бітових цілих чисел зі знаковим поданням, що відповідає формату типових аналогово-цифрових перетворювачів. Для забезпечення ефективної апаратної реалізації та уникнення зайвого використання логічних ресурсів, передбачається застосування фіксованої точності з масштабуванням коефіцієнтів фільтра до відповідного діапазону.

У процесі реалізації необхідно забезпечити синхронну обробку

кожного вибіркового значення сигналу протягом одного або декількох тактів тактового генератора, не порушуючи безперервності даного потоку. Таким чином, цифровий фільтр має функціонувати у режимі потокової обробки, без накопичення сигналу в пам'яті, з мінімальною затримкою та гарантією детермінованого виходу.

Загальна постановка задачі охоплює весь цикл інженерного проектування – від вибору архітектури фільтра та розрахунку його параметрів до опису логіки мовою VHDL, моделювання її роботи у програмному середовищі, синтезу апаратної реалізації для конкретної FPGA-платформи та перевірки працездатності розробки в тестовому стенді. При цьому особлива увага приділяється узгодженню між теоретично обчисленими характеристиками та реальними результатами, отриманими у ході моделювання та апаратного тестування.

Таким чином, сформульована задача охоплює розробку, реалізацію та оцінку ефективності цифрового фільтра, побудованого на основі FIR-структури, з урахуванням обмежень обраної ПЛІС-платформи, що забезпечує практичну реалізацію систем обробки сигналів у реальному часі.

2.2 Вибір структури фільтра

Вибір архітектурної структури цифрового фільтра є одним із ключових етапів проектування, що визначає як функціональні, так і апаратні характеристики майбутньої реалізації. У межах цієї роботи було прийнято рішення про використання фільтра скінченної імпульсної характеристики (FIR), з огляду на його властиву стабільність, передбачувану поведінку та здатність до реалізації фази, близької до лінійної. На відміну від рекурсивних IIR-фільтрів, FIR-структури не використовують зворотних зв'язків, а отже, позбавлені ризику нестабільної поведінки в умовах квантування або обмеженої розрядності, що робить їх особливо придатними для апаратного втілення у системах із фіксованою точністю.

З математичної точки зору, FIR-фільтр описується як згорткова сума, що пов'язує кожне нове значення вихідного сигналу зі скінченим числом попередніх вхідних значень, зважених на відповідні коефіцієнти. Таке представлення реалізується у вигляді різницевого рівняння:

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n - k], \quad (2.3)$$

де $y[n]$ – значення вихідного сигналу на кроці n , $x[n-k]$ – послідовність попередніх вхідних значень, а $h[k]$ – імпульсна характеристика або фільтраційні коефіцієнти. Кількість коефіцієнтів NN визначає порядок фільтра і прямо впливає на вибірковість, амплітудно-частотну характеристику, а також ресурсну насиченість апаратної реалізації.

У практичному сенсі, для забезпечення належної смуги пропускання та пригнічення високочастотних компонентів було обрано фільтр 21-го порядку (тобто 21 коефіцієнт). Такий порядок забезпечує достатню крутість фронту зрізу на межі 8 кГц при частоті дискретизації 48 кГц, а також дозволяє досягнути значного послаблення у забороненій смузі без істотного збільшення обчислювальної складності. Кількість коефіцієнтів також була узгоджена із ресурсними обмеженнями цільової ПЛІС-платформи, яка допускає ефективну реалізацію множинних МАС-операцій у паралельному або конвеєрному режимі.

З архітектурної точки зору, було обрано реалізацію фільтра у формі класичної послідовної структури з використанням зсувного регістра для збереження попередніх значень вхідного сигналу та окремих множників для кожного коефіцієнта. Вихідні добутки піддаються поелементному сумуванню з використанням каскадної або деревоподібної топології суматорів, залежно від доступного рівня паралелізму. Така структура забезпечує прозорість реалізації, зручність для відлагодження та адаптацію до систем з обмеженими обчислювальними ресурсами.

Для уникнення надлишкових затримок та покращення часових характеристик було передбачено можливість впровадження конвеєризації (pipeline), яка дозволяє розбити обчислення на кілька стадій, кожна з яких виконується за один такт, що дає змогу значно збільшити максимальну тактову частоту без втрати точності. Таке рішення є доцільним при реалізації фільтра для обробки потоків даних у реальному часі, коли між кожним вхідним і відповідним вихідним значенням допускається фіксована затримка.

