

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)

Моделі та методи проектування апаратного біт-потокowego  
обчислювача ірраціональних функцій

(тема)

Виконав:

студент II курсу, групи СКСм-20-1  
Кузніченко Т.Б.  
(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: доц. Ларченко Л.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ

(підпис)

Чумаченко С.В.

(прізвище, ініціали)

2021 р.

# Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
Кафедра Автоматизації проектування обчислювальної техніки  
Рівень вищої освіти другий (магістерський)  
Спеціальність 123 – Комп'ютерна інженерія  
(код і повна 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 АТЕСТАЦІЙНУ РОБОТУ

студентові Кузніченко Тарасу Богдановичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Моделі та методи проектування апаратного біт-потокowego обчислювача ірраціональних функцій

затверджена наказом по університету від " 04 " 11 2021 р. № 1635 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15.12.2021

3. Вихідні дані до роботи

САПР Active-HDL

Мова опису апаратури VHDL

FPGA кристал сімейства Xilinx Spartan-3E серії XC3S500E

4. Перелік питань, що потрібно опрацювати в роботі

1 Біт-потоківі функціональні обчислювачі в системах управління

2 Декомпозиція математичної моделі обчислювача ірраціональних функцій

3 Архітектура обчислювача ірраціональних функцій

4 Апаратна реалізація обчислювача ірраціональних функцій

5 Розробка автоматної HDL-моделі обчислювача та верифікація проекту

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 20 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	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2021 - 02.09.2021	
2	Аналіз предметної області	03.09.2021 - 15.09.2021	
3	Аналіз джерел з проблемної галузі	16.09.2021 - 01.10.2021	
4	Розробка математичної моделі біт-потокowego обчислювача ірраціональних функцій	02.10.2021 - 15.10.2021	
5	Розробка архітектури обчислювача ірраціональних функцій	16.10.2021 - 30.10.2021	
6	Проектування апаратної моделі пристрою на основі кінцевого автомату	01.11.2021 - 14.11.2021	
7	Розробка автоматних HDL-моделей обчислювача та верифікація проекту	15.11.2021 - 18.11.2021	
8	Оформлення пояснювальної записки	19.11.2021 - 29.11.2021	
9	Оформлення графічного матеріалу	30.11.2021 - 07.12.2021	
10	Перевірка виконаного проекту керівником	08.12.2021 - 15.12.2021	

Дата видачі завдання 01 вересня 2021 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Ларченко Л.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 77 сторінок, 17 рисунків, 12 таблиць, 16 джерел за переліком посилань.

БІТОВИЙ ПОТІК ДАНИХ, ФУНКЦІОНАЛЬНЕ ПЕРЕТВОРЕННЯ,  
БІТ-ПОТОКОВИЙ ОБЧИСЛЮВАЧ, ІРРАЦІОНАЛЬНА ФУНКЦІЯ,  
КОНВЕЄРНА АРХІТЕКТУРА, АПРОКСИМАЦІЯ, АЛГОРИТМ, VHDL-  
МОДЕЛЬ, ВЕРИФІКАЦІЯ

В кваліфікаційній роботі спроектовано та досліджено апаратний біт-потоківий обчислювач ірраціональних функцій. Об'єктом дослідження є біт-потоківі обчислювачі ірраціональних функцій.

В процесі виконання роботи проаналізовано метод ступінчастої апроксимації функцій, на основі зворотних функцій, розроблено математичну модель біт-потоківого обчислювача ірраціональних функцій, що є декомпозицією математичних моделей обчислювачів степеневих та лінійних функцій. Розроблено архітектуру досліджуваного пристрою на основі швидкодіючих конвеєрних структур, що є синтезом двох блоків: біт-потоківого степеневого обчислювача та дільника чисел, побудованого на структурі обчислювача лінійних функцій.

Розроблено апаратну реалізацію пристрою на основі кінцевого автомату, дано теоретичні розрахунки для виконання експериментального дослідження пристрою, отримано граф-схему алгоритму реалізації ірраціональної функції і граф переходів керуючого автомату обчислювача, розроблено HDL-моделі пристрою на основі автоматних шаблонів, виконано верифікацію отриманого рішення, синтез та імплементацію пристрою в ПЛІС Xilinx.

## ABSTRACT

The explanatory note contains 77 pages, 17 figures, 12 tables, 16 sources by the list of references.

BIT-STREAM DATA, FUNCTIONAL CONVERSION, BIT-STREAM COMPUTER, IRRATIONAL FUNCTION, PIPELANE ARCHITECTURE, APROXIMATION, ALGORITHM, VHDL-MODEL, VERIFICATION

In the qualification work the hardware bit-stream computer of irrational functions was researched and designed. The object of study is bit-stream computers of irrational functions. In the process of work the method of step approximation of functions on the basis of inverse functions was analyzed, the mathematical model of the bit-stream computer of irrational functions was developed, which is the decomposition of mathematical models of computers of power and linear functions.

The architecture of the researched device on the basis of high-speed pipeline structures was developed, which is a synthesis of two blocks: bit-stream power computer and number divider, built on the structure of the linear function computer.

The hardware implementation of the device on the basis of the finite state machine was developed, theoretical calculations for performance of experimental research of the device are given. The graph-scheme of the algorithm of realization of the irrational function and the graph of transitions of the automatic control of the computer were obtained, HDL-models of the device on the basis of automatic templates were developed, verification of the received solution was executed, synthesised and implemented as a device in Xilinx FPGA.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 БІТ-ПОТОКОВІ ФУНКЦІОНАЛЬНІ ОБЧИСЛЮВАЧІ ДЛЯ СИСТЕМ УПРАВЛІННЯ.....	11
1.1 Форма представлення бітових-потоків даних.....	11
1.2 Призначення та місце біт-потоків функціональних обчислювачів в системах управління.....	13
1.3 Постановка завдання.....	18
2 ДЕКОМПОЗИЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА ІРРАЦІОНАЛЬНИХ ФУНКЦІЙ.....	20
2.1 Метод ступінчастої апроксимації неперервних висхідних функцій.....	20
2.2 Математичне обґрунтування обчислення ірраціональних функцій.....	24
2.3 Математична модель біт-потоків обчислювача ірраціональних функцій.....	27
3 СИНТЕЗ АРХІТЕКТУРИ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА ІРРАЦІОНАЛЬНИХ ФУНКЦІЙ.....	34
3.1 Узагальнена архітектура біт-потоків обчислювача ірраціональних функцій.....	34
3.2 Синтез архітектури досліджуваного обчислювача ірраціональних функцій.....	37
3.3 Вибір технологічної платформи для апаратної реалізації.....	44
4 АПАРАТНА РЕАЛІЗАЦІЯ ОБЧИСЛЮВАЧА ІРРАЦІОНАЛЬНИХ ФУНКЦІЙ.....	48
4.1 Результати обчислювального процесу в компонентах пристрою..	48

4.2 Граф-схема алгоритму та граф переходів керуючого автомату обчислювача.....	55
4.3 Структурно-блокова схема пристрою.....	58
4.4. Опис проекту на МОА.....	62
4.5 Верифікація поведінкової моделі обчислювача.....	66
4.6 Порівняльний аналіз частотних характеристик ірраціональних обчислювачів.....	72
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	76
ДОДАТОК А Графічна частина кваліфікаційної роботи.....	78
ДОДАТОК Б Лістинг коду програм.....	88

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВІС – велика інтегральна схема

ВП – вторинний перетворювач

ГСА – граф-схема алгоритму

ЕОМ – електронно-обчислювальна машина

КСАУ – комп'ютерна система управління

ІВС – інтелектуальна вимірювальна система

ІКС – інформаційно-керуюча система

ІС – інтегральна схема

МОА – мова опису апаратури

НВІС – надвелика інтегральна схема

ПЛІС – програмована логічна інтегральна схема

ПСО – пристрій сполучення з об'єктом

САПР – система автоматизації проектування

СУ – система управління

ТЗ – технічне завдання

ШІМ – широтно-імпульсна модуляція

ЦФП – цифрові функціональні перетворювачі

CLB (Configurable Logic Block) – програмований логічний блок;

IOB (Input/output blocks) – блок введення – виведення;

FPGA (Field Programmable Gate Arrays) – програмована користувачем  
вентильна матриця;

SDH (Synchronous Digital Hierarchy) – принцип побудови цифрових систем  
передачі;

## ВСТУП

Системи управління реального часу мають у своєму складі підсистеми обробки потокової інформації, що включають в себе множину функціональних перетворювачів і обчислювачів для вирішення завдань математичної обробки та лінеаризації сигналів, що отримують з сенсорів фізичних величин.