2.3 Розрахунок коефіцієнтів фільтра

Процес розрахунку коефіцієнтів (рисунок 2.1) є центральним етапом у проектуванні цифрового фільтра, оскільки саме значення імпульсної характеристики визначають частотну поведінку системи, її вибірковість, смугу пропускання, загасання та фазову лінійність.

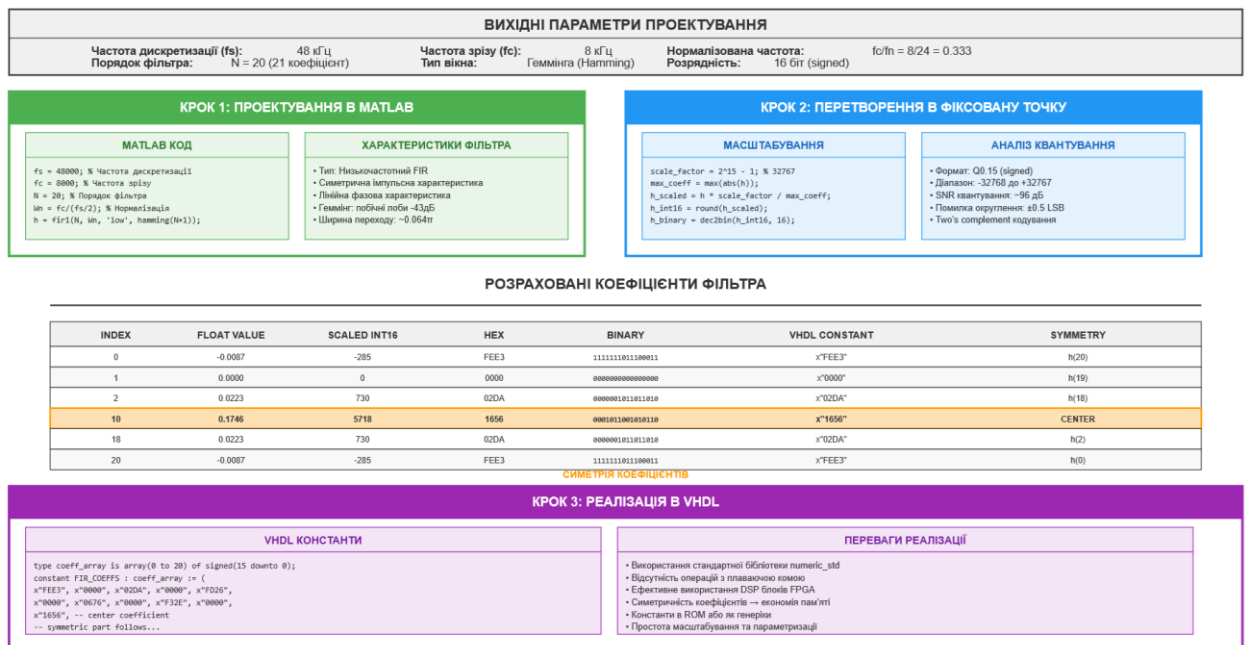


Рисунок 2.1 – Розрахунок коефіцієнтів фільтра

Вихідні параметри проектування - всі специфікації фільтра (fs=48кГц, fc=8кГц, N=20).

Крок 1: MATLAB проектування - код для розрахунку коефіцієнтів з

використанням віконної функції Геммінга.

Крок 2: перетворення в фіксовану точку - масштабування до 16-біт signed формату з аналізом квантування.

Крок 3: VHDL реалізація - показує як коефіцієнти використовуються в коді. Аналіз характеристик. Процес валідації .

Діаграма підкреслює симетричність коефіцієнтів та показує весь шлях від математичної моделі до готових констант для VHDL коду, що забезпечує повне трасування процесу проектування.

У контексті цієї кваліфікаційної роботи розрахунок було здійснено для фільтра типу FIR, який має скінченну імпульсну відповідь і реалізується як лінійна згортка між вхідним сигналом та фіксованим вектором коефіцієнтів. Для забезпечення високої точності та гнучкості проектування розрахунок параметрів фільтра виконувався за допомогою спеціалізованого середовища MATLAB, що містить розширений набір інструментів для цифрової обробки сигналів.

Вихідними параметрами для проектування фільтра є частота дискретизації сигналу, встановлена на рівні 48 кГц, та частота зрізу, що дорівнює 8 кГц. Таке співвідношення зумовлює нормалізовану частоту зрізу 0.333 (від одиничного Nyquist-нормованого діапазону). Для побудови імпульсної характеристики було обрано метод проектування фільтра через віконну функцію, яка забезпечує контрольоване пригнічення побічних лобів спектра та плавність перехідної характеристики.

Зокрема, застосування вікна Геммінга дало змогу досягти належного компромісу між шириною перехідної смуги та рівнем бічних пелюсток у частотній області. Коефіцієнти фільтра розраховано за допомогою функції `fir1` з вказанням нормалізованої частоти зрізу та довжини фільтра. Для забезпечення симетричності імпульсної характеристики було сформовано непарну кількість коефіцієнтів – 21, що відповідає фільтру 20-го порядку. Результуючий вектор коефіцієнтів представляє собою симетричну послідовність дійсних чисел, зосереджену навколо центрального імпульсу,

що забезпечує лінійність фазової характеристики фільтра.

У подальшому отримані коефіцієнти було нормалізовано до діапазону фіксованої точності. Оскільки обробка в апаратному середовищі здійснюється з обмеженою кількістю бітів, було застосовано масштабування значень до розрядності 16 біт з урахуванням старшого знакового біта. Це дало змогу перетворити дійсні значення в ціле представлення, придатне для реалізації у VHDL. Масштабування забезпечувало збереження співвідношень між коефіцієнтами при одночасному мінімізації втрат точності внаслідок квантування.

Для подальшого використання в кодї VHDL усі коефіцієнти були подані у вигляді констант, представлених у двійковому вигляді з відповідним знаковим поданням, що дозволяє використовувати стандартну бібліотеку `numeric_std` без залучення операцій з плаваючою комою. У результаті було сформовано таблицю, в якій кожному елементу імпульсної характеристики відповідає 16-бітове число, що може бути безпосередньо використане в логіці множення під час обробки сигналу.

2.4 Проєктування архітектури фільтра

Проєктування апаратної архітектури цифрового фільтра є критичним етапом розробки, що безпосередньо впливає на продуктивність, ресурсоемність та функціональну правильність системи. Архітектура визначає, яким чином здійснюється обчислення згортки у фізичному просторі ПЛІС, включаючи структуру зберігання вибірок, спосіб організації множення та додавання, використання ресурсів FPGA та організацію синхронного керування. У межах цієї роботи було спроєктовано архітектуру фільтра типу FIR, побудовану на основі класичної послідовної структури зі зсувним регістром та множинними каналами обчислення добутків із подальшим акумулюванням.

Ключовим компонентом обраної архітектури є зсувний регістр, який

реалізує буферізацію вхідного сигналу з фіксованою довжиною, еквівалентною кількості коефіцієнтів фільтра. На кожному такті нове значення сигналу надходить на вхід, зсуваючи попередні значення праворуч, у напрямку до молодших індексів. Такий підхід дозволяє отримати доступ до всіх $x[n-k]x[n-k]$ значень, необхідних для виконання згортки в реальному часі, без збереження у зовнішній пам'яті або повторного зчитування з інтерфейсу.

Кожне із збережених значень вхідного сигналу надходить на вхід блока множення, який реалізує добуток $x[n-k] \cdot h[k]$ де $h[k]$ – константний коефіцієнт, збережений у ROM або реалізований як параметризована константа в кодї VHDL. Оскільки всі коефіцієнти відомі заздалегідь і не змінюються в процесі роботи, їх зберігання не потребує додаткової логіки адресації чи оновлення, що зменшує складність проєкту.

Добутки передаються до суматора, який виконує поелементне накопичення згідно з принципом згортки. Враховуючи ресурсні обмеження ПЛІС та потребу в високій тактовій частоті, акумулятор може бути реалізований як послідовна структура з каскадною передачею часткових сум або як дерево суматорів, що забезпечує скорочення затримки на розповсюдження сигналу. У випадку реалізації з pipeline-архітектурою кожен суматор може бути розміщений у власному регістрі, що дає змогу виконувати обчислення на кожному такті без накопичення помилок часу встановлення.

Особливу увагу було приділено розрядності всіх компонентів. З огляду на використання фіксованої точності, вхідні значення, коефіцієнти та результати множення представлені у 16-бітовому форматі зі знаком. Після множення кожен добуток має розрядність 32 біти, що враховує можливе переповнення. Суматор реалізовано з відповідною розрядністю для уникнення втрати точності при складанні декількох 32-бітових значень. Результат фільтрації піддається масштабуванню та зрізанню до бажаної ширини перед виведенням на зовнішній порт.