Одним з актуальних напрямків, пов'язаних з розробкою сучасних удосконалених і нових базових елементів для побудови названих систем є розробка апаратних обчислювачів функціональної обробки сигналів, які представляють собою бітові потоки. Здійснення обчислювальних операцій у пристроях з такою формою подання вхідних сигналів виконується більш простими способами. Основна мета розвитку архітектур апаратних обчислювачів систем управління (СУ), що виконують потокову обробку є спрощення структур, блоків та компонентів обчислювачів та перетворювачів завдяки застосуванню інформаційних сигналів у вигляді вхідної бітової послідовності.

При проведенні математичної обробки інформаційних сигналів у вимірювальних підсистемах СУ за допомогою вторинних перетворювачів і обчислювачів, що отримують з вимірювальних сенсорів, часто є потрібним здійснення лінеаризації цифрових сигналів, а саме, реалізацію різних нелінійних перетворень.

Цифрові функціональні обчислювачі дають можливість спростити КСАУ за допомогою використання простих і надійних пристроїв. Цифрові функціональні обчислювачі, які можуть бути імплементовані у платформу ПЛІС дають можливість підвищити швидкодію пристроїв, використовувати уніфіковані елементи і блоки та підвищити точність обчислень та перетворень. Отже, актуальне і важливе завдання проектування пристроїв – це розробка і

дослідження обчислювачів математичних функцій, інформаційними сигналами яких є бітові потоки даних.

Метою атестаційної роботи є дослідження та розробка моделей та методів проектування біт-потокowego обчислювача ірраціональних функцій на основі технологічної платформи ПЛІС з використанням САПР на основі мов опису апаратури.

Біт-потокowy обчислювач може бути застосований в розподілених системах управління і контролю, в інтелектуальних вимірювальних системах в якості функціональних перетворювачів при вимірюванні діючих значень напруг і струмів, а також в приборах вимірювання вібрації обертового обладнання при вимірюванні середньоквадратичних значень віброшвидкості та віброприскорення агрегатів з обертовими частинами.

# 1 БІТ-ПОТОКОВІ ФУНКЦІОНАЛЬНІ ОБЧИСЛЮВАЧІ ДЛЯ СИСТЕМ УПРАВЛІННЯ

У розділі дано поняття бітових потоків даних, розглянуто призначення та місце біт-потоків обчислювачів математичних функцій в системах управління реального часу, сформульовано мету та постановку завдання дослідження.

## 1.1 Форма представлення бітових потоків даних

В системах управління реального часу сигнали отримують від сенсорів та датчиків фізичних величин. На основі результатів вимірювань здійснюють виконання завдань управління. При обробці сигналів з сенсорів і датчиків часто потрібно виконання різних нелінійних перетворень інформаційних сигналів, що представлені бітовими потоками, при цьому цифрова обробка бітових потоків здійснюється функціональними перетворювачами, а також обчислювачами.

У розподілених СУ часто доцільно застосовувати не аналого-цифрове перетворення сигналів сенсорів, а перетворення аналогового сигналу в частоту, оскільки передача частотних сигналів має перевагу у розподілених системах: такі сигнали мають підвищену завадостійкість [1].

Форми бітового потоку припускають уявлення даних у вигляді потоків імпульсів з одиничною амплітудою. Дані в таких потоках представлені таким чином:

- за кількістю окремих імпульсів за одиницю часу;
- потоками сигналів широтно-імпульсної модуляції (ШІМ), в яких безперервна тривалість одиничного значення, пов'язаного з періодом імпульсів, визначає значення, що передаються.

Інформаційні потоки сигналів пов'язані з кількістю імпульсів (біт), що проходять у одному потоці за одиницю часу. Крім того інформаційні потоки пов'язані з тривалістю одиничного біта.

Біт-потокове кодування передбачає, що дані представлені потоками імпульсів з одиничною амплітудою частотних, час-імпульсних, широтно-модульованих сигналів.

Інформативний параметр в біт-потоківому кодуванні - це фіксоване значення імпульсів (біт)  $X_{\max}$  за часовий інтервал  $T$  (рис 1.1).

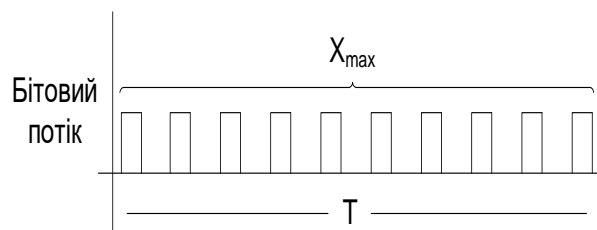


Рисунок 1.1. – Бітовий потік даних

Вхідний і вихідний інформаційні сигнали біт-потоківих обчислювачів являють собою два бітових (імпульсних) потоки, періодичність проходження біт вхідного потоку визначається способом квантування відтворюваної функції по аргументу. При цьому має місце рівномірне квантування аргументу в цілих числах.

Бітові потоки можуть здійснювати передачу і обробку інформації такими способами, які дозволяють виконувати послідовну обробку потоків даних у темпі надходження на вхід функціональних обчислювачів одиночних імпульсів (біт). При цьому форма сигналів, представлених бітовим потоком має високу завадостійкість, яка є слідством непозиційності та еквівалентності одиничних імпульсів [2]. При такій формі сигналів забезпечується висока швидкодія пристроїв, тому що відсутня інформаційна надмірність сигналів.

Для систем управління і контролю є актуальним розробка відповідних пристроїв, та певні підходи до організації обчислювальних перетворень [3].

Носіями інформаційних даних можуть бути електричні, оптичні, пневматичні, біологічні та інші сигнали залежно від типу реалізації.

В сучасних розподілених системах серед чутливих елементів широко представлені сенсори з частотним виходом, вони здійснюють пряме перетворення фізичної аналогової величини в число-імпульсний або широтно-модульований сигнал. Перспективи подальшого розвитку та впровадження чутливих елементів частотного типу оцінюються позитивно, цьому багато в чому сприяє об'єднанню досліджувальників, розробників та великих фірм-виробників сенсорів та датчиків до міжнародної асоціації IFSA (International Frequency Sensor Association). Ця організація забезпечує професійне обговорення результатів досліджень, обмін досвідом у проектуванні та технологіях у сфері розробки різних інтелектуальних датчиків із цифровим, частотним, час-імпульсним виходом, що сприятиме збільшенню кількості та розширенню номенклатури частотних датчиків, що випускаються.

Технічна реалізація підходів обробки бітового потоку при використанні електричних сигналів орієнтована на цифрову елементну базу та дозволяє застосовувати програмовані логічні мікросхеми (CPLD та FPGA). Внаслідок подальшої розробки запропонованих методів відтворення математичних функцій та збільшенню циркуляції типових пристроїв, стане можливим перехід на спеціально замовлені мікросхеми (ASIC). Застосування зазначеної елементної бази забезпечить високу технологічну надійність блоків обробки інформаційних даних бітового потоку [4].

## 1.2 Призначення та місце біт-потоків функціональних обчислювачів в системах управління

Сучасні системи управління (рис.1.2) є розподіленими системами, в яких основою комплексу технічних засобів є обчислювальні системи, що здатні реалізувати основні функції СУ:

- прийняти та провести аналіз інформаційних даних про стан об'єкту управління;
- здійснити обробку даних, виконати порівняння інформації, що була отримана з задачами управління;
- здійснити за результатами порівняння відповідні керуючі сигнали на об'єкт управління.