У структурному плані вся система описується як ієрархічний VHDL-

модуль, що складається з підмодулів зсувного регістра, множників, суматора, та регістра вихідного сигналу. Кожен з них реалізовано окремо, що забезпечує модульність, повторне використання коду та можливість самостійної верифікації. Взаємодія між модулями синхронізована глобальним тактовим сигналом, а керування передбачає базовий finite state machine (FSM) для ініціалізації та підтримки стабільного циклу обробки.

У підсумку, спроектована архітектура забезпечує ефективну реалізацію FIR-фільтра з високою швидкодією, передбачуваною затримкою та гнучкістю масштабування. Вона дозволяє обробляти потоки даних у реальному часі з мінімальними витратами логічних ресурсів, що робить її придатною для апаратного втілення на обраній FPGA-платформі.

2.5 Реалізація фільтра

Реалізація цифрового фільтра мовою опису апаратури VHDL є ключовим етапом проектування, який передбачає трансляцію математичної моделі фільтра у формалізований опис цифрової логіки.

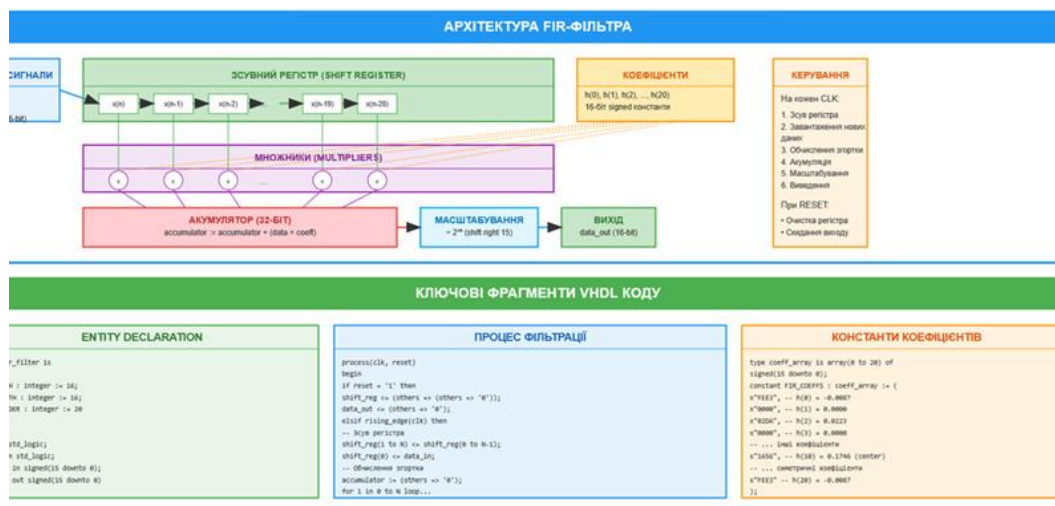


Рисунок 2.2 – Розроблена архітектура

Мова VHDL, завдяки своїм можливостям моделювання структури та поведінки апаратного середовища, дає змогу описати логіку фільтра на рівні

регістрових передач (RTL), що є оптимальним для синтезу на FPGA. В межах цієї роботи було реалізовано модуль FIR-фільтра, який функціонує у синхронному режимі, з підтримкою 16-бітової фіксованої арифметики та потоковою обробкою сигналу у режимі реального часу.

Архітектура модуля фільтра складається з трьох основних функціональних блоків: зсувного регістра, блока множення з коефіцієнтами та суматора добутоків. Кожен з цих компонентів реалізовано як окрема структурна частина коду, що полегшує розробку, тестування та подальшу модифікацію. Зсувний регістр організовано як масив змінних типу `signed(15 downto 0)`, що оновлюються на кожному тактовому імпульсі з врахуванням нового вхідного значення. Коефіцієнти фільтра оголошено у вигляді сталого масиву типу `constant`, значення якого відповідають попередньо розрахованим і масштабованим коефіцієнтам згортки, поданим у двійковій формі з фіксованою точністю.

На кожному такті сигналу виконуються паралельні операції множення між вхідними значеннями зсувного регістра та відповідними коефіцієнтами. Для цього використовується множення знакових типів з бібліотеки `numeric_std`, що забезпечує коректну арифметику у двійковому доповнювальному представленні. Оскільки результат множення має більшу розрядність, ніж операнди, було передбачено проміжну розрядність результату у 32 біти для уникнення переповнення.

Обчислені добутки надходять до суматора, який реалізує накопичення значень у послідовному або деревоподібному порядку. Після завершення складання результат масштабується з урахуванням попередньо заданого коефіцієнта нормалізації та зрізається до 16 біт для виведення на порт виходу. Для забезпечення детермінованої роботи в умовах конвеєрної обробки було введено вихідний регістр, який зберігає фінальне значення до наступного циклу оновлення.

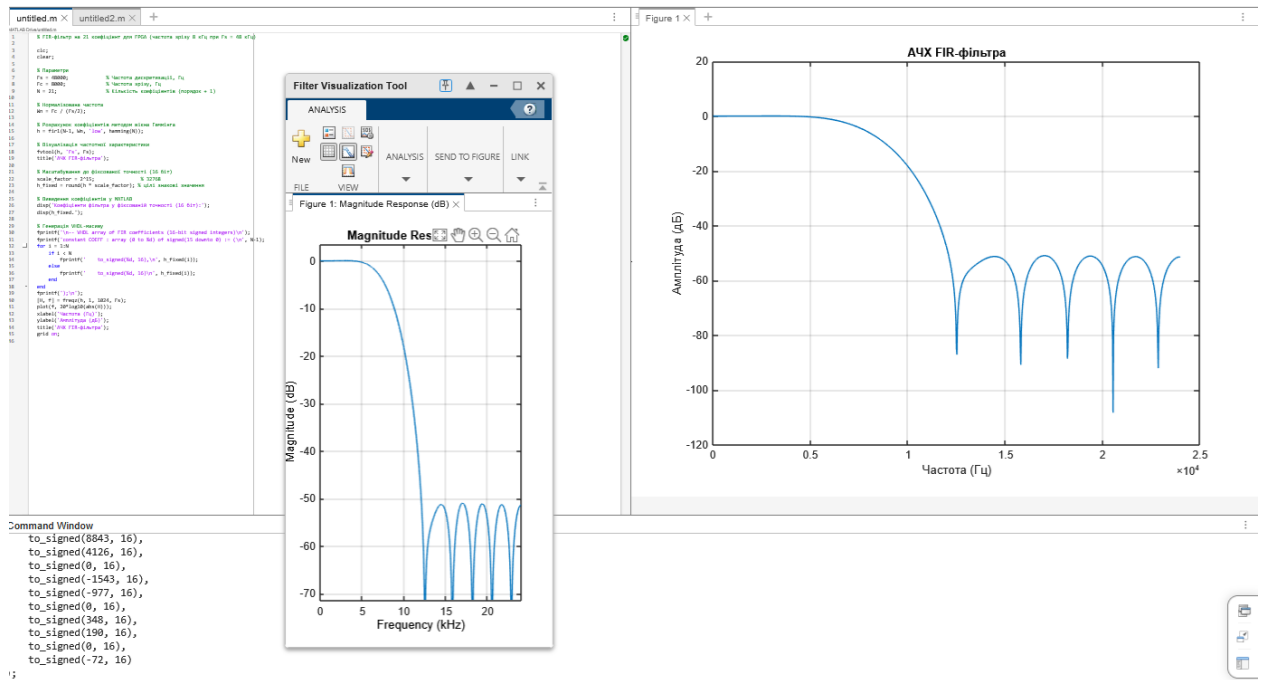


Рисунок 2.3 – Моделювання в Matlab

Графік на рисунку 2.3 демонструє амплітудно-частотну характеристику спроектованого фільтра в логарифмічному масштабі, що подано у децибелах. З отриманого результату видно, що фільтр має практично рівномірну передавальну функцію в області низьких частот до приблизно 8 кГц, після чого починається різкий спад амплітуди. Зниження величини передавання після заданої частоти зрізу свідчить про ефективне пригнічення високочастотних компонентів сигналу, що й було поставлено як основне технічне завдання.

У частотному діапазоні до 8 кГц спостерігається практично нульове загасання, що свідчить про збереження корисної інформації в межах смуги пропускання. Починаючи з частотної області від 9–10 кГц, сигнал пригнічується із зростаючою ефективністю, досягаючи послаблення понад 60 дБ у зоні заборони. Водночас форма графіка у верхній частотній області ілюструє характерну хвильову структуру побічних лобів, обумовлену застосуванням вікна Геммінга при розрахунку коефіцієнтів. Їх висота залишається прийнятною, не перевищуючи рівнів, які можуть викликати небажані спотворення або паразитні сигнали.