Система управління має у своєму складі компоненти:

- об'єкт управління;
- датчики, сенсори, первинні і вторинні перетворювачі;
- виконавчі пристрої (актуатори);
- підсистема управління реального часу.



Рисунок 1.2 – Система управління

В системі управління об'єкт управління має технічні стани.

Об'єкт управління, сенсори та перетворювачі створюють інтерфейс зв'язку. Інтерфейси зв'язку СУ мають у своєму складі спеціалізовані пристрої – функціональні обчислювачі, що працюють з потоковими формами даних.

Найбільш ефективною буде система управління, архітектура якої відповідає класу розв'язуваних завдань та динамічним характеристикам об'єкта управління. Керуючі СУ мають у своєму складі не тільки штатні набори зовнішніх пристроїв, а й пристрої спеціалізованого типу –

функціональні перетворювачі та обчислювачі, що реалізують певні елементарні функції.

В архітектурі таких СУ можна виділити підсистеми первинної та вторинної обробки інформації. Об'єкт управління генерує аналогові та дискретні сигнали стану (вектор стану), отримує аналогові та дискретні сигнали управління (вектор управління) від керуючої системи [5].

Пристрої сполучення з об'єктом (ПСО) управління виконують перетворення сигналів внутрішньої мови об'єкта, сигнали внутрішньої мови обчислювальної системи і зворотне перетворення. УСО – це спеціалізований інтерфейс.

Основне функціональне призначення підсистеми первинної цифрової обробки інформації – перед процесорна та пост процесорна обробка сигналів: декодування, демодуляція, цифрова фільтрація та кореляція, дискретне перетворення Фур'є, лінеаризація сигналів тощо. Тут вирішуються та обробляються завдання: кодування, декодування, генерація, модуляція та інші.

Важливою особливістю первинної обробки є те, що вона повинна проводитися в темпі надходження вхідних даних, щоб унеможливити інформаційні втрати, тобто у реальному масштабі часу [6].

Традиційним способом обробки інформаційних сигналів, які надходять від сенсорних елементів, які мають частотний вихід, є прямі обчислення. Прямі обчислення мають на увазі перетворення частотних сигналів. Далі передбачено перетворення частотних сигналів з виконанням необхідних математичних дій та операцій.

При використанні сенсорів часто є необхідним виконання лінеаризації сигналів. При цьому Існує необхідність обчислювати різні нелінійні функції. Таким чином, проектування апаратних засобів, які виконують обчислювальні перетворення біт-потоківих даних є актуальною.

За допомогою функціональних обчислювачів та перетворювачів на БІС та НВІС можна будувати розподілені системи, наближені безпосередньо до джерел даних та об'єктів управління. Організація засобів інформаційно –

керуючих систем (ІКС) у вигляді розподіленої системи забезпечує роботу в реальному часі, технічну досконалість архітектури обчислювальних засобів та на цій основі їх швидку модернізацію з мінімальними витратами; можливість нарощування продуктивності обчислювальних засобів; можливість створення адаптивних систем; реалізацію порівняно простих інформаційних зв'язків між розподіленими підсистемами по об'єкту; вбудованість апаратних засобів безпосередньо в об'єкти управління; підвищення стійкості до відмов системи та покращення ремонтпридатності систем.

Обчислювальні засоби підсистеми первинної обробки інформації повинні мати високу продуктивність. Домогтися високої продуктивності можна лише шляхом структурної організації за рахунок паралелізму алгоритмів завдань, вирішуваних підсистемою первинної обробки інформації.

Для вимірювання цілого ряду параметрів технологічних процесів (рівень фізичних середовищ, щільність, тиск, швидкість, температура, рідина) широко використовуються частотні сенсори та датчики з поданням вимірювальних даних у вигляді частотних та час-імпульсних сигналів. Важливим напрямом підвищення точності та надійності вимірювання параметрів є розвиток функціональних можливостей перетворювачів, що здійснюють первинне перетворення та вторинну математичну обробку імпульсних та частотних сигналів при вирішенні завдань лінеаризації.

За допомогою спеціалізованих цифрових функціональних перетворювачів (ЦФП) однією з поширених обчислювальних операцій є здійснення розгортання у реальному масштабі часу неперервних часових залежностей. У багатьох випадках цифрові функціональні перетворювачі виконують обчислювальні операції, які забезпечують процес перетворення та обробки інформаційних сигналів, що поступають на обробку з датчиків в системах управління та інтелектуальних вимірювальних системах.

Функціональні біт-потоківі обчислювачі, що розглядаються, побудовані на основі цифрового функціонального перетворення, що передбачає послідовне обчислення значень функції, що виконуються для сусідніх значень

аргументу. Таке перетворення має назву розгортуючого. У розгортках є можливість враховувати попередню історію обчислень: кожне чергове значення відтворюваної функції обчислюють на основі попереднього обчислення. В таких пристроях перше обчислення функції здійснюється з врахуванням початкових значень [7].

Серед датчиків сучасних систем управління та контролю значну частину становлять частотні датчики, в яких вихідним інформативним параметром є частота імпульсної послідовності. Тому актуальними залишаються проблеми вимірювання частотно-часових параметрів сигналів. Використання частотних датчиків забезпечує більш просте перетворення частоти в цифровий код. Такі завдання ставляться при вимірюванні постійного і диференціального тиску в трубопроводах, швидкостей, переміщень, прискорень в авіаційній техніці при вимірюванні вібрацій, при вимірюванні температури [8, 9].

Особливо широке застосування функціональні обчислювачі з біт-потоким кодуванням знайшли при проектуванні цифрової контрольно-вимірювальної апаратури, коли пристрої використовуються в якості обчислювальних вузлів у цифрових вимірювальних приладах. Функціональні обчислювачі використовуються у амперметрах, вольтметрах, аналізаторах спектрів. Вони реалізують різні функції між вхідними та вихідними величинами. Використовуються також при побудові керованих генераторів імпульсів низьких частот на основі перетворювачів код-фаза, код-частота. Виконання обчислювальних операцій в таких перетворювачах здійснюється вирішенням завдання отримання нової послідовності імпульсів (біт) частоти, пов'язаної певною функціональною залежністю з частотами вхідної бітової послідовності.

При проведенні математичної обробки для вирішення завдань первинної вимірювальної інформації у підсистемах СУ, зокрема, інформаційно-вимірювальних системах (ІВС) поряд з арифметичними, алгебраїчними операціями буває часто необхідним виконання нелінійних перетворень частоти імпульсних послідовностей. Вирішувати такі завдання доводиться при

лінеаризації функцій перетворення частотних датчиків, при виробленні нелінійних поправок в результаті вимірювання вплив зовнішніх невимірюваних параметрів, при вирішенні завдань непрямого вимірювання, при отриманні сигналів, що коригують в СУ.

Досліджуваний апаратний біт-потоківий обчислювач, може знайти застосування в якості обчислювального вузла у інтелектуальних вимірювальних системах і приборах для вимірювання діючих значень напруг і струмів, в приборах вимірювання вібрації обертового обладнання при вимірюванні середньоквадратичних значень віброшвидкості та віброприскорення агрегатів з обертовими частинами та в системах управління та контролю в якості цифрових функціональних перетворювачів.

### 1.3 Постановка завдання

Мета кваліфікаційної роботи – дослідження та розробка моделей та методів проектування спеціалізованого апаратного біт-потоківого обчислювача ірраціональних функцій на технологічній платформі ПЛІС з використанням мов опису апаратури.

Об'єкт дослідження – апаратні обчислювачі ірраціональних функцій з бітовими потоками даних. Предмет дослідження – математичні, структурні та автоматні моделі біт-потоківого обчислювача ірраціональних функцій.

Мета визначила наступні завдання:

- аналіз форми представлення бітових потоків даних в апаратних обчислювачах математичних функцій;
- аналіз методу ступінчастої апроксимації відтворення неперервних висхідних функцій на основі зворотних функцій;
- розробка математичної моделі біт-потоківого обчислювача ірраціональних функцій;
- аналіз способу побудови конвеєрних архітектур біт-потоківих обчислювачів;

- синтез архітектури обчислювача заданої ірраціональної функції;
- розробка апаратної реалізації пристрою на основі кінцевого автомату;
- розробка HDL-моделей обчислювача;
- верифікація та імплементація моделі пристрою з використанням САПР цифрових пристроїв XILINX ISE в платформу ПЛІС.

## 2 ДЕКОМПОЗИЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА ІРРАЦІОНАЛЬНИХ ФУНКЦІЙ

Розділ присвячено розробці математичної моделі обчислювача ірраціональних функцій, що є декомпозицією математичної моделі степеневого біт-потокowego обчислювача та обчислювача лінійних функцій. Для отримання математичної моделі обчислювача використано метод ступінчастої апроксимації неперервних функцій на основі зворотних функцій.

### 2.1 Метод ступінчастої апроксимації на основі зворотних функцій

При побудові нелінійних біт-потокowych обчислювачів та перетворювачів, призначених для відтворення елементарних функцій, використовуються методи обчислення апроксимуючих функцій, що є принципово-точними та наближеними. При цьому застосовуються різні апроксимуючі функції.

Абсолютна похибка апроксимації є визначальною, що впливає на результат відтворення функцій.

Принципово-точні методи дають більш точний результат апроксимації оскільки не мають методичної похибки і використовують різні методи, зокрема на основі алгебраїчних рівнянь – визначенні зворотної функції.

При проектуванні біт-потокowego обчислювача ірраціональних функцій використано математичний апарат ступінчастої апроксимації неперервних висхідних функцій на основі зворотних, що покладений в основу побудови обчислювача.

В [10] розглянуто метод ступінчастої апроксимації, який передбачає формування приростів ступінчастих функцій, що апроксимують безперервні функції, який застосовується при побудові обчислювачів елементарних

функцій. При цьому безперервні функції  $y^* = f(x^*)$ , що відтворюються функціональними обчислювачами є монотонно зростаючими з обмеженнями:

$$x^*, y^* \geq 0; y^* \leq x^*; \frac{dy^*}{dx^*} > 0, \quad (2.1)$$

функція  $y^* = f(x^*)$  має зворотну  $x^* = \psi(y^*)$ .

Вхідний  $x$  і вихідний  $y$  інформаційні сигнали біт-потоків обчислювачів – дві періодичні бітові послідовності  $x$  та  $y$ . При синтезі функціональних обчислювачів час відтворення та абсолютна похибка обчислення мають бути раціональними. Тобто з точки зору точності обчислення функції раціональним є забезпечення для всіх цілочисельних значень аргументу значення мінімальної абсолютної похибки обчислення функції, що дорівнює .

Метод ступінчастої апроксимації математичних елементарних функцій на основі зворотних, передбачає процес обчислення цілочисельних значень. При цьому здійснюється операція округлення дробових значень ґратчастої функції. Округлення дробових значень здійснюється до цілих чисел, яке відбувається в цілочисельних точках аргументу.

На виході обчислювача значення вихідної ступінчастої функції можуть бути отримані за допомогою апроксимуючі функції:

$$y = [f(x) + |\delta_{\max}|], \quad (2.2)$$

де аргумент функції приймає значення  $x = 1, 2, 3, \dots, i$ ;

$|\delta_{\max}|$  - граничне значення абсолютної похибки обчислення для неперервної функції. Квадратні дужки у виразі (2.2) означають цілу частину числа.

На рис. 2.1 зображені неперервна та її апроксимуюча функції.

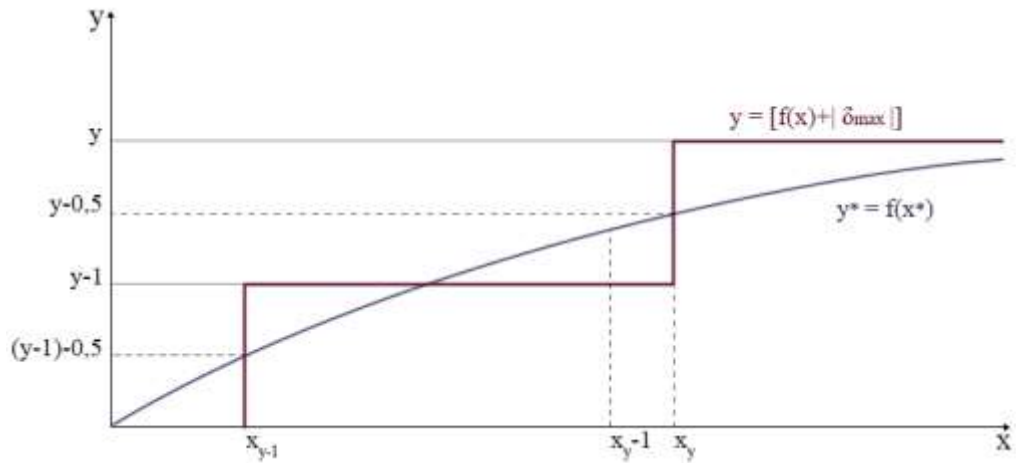


Рисунок 2.1 – Ступінчаста апроксимуюча функція

Моментом початку формування кожної сходинки функції  $y = 1, 2, 3, \dots, k$ , що апроксимує безперервну, відповідають значення аргументу - вибірок  $x_y = x_1, x_2, x_3, \dots, x_k$ , біти, що обираються з вхідного бітового потоку  $x$ . На графіку формування кожної сходинки апроксимуючої функції здійснюється у точках перетину неперервної кривої та рівнів  $y - |\delta_{\max}|$ .

Для апроксимуючої функції з обмеженням  $y \leq x$  для будь-якого рівня  $y - |\delta_{\max}|$ , коли  $y = 1, 2, 3, \dots, k$  для пари сусідніх цілочисельних значень аргументу  $x_{y-1}$  і  $x_y$ :

$$\begin{cases} f(x_{y-1}) < y - |\delta_{\max}| \\ f(x_y) \geq y - |\delta_{\max}|, \end{cases} \quad (2.3)$$

де  $f(x_y)$  - значення функції в точці  $x_y$ ,

$f(x_{y-1})$  - значення функції в точці  $x_{y-1}$ .

Процес відтворення апроксимуючої функції (2.2) може бути здійснений на основі вибірки певного числа біт  $x_y$  з вхідного потоку  $x$ .

Номери вибірок визначають за допомогою системи нерівностей (2.3). В результаті формулу для визначення  $x_y$  має вигляд:

$$\Psi(y - |\delta_{\max}|) \leq x_y < \Psi(y - |\delta_{\max}|) + 1, \quad (2.4)$$

де  $\Psi(y - |\delta_{\max}|)$  – функція, зворотна  $f(x)$ .

Нерівність (2.4) – основна нерівність знаходження вибірових значень  $x_y$ . Значення вибірок  $x_y$ , можуть бути знайдені при послідовній підстановці  $y = 1, 2, 3, \dots, k$  в нерівність (2.4).

Нерівності (2.4) можна замінити рівністю:

$$x_y = [\Psi(y) - |\delta_{\max}|] + 1, \quad (2.5)$$

де  $y = 1, 2, 3, \dots, k$ .

При похибці обчислень  $|\delta_{\max}| = 0,5$  отримано формули для знаходження значення вибірок  $x_y$ , які визначаються:

$$\Psi(y - 0,5) \leq x_y < \Psi(y - 0,5) + 1, \quad (2.6)$$

$$x_y = [\Psi(y - 0,5)] + 1, \quad (2.7)$$

$$x_y = \Psi(y - 0,5). \quad (2.8)$$

Вирази (2.6), (2.7), (2.8) відповідають раціональному варіанту вибірки за точністю обчислення значень функції, які обчислюються в цілочисельних точках аргументу.

## 2.2 Математичне обґрунтування обчислення ірраціональних функцій

Неперервні ірраціональні функції з обмеженнями (2.1) можуть бути відтворені на виході біт-потокowego ірраціонального обчислювача апроксимуючими ступінчастими функціями.

В системах управління при обробки бітових потоків, що отримують від сенсорів фізичних величин виникає завдання вторинного перетворення бітового потоку при виконанні лінеаризації сигналу, зокрема, визначення діючих (середньоквадратичних) значень при вимірюванні вібрації, віброприскорення та періодичних несинусоїдних струмів і напруг.

В спеціалізованих функціональних обчислювачах ірраціональну функцію реалізують також в аналізаторах спектра частотно-модульованих сигналів [11].

Ірраціональна функція, що може бути відтворена на виході обчислювача має вигляд:

$$y = \left[ \sqrt{\left[ \frac{\sum_{i=0}^n x_i^2}{n} + |\delta_{1\max}| \right] + |\delta_{2\max}|} \right], \quad (2.9)$$

де  $x_i$  – пакки бітового (імпульсного) потоку;

$n$  – кількість пачок імпульсів;

$|\delta_{1\max}|$  – граничне значення абсолютної похибки, що виникає при діленні полінома на число  $n$ ;

$|\delta_{2\max}|$  – граничне значення абсолютної похибки добування квадратного кореня.

При цьому якщо похибки обчислення дробово-раціональної функції в підкореневому виразі (2.9)  $|\delta_{1\max}| = |\delta_{2\max}| = 0,5$ , вони є мінімальними.

З урахуванням  $|\delta_{1\max}| = |\delta_{2\max}| = 0,5$  можна записати, що апроксимуюча ірраціональна функція, яка реалізується обчислювачем має вигляд:

$$y = \left[ \sqrt{\left[ \frac{\sum_{i=0}^n x_i^2}{n} + 0,5 \right] + 0,5} \right]. \quad (2.10)$$

При обчисленні ірраціональної функції (2.9) необхідно виконати наступні операції над вхідними пачками бітового потоку в два етапи:

– на першому етапі необхідно обчислювати поліном другого степеню і здійснювати ділення результату на константу  $n$  у підкореневому

виразі  $\frac{\sum_{i=0}^n x_i^2}{n}$ . Названі обчислювальні операції можна поєднати в біт-

потоківому обчислювачі дробово-раціональних функцій;

– на другому етапі необхідно добувати корінь з результату, отриманого на першому етапі обчислень, що можна здійснити в пристрої добування кореня.

Аналіз показав, що дробово-раціональний обчислювач заданої функції при подачі на вхід пристрою перших біт вхідного потоку  $x$  працює в режимі вибірки певних номерів біт, а далі переходить в режим генерації пачок бітових послідовностей. При цьому в моменти генерації пачки біт на виході пристрою не можна подавати вхідні біти для функціональної обробки сигналу до тих пір, поки не буде сформована пачка вихідних біт. Це обмежує частоту вхідного бітового потоку і є недоліком даного пристрою.

Альтернативою обчислювача на основі дробово-раціонального модуля було розроблено обчислювач на основі степеневого модуля, що дозволяє розширити частотний діапазон вхідного бітового потоку.

З ірраціональної функції випливає, що розширити частотні можливості обчислювача можна забезпечити відповідним числом пачок бітового потоку.

Було визначено при аналізі функції, що якщо число  $n$  є точним квадратом числа  $p$ , тобто  $n = p^2$ , то функція (2.9) набуває вигляду:

$$y = \left[ \frac{1}{p} \left[ \sqrt{\sum_{i=1}^n x_i^2} + |\delta_{2\max}| \right] + |\delta_{1\max}| \right] \quad (2.11)$$

де  $x_i$  – пакки (серії) бітового потоку;

$n$  – кількість пакчок;

$|\delta_{1\max}|$  – значення похибки поділу функції  $\sqrt{\sum_{i=1}^n x_i^2}$  на  $p$ ;

$|\delta_{2\max}|$  – граничне значення абсолютної похибки добування кореня квадратного з  $\sum_{i=1}^n x_i^2$ .

Пристрій обчислення ірраціональних функцій має обчислювати функцію із заданою абсолютною похибкою обчислень  $|\delta_{1\max}| = |\delta_{2\max}| = 0,5$ :

$$y = \left[ \frac{1}{p} \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right] + 0,5 \right]. \quad (2.12)$$

Обчислення з похибкою  $|\delta_{\max}| = 0,5$  забезпечує мінімальну похибку апроксимації відтворення заданої функції, що є раціональною.

Вхідним інформаційним сигналом обчислювача є бітовий потік  $x$  (серії біт), на виході пристрою формується бітовий потік  $y$ , що відтворює безперервну досліджувану ірраціональну функцію.

Обчислення ірраціональної функції виконується при здійсненні наступних операцій над вхідним бітовим потоком  $x$ :

- піднесення до степеню  $i$  добування кореня у виразі  $\sqrt{\sum_{i=1}^n x_i^2}$ ;
- поділу отриманого результату на ціле число  $p$ .

В біт-потоківому обчислювачі ірраціональних функцій обчислення досліджуваної функції здійснюється в два етапи:

а) виконуються операції піднесення до степеню, а також добування кореня, тобто обчислюється степенева функція з дробовим показником. Тому названі обчислювальні операції можна поєднати в біт-потоківому обчислювачі степеневих функцій;

б) здійснюється операція ділення проміжного результату обчислення степеневої функції на константу, яке може бути здійснено в обчислювачі лінійних функцій.

Виходячи з того, що обчислення досліджуваної функції відбувається в два етапи, математична модель біт-потоківого обчислювача ірраціональної функції може бути представлена декомпозицією математичної моделі біт-потоківого обчислювача степеневих функцій та математичною моделлю обчислювача лінійних функцій.

### 2.3 Математична модель біт-потоківого обчислювача ірраціональних функцій

При розробці математичної моделі біт-потоківого обчислювача заданої ірраціональної функції здійснено декомпозицію математичної моделі.

Декомпозиція математичної моделі обчислювача досліджуваної ірраціональної функції складається з математичної моделі біт-потоківого обчислювача степеневої функції та математичної моделі обчислювача лінійної функції.

Розглянемо отримані математичні моделі названих обчислювачів.

Досліджувана ірраціональна функція має вигляд:

$$y = \left[ \frac{1}{p} \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right] + 0,5 \right], \quad (2.13)$$

де  $x_i$  – пакки бітових послідовностей.

$|\delta_{1\max}| = |\delta_{2\max}| = 0,5$  – абсолютна похибка обчислення.

На першому етапі ірраціональним обчислювачем реалізується проміжна апроксимуюча степенева функція, що має вигляд:

$$y = \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right]. \quad (2.14)$$

Функція (2.14) представляє інтерес, так як при реалізації обчислювачем даної функції розв'язується практична задача – добування квадратного кореня з суми квадратів чисел, представлених пачками бітових послідовностей.

При розробці математичної моделі степеневого обчислювача функція, що реалізується в пристрої має вигляд:

$$y = \left[ \sqrt{x^2} + 0,5 \right] \quad (2.15)$$

Нерівність, що реалізується в степеневому обчислювачі отримана на основі методу ступінчастої апроксимації, що базується на визначенні зворотної функції, що реалізуються у функціональних перетворювачах та обчислювачах.

Визначивши зворотну функцію для функції (2.15) отримано нерівність:

$$2^2 x_y^2 \geq (2y_k - 1)^2. \quad (2.16)$$

Для того, щоб зменшити розрядність компонентів структури доцільно порівнювати прирости відтворюваних двох функцій нерівності (2.16).



$(2y_k - 1)^2$  з урахуванням різниці  $\Delta_{y-1}$  обчислення, яка отримана на попередньому етапі обчислення.

При надходженні на вхід пристрою біта з номером  $X_y$  на виході пристрою буде сформований вихідний біт  $U_k$ , коли буде виконана кожна нерівність у (2.17).

В ірраціональному обчислювачі, що досліджується, на другому етапі здійснення обчислень необхідно виконувати ділення проміжного результату обчислення степеневі функції на константу  $p$ . При цьому враховується похибка  $|\delta_{\max}| = 0,5$ . Обчислення може бути реалізовано з використанням обчислювача лінійних функцій, який буде відтворювати функцію:

$$y = \left[ \frac{1}{p} x + 0,5 \right], \quad (2.20)$$

де  $x$  – бітовий потік, що надходить на вхід обчислювача лінійних функцій із виходу степеневого обчислювача проміжної функції.

Для отримання математичної моделі обчислювача лінійних функцій, скористаємося методом ступінчастої апроксимації функцій.

Отже, визначивши зворотну функцію для функції (2.20), з урахуванням похибки  $|\delta_{\max}| = \frac{1}{2}$ , нерівність для вибірок  $X_y$  має вигляд:

$$\frac{p(2y-1)}{2} \leq x_y < \frac{p(2y-1)}{2} + 1. \quad (2.21)$$

де  $y = 1, 2, 3, \dots$  – номер вибірки;  $x_y$  - номер обраного біту, який з'явиться на виході пристрою.

Ліва частина нерівності має вигляд

$$2x_y \geq p(2y_k - 1). \quad (2.22)$$

У нерівності (2.22) значення  $x_y$  визначаються шляхом обчислення приростів функції  $2x_y$ , що формується на кожному з інтервалів сходинки апроксимуючої функції. Порівнянні приростів з приростами функції  $p(2y_k - 1)$  з урахуванням різниці обчислень  $\Delta_{y-1}$ , що може бути отримана на попередньому кроці обчислень при виконанні (2.22).

Підставляючи в праву частину нерівності (2.22) значення  $y = 1, 2, 3, \dots$ , отримаємо арифметичний ряд:  $p, 3p, 5p, 7p, 9p, \dots$

Для отриманого ряду арифметичний ряд різниць 1-го порядку буде дорівнюватиме константі  $2p$ .

Перейдемо до системи нерівностей, що представляє собою математичну модель обчислювача лінійних функцій, записану в різницях:

$$\begin{aligned} 2x_1 &\geq p \\ 2(x_2 - x_1) + \Delta_1 &\geq 2p, \\ 2(x_3 - x_2) + \Delta_2 &\geq 2p, \\ &\dots\dots\dots, \\ 2(x_y - x_{y-1}) + \Delta_{y-1} &\geq 2p, \end{aligned} \quad (2.23)$$

де

$$\Delta_{y-1} = 2(x_y - x_{y-1}) + \Delta_{y-2} - 2p. \quad (2.24)$$

З (2.23) випливає, що першому імпульсу  $y = 1$  відповідає імпульс з номером  $x_1$ , що обирається з вхідного потоку  $x$ , при якому буде виконана перша нерівність системи. Аналогічно другому обраному імпульсу  $y = 2$  відповідає імпульс з номером  $x_2$ , при якому виконається друга нерівність і так далі.

У розділі розглянуто математичні моделі пристроїв обчислення степеневих та лінійних функцій, архітектурний синтез яких дозволив реалізувати досліджувану ірраціональну функцію.

При обчисленні заданої функції використовується відомий алгоритм обчислення поліномів, що використовується при побудові базової структури поліноміального обчислювача. Цей обчислювач використовується при побудові архітектури досліджуваного обчислювача.

Для поліноміальної функції, що має вигляд:

$$y = \sum_{i=0}^n a_i x^i, \quad (2.25)$$

при розгляданні алгоритму обчислення поліномів, використовуючи конвеєрні обчислення, математична модель поліноміального обчислювача має вигляд:

$$\begin{aligned} y_i &= f(i+1) - f(i), \\ \Delta_i &= y_{i+1} - y_i, \\ \Delta_i^2 &= \Delta_{i+1} - \Delta_i, \\ \Delta_i^3 &= \Delta_{i+1}^2 - \Delta_i^2, \\ &\dots \dots \dots \\ \Delta_i^n &= \Delta_{i+1}^{n-1} - \Delta_i^{n-1}, \end{aligned} \quad (2.26)$$

де  $i = 0, 1, 2, 3, \dots$ ,  $f(i+1)$  – значення функції при  $x_{i+1}$ ,  $f(i)$  – значення функції при  $x_i$  відповідно.

Синтез конвеєрного біт-потокowego обчислювача поліноміальних функцій здійснюється за допомогою зниження порядку різниць.

Компоненти конвеєрної архітектури, побудованої на суматорах, ініціалізуються значеннями  $y_0, \Delta_0, \Delta_0^2, \dots, \Delta_0^n$ , тобто, першими членами арифметичних рядів 1-го, 2-го, ..., n-го порядків.

Досліджувана ірраціональна функція в підкореновому виразі має обчислювати суму квадратів, тому необхідно обчислювати поліноміальну функцію, що має вигляд:

$$y = x^2. \quad (2.27)$$

При підстановці у функцію значень  $x = 1, 2, 3, \dots$  значення функції у утворюють числову послідовність  $1^2, 2^2, 3^2, 4^2, \dots$ . Ця послідовність являє арифметичний ряд 2-го порядку. У результаті обчислень арифметичні ряди різниць 1-го і 2-го порядків визначаються, як

$$\Delta \quad 2^2 - 1^2, 3^2 - 2^2, 4^2 - 3^2, \dots$$

$$\Delta^2 \quad 2, 2, 2, \dots$$

При проектуванні біт-потокowego обчислювача ірраціональних функцій було використано алгоритм відтворення поліноміальної функції (2.27).

### 3 СИНТЕЗ АРХІТЕКТУРИ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА ІРРАЦІОНАЛЬНИХ ФУНКЦІЙ

В розділі розглянуто узагальнена архітектура біт-потоків обчислювачів ірраціональних функцій, синтез архітектури біт-потоків обчислювача досліджуваної ірраціональної функції, що складається з блоку біт-потоків степеневих обчислювачів і блоку обчислення лінійних функцій. У пристрої використано конвеєрну архітектуру поліноміального обчислювача. Дано опис блоків пристрою та принцип його роботи, обрано елементну базу для апаратної реалізації обчислювача.

#### 3.1 Узагальнена архітектура біт-потоків обчислювачів ірраціональних функцій

Узагальнену архітектуру біт-потоків обчислювачів ірраціональних функцій (рис.3.1) приведено в [12]. Архітектура може бути використана для побудови при проектуванні досліджуваного біт-потоків обчислювача ірраціональних функцій.

Узагальнена архітектура дозволяє здійснювати синтез структур обчислювачів, що реалізують ірраціональні функції визначеного класу, в яких  $x$ ,  $y$  є бітовими потоками даних.

У архітектурах обчислювачів використовуються уніфіковані блоки при їх технічній реалізації.

Математичні моделі обчислювачів ірраціональних функцій в загальному вигляді представляють собою системи різницевої нерівності. При цьому кожна нерівність системи математичної моделі описує умову появи на виході пристрою вихідного біта  $y$ .

Порівняння пристроїв відтворюваних функцій, що є у лівій та правій частини нерівностей математичних моделей пристроїв відбувається при реалізації кожної нерівності.

Узагальнена архітектура обчислювача ірраціональних функцій дозволяє реалізувати математичні моделі пристроїв, що відтворюють ірраціональні функції.

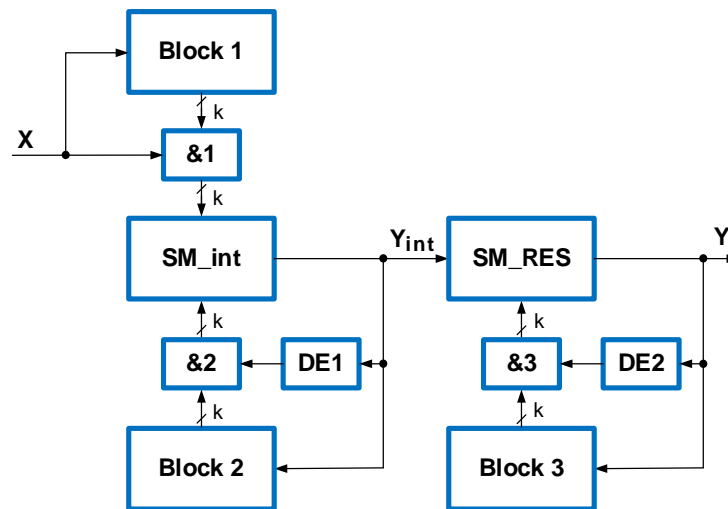


Рисунок 3.1 – Узагальнена архітектура біт-потокowego обчислювача ірраціональних функцій

Обчислення ірраціональної функції у даній структурі виконується послідовно у два етапи.

1. На першому етапі обчислюється значення функції  $Y_{int} := f(x)$ , що є проміжною, а саме, результатом обчислення степеневі функції.

2. Другий етап передбачає обчислення значення ірраціональної функції. Вона є результатом ділення на константу результату обчислення функції на попередньому етапі. На виході пристрою формується результат обчислення ірраціональної функції, а отже,  $y = f(x)$ .

Загальна розгорнута архітектура ірраціонального обчислювача містить два модулі (рис.3.2).

Перший модуль містить блоки Block1, Block2, а також суматор SM\_int. На його виході формується вихідний проміжний бітовий потік  $y_{int} := f(x)$ , який являє собою результат обчислення проміжної степеневі функції.

Другий модуль структури містить блок Block3 і суматор результату SM\_RES. Він реалізує ділення значення проміжної степеневі функції на число  $p$ .

Блоки структури Block1, Block2 – це конвеєрні архітектури, що являють собою базові архітектури обчислювача поліноміальних функцій.

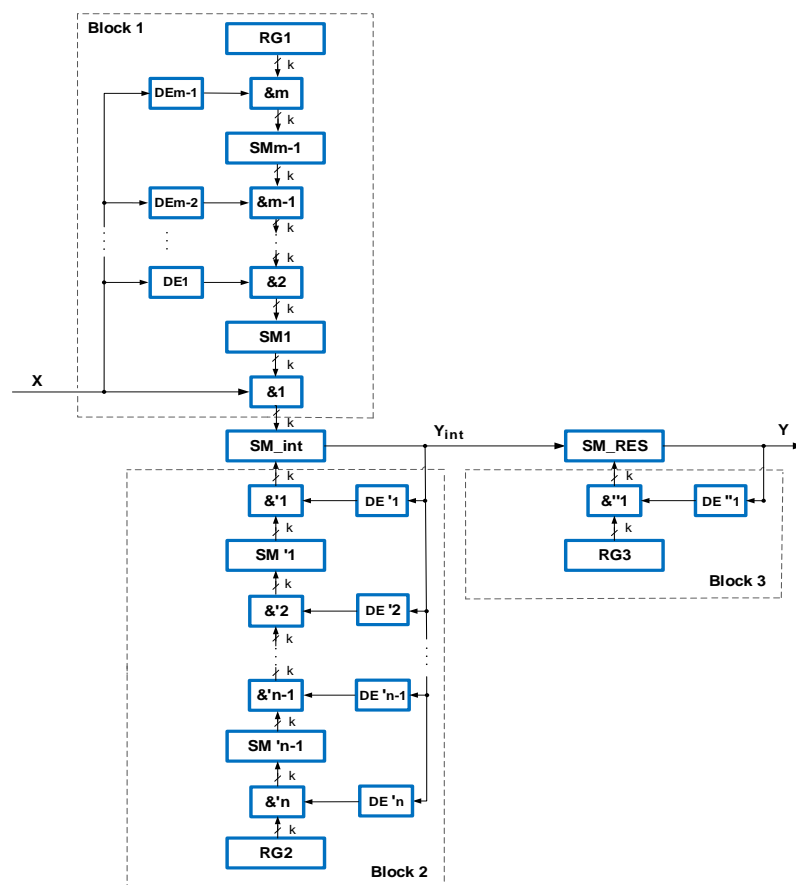


Рисунок 3.2 – Загальна архітектура біт-потокowego пристрою обчислення ірраціональних функцій на основі степеневого модулю

Основні обчислювальні компоненти архітектури – це паралельні суматори SM\_int і SM\_RES зі зворотним зв'язком. Вони використовуються як елементи порівняння ґратчастих функцій нерівностей математичних моделей

обчислювачів степеневі та лінійної функції.

Біти переповнення суматора SM\_RES є бітами у вихідного потоку пристрою та результатом обчислення ірраціональної функції, що задана. Поява вихідних бітів пристрою відповідає початку формування кожної із сходинок функції, що апроксимує неперервну.

### 3.2 Синтез архітектури досліджуваного обчислювача ірраціональних функцій

На основі загальної архітектури біт-потоків обчислювача ірраціональних функцій було розроблено архітектуру, яка реалізує досліджувану функцію (2.13) та наведена на рис. 3.3 [13, 14].

Розглянемо архітектуру та принцип роботи досліджуваного обчислювача.

Апроксимуюча функція, що реалізується обчислювачем ірраціональних функцій:

$$y = \left[ \frac{1}{p} \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right] + 0,5 \right], \quad (3.1)$$

де  $x_i$  – пакки імпульсних послідовностей.

На першому етапі обчислень виконуються операції піднесення до степеню та добування кореня, які суміщені в степеневому модулі.

На другому етапі обчислень виконується ділення степеневі функції на константу  $p$ , що реалізовано в архітектурі обчислювача лінійних функцій.

Розглянемо структуру і роботу досліджуваного пристрою (рис 3.3).

Архітектура містить 2 модулі:

1. Степеневий модуль, що включає суматори SM1, SM2, SM\_int, регістри RG1, RG2, групи елементів &1, &2, &3; &4 та елементи затримки DE1, DE2, DE3.

2. Дільник чисел, що містить суматор SM\_RES, регістр RG3, групу логічних елементів &5 та елемент затримки DE4.

I. Степеновий модуль. Степеновий модуль з дробовим показником обчислює функцію:

$$y_{\text{int}} = \left[ \sqrt{\sum_{i=1}^n x_i^2 + 0,5} \right]. \quad (3.2)$$

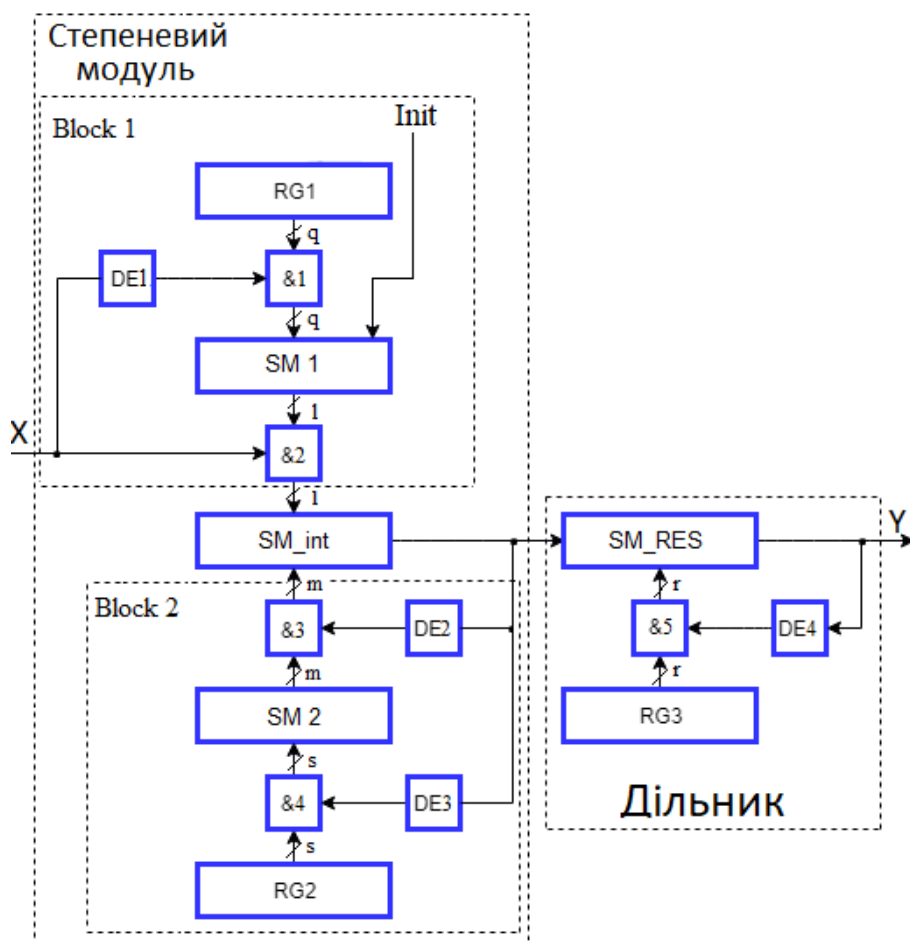


Рисунок 3.3 – Архітектура досліджуваного обчислювача ірраціональних функцій

В степеневому модулі реалізується система нерівностей (2.18). Основна нерівність, яке реалізується в приростах в обчислювачі має вигляд  $2^2 x_y^2 \geq (2y_1 - 1)^2$ .

З нерівності випливає, що значення результату обчислення степеневі функції можуть бути отримані шляхом формування гратчастої функції  $2^2 x_y^2$ , а також послідовному порівнянні поточних значень цієї функції з цілочисельними рівнями  $(2y-1)^2$ .

Пристрій містить два блоки: Block 1 реалізує функцію  $2^2 x_y^2$ , а Block 2 – функцію  $(2y-1)^2$ . Block 1 і Block 2 призначені для формування різниць першого порядку числових послідовностей відповідно.

Основним обчислювальним вузлом в степеневому модулі є суматор SM\_int, який використовується в якості елемента порівняння кодів приростів функції  $2^2 x_y^2$  з приростами функції  $(2y-1)^2$ .

Степневий модуль містить два блоки.

1. В перший блок входять суматор SM1, регістр RG1, елемент затримки DE1 і групи елементів &1, &2. Блок реалізує функцію  $2^2 x_y^2$ .
2. В другий блок входять SM2, регістр RG2, два елемента затримки DE2, DE3 і групи елементів &3; &4. Блок реалізує функцію  $(2y-1)^2$ .

Множення поточних значень функції на постійну  $2^2$  здійснюється внесенням в суматор SM\_int членів арифметичного ряду різниць першого порядку, які утворюються зсувом на 2 розряди вправо.

В процесі роботи пристрою в SM1 формуються члени арифметичного ряду різниць першого порядку числової послідовності функції  $x_y^2$ , а в суматорі SM2 - формуються члени арифметичного ряду різниць першого порядку функції  $(2y-1)^2$ .

Двійковий код числа, записаний в суматор SM1, надходить в SM\_int в прямому коді зі зсувом на 2 розряди вправо, а двійковий код числа, записаний в суматорі SM2, надходить в SM\_int в додатковому коді.

Ініціалізація компонентів степеневого модуля:

суматор SM\_int ініціалізується числом  $2^i - 1$ ;

суматор SM1 ініціалізується числом  $1^2$ ;

суматор SM2 - ініціалізується числом  $3^2 - 1^2$ ;

регістр RG1 ініціалізується константою, що була утворена в ряду різниць 2-го порядку для функції  $x_y^2$ , тобто двійковим кодом числа 2;

регістр RG2 ініціалізується константою, що була утворена в ряду різниць 2-го порядку для функції  $(2y - 1)^2$ , тобто двійковим кодом числа 8.

Вихід елемента затримки DE1 з'єднаний з групою елементів &2, за допомогою якої з регістра RG1 в SM1 кожним вхідним бітом послідовності  $x$  переноситься число 2! Перенесення прямого коду числа з SM1 в SM\_int потрібно здійснювати зі зсувом на два розряди вправо, що відповідає множенню числа  $x_y^2$  на  $2^2$ .

Вихід пристрою з'єднаний з групою елементів &4 через елемент затримки DE3 таким чином, що кожен вихідний біт SM\_int, що є вихідним бітом степеневого модуля, по ланцюгу зворотного зв'язку виконує наступні дії:

- вносить в суматор SM\_int з суматора SM2 паралельний двійковий додатковий код чисел, що утворюються в ряду різниць першого порядку функції  $(2y - 1)^2$ :

- вносить в суматор SM2 константу, яка зберігається в регістрі RG2, двійковий код числа 8.

Степневий модуль працює наступним чином. З приходом на вхід пристрою першого біту першої пачки бітового потоку  $x_i$  виконуються операції:

- прямий двійковий код числа  $1^2$  із SM1 за допомогою групи елементів &2 переноситься в SM\_int зі зсувом на два розряди вправо і підсумовується з його вмістом.

В результаті показання SM\_int стануть рівними  $(2^i - 1) + 2^2 \cdot 1^2 = \Delta_1$ . При цьому SM\_int переповниться, на його виході з'явиться біт переповнення і в SM\_int залишиться число  $\Delta_1$ .

–прямий код числа 2, записаного в RG1, за допомогою групи елементів &1 переноситься в SM1 і підсумовується з його вмістом. Отже, вміст SM1 стане рівним  $1^2 + 2 = 3$ . Тобто в SM1 утворюється двійковий код числа, що відповідає першому члену ряду різниць 1-го порядку. Далі при надходженні на вхід модуля бітового потоку x в SM1 будуть утворюватися поступово члени арифметичного ряду різниць 1-го порядку.

В результаті перший біт переповнення SM\_int буде з'являтися при надходженні на вхід обчислювача, що є степеневим обчислювачем, першого біту послідовності x. В результаті виконається перша нерівність системи (2.17)  $2^2 x_1^2 \geq (2y_1 - 1)^2$ .

Біт переповнення виконує наступні дії:

– з виходу SM\_int надходить на вихід степеневого модуля і далі на вхід модуля – дільника чисел;

– пройшовши через елемент затримки DE2, він відкриває групу елементів & 3. При цьому число, що зберігається в SM2  $3^2 - 1^2$  переноситься в SM\_int в додатковому коді і підсумовується з вмістом SM\_int. В результаті утворюється число  $2^i - (3^2 - 1^2) + \Delta_1$ . Так як  $\Delta_1 < 3^2 - 1^2$ , біт переповнення в SM\_int буде відсутній;

– пройшовши через елемент затримки DE3, він відкриває групу елементів &4. При цьому з регістру RG2 в SM\_int переноситься константа, що утворена в ряду різниць 2-го порядку функції  $(2y - 1)^2$  і підсумовується з його вмістом. В результаті в SM\_int буде сформоване число  $5^2 - 3^2$ . Тобто в SM\_int утворюються члени ряду різниць 1-го порядку функції  $(2y - 1)^2$ .

Надалі з надходженням чергових вхідних біт x описані процеси в модулі будуть циклічно повторюватися.

Другий біт переповнення SM\_int буде мати місце при надходженні на вхід пристрою буту з номером  $x_2$ , при якому виконається друга нерівність (2.19).

Другий біт переповнення надходить на вихід обчислювача та по ланцюгу зворотного зв'язку і забезпечує перенесення вміст регістру SM2 в SM\_int у додатковому коді. В результаті вміст SM\_int стане рівним  $2^i - (5^2 - 3^2) + 2^2 \cdot i^2$ . Цей же біт встановлює в SM2 число  $7^2 - 5^2$ .

Надалі описані процеси будуть циклічно повторюватися.

Якщо на вхід степеневого модуля подати  $n$  пачок біт  $x_i$  і після надходження на вхід пристрою кожної пачки  $x_i$  бітом «скидання» (initial state) суматор SM1 встановлювати в початковий стан, рівний  $1^2$ , то на виході SM\_int буде формуватися результат обчислення проміжної функції

$$y_{\text{int}} = \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right].$$

II. Дільник чисел. В