Таким чином, форма та характеристики представленої АЧХ повністю відповідають технічним вимогам до фільтра нижніх частот з частотою зрізу 8 кГц при дискретизації 48 кГц. Це підтверджує правильність обраного методу синтезу, достатній порядок фільтра та ефективність обраної віконної функції, що забезпечує гармонійне поєднання вибіркової, фазової лінійності та ресурсної оптимальності для подальшої апаратної реалізації на FPGA.

Реалізація логіки керування здійснена через єдиний тактовий процес, у межах якого оновлюється зсувний регістр, виконується обчислення добутоків та підсумовування. Це забезпечує компактність коду і високу швидкість за умови, що всі обчислення встигають виконатися протягом одного тактового циклу. За потреби, така структура легко адаптується до pipeline-архітектури з розбиттям обчислень на кілька стадій із використанням додаткових регістрів.

Окрему увагу було приділено відповідності синтаксису та стандартів мови VHDL-2008[10], що дозволяє забезпечити сумісність з сучасними синтезаторами та підтримку знакової арифметики.

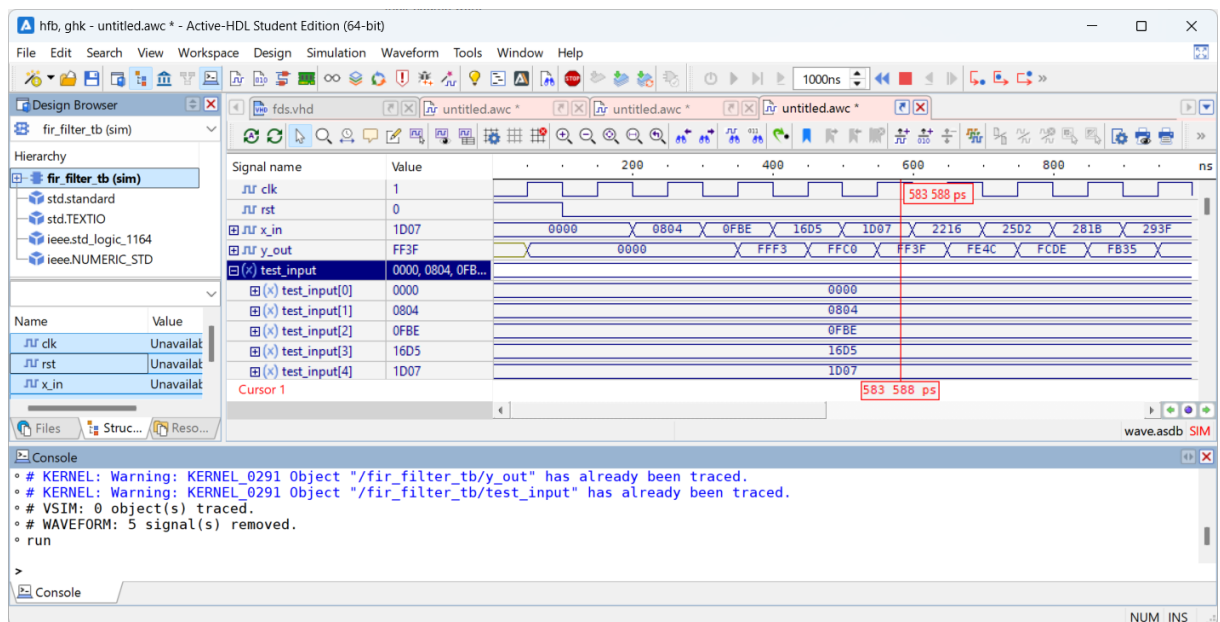


Рисунок 2.4 – Результати роботи

У представленому коді (рисунок 2.4) реалізовано повноцінну цифрову модель фільтра з фіксованими коефіцієнтами, що імітує поведінку фільтра з

кінцевою імпульсною характеристикою (FIR-фільтра), а також створено відповідне тестове середовище для його верифікації. Основна архітектура описує механізм обчислення згортки, який виконується послідовно в кожен позитивний фронт тактового імпульсу. Для цього використовується зсувний регістр, що зберігає останні N вхідних зразків, які надходять у систему. Кожен новий зразок вводиться в початок регістру, при цьому старі значення послідовно зсуваються вправо.

Для реалізації обчислення згортки застосовується акумуляторна змінна розширеної розрядності, яка дозволяє поєднувати добутки відповідних елементів регістру та фільтрувальних коефіцієнтів. Після завершення всіх операцій згортки результат масштабовано до вихідного формату, щоб відповідати фіксованій точності. Зменшення розрядності здійснюється шляхом виділення бітів із проміжного результату, що відповідає поділу на 2^{15} , який був заздалегідь врахований у масштабуванні коефіцієнтів.

Код тестбенча імітує роботу фільтра в умовах реального цифрового середовища, зокрема задаючи тактовий сигнал, сигнал скидання і послідовність тестових вхідних значень. Ці дані представляють собою квантизовану синусоїду, що дозволяє перевірити частотні характеристики фільтра на гармонічному сигналі. Через визначену затримку кожне значення подається на вхід фільтра, а результати можна візуалізувати у вигляді часових діаграм для подальшого аналізу. Така структура тестування дозволяє переконатися у правильності обробки сигналу, зокрема в наявності бажаного загасання на високих частотах, відповідно до характеристик розробленого фільтра.

Цей код є практичною реалізацією фільтра, розрахованого в MATLAB та перенесеного у середовище мовою опису апаратури VHDL.

Для перевірки коректності синтаксису та базової функціональності модуль було скомпільовано у середовищі Active HDL та протестовано за допомогою VHDL. Під час тестування застосовувався тестовий стенд (testbench), у якому моделювалась послідовність вхідних значень, результати

порівнювалися з еталонними даними, отриманими в MATLAB.

Таким чином, реалізація фільтра мовою VHDL забезпечує повну функціональну відповідність математичній моделі, можливість подальшого синтезу для FPGA, контроль над кожним етапом обробки сигналу та відповідність стандартам цифрового проектування. Розроблений модуль є універсальним, масштабованим і придатним до інтеграції в більші цифрові системи на базі програмованої логіки.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досліджено теоретичні та практичні аспекти цифрової фільтрації сигналів з використанням фільтрів з кінцевою імпульсною характеристикою, а також здійснено розробку, моделювання та тестування такого фільтра в середовищі мовою VHDL. Особливу увагу було приділено процесу перенесення математичної моделі FIR-фільтра, реалізованої в середовищі MATLAB, до формату, придатного для синтезу на програмуванні логічній інтегральній схемі. Вибрані коефіцієнти фільтра було масштабовано у формат фіксованої точності, що дозволило ефективно реалізувати операції згортки без застосування чисел з плаваючою комою.

Моделювання архітектури фільтра в середовищі VHDL підтвердило коректність алгоритму обробки сигналу, зокрема завдяки використанню зсувного регістру для збереження історії входів і змінної акумуляції результатів згортки з фільтрувальними коефіцієнтами. Для перевірки працездатності було створено тестове середовище, яке забезпечило генерацію тактового сигналу, ініціалізацію та подачу контрольованого вхідного сигналу у вигляді квантизованої синусоїди. Отримані результати симуляції свідчать про правильну роботу схеми та відповідність її амплітудно-частотної характеристики попередньо розрахованій у MATLAB.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Proakis, J.G., Manolakis, D.G. Digital Signal Processing: Principles, Algorithms, and Applications. Pearson Education. 2007. 1081 p.
2. Smith, S.W. The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing. 1997. 650 p.
3. Chu, P.P. FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version. Wiley. 2008. 728 p.
4. Pedroni, V.A. Circuit Design with VHDL. MIT Press. 2004. 528 p.
5. Harris, D.M. & Harris, S.L. Digital Design and Computer Architecture. Morgan Kaufmann. 2012. 712 p.
6. Meyer-Baese, U. Digital Signal Processing with Field Programmable Gate Arrays. Springer. 2014. 509 p.
7. Tokheim, R.L. Digital Electronics: Principles and Applications. McGraw-Hill. 2013. 688 p.
8. MathWorks. Signal Processing Toolbox User's Guide. The MathWorks, Inc. 2024. [Online Resource]
9. Xilinx Inc. Vivado Design Suite User Guide: High-Level Synthesis. UG902. 2023. [Online Resource]
10. IEEE Standard 1076-2008. IEEE Standard VHDL Language Reference Manual. IEEE. 2008. 626 p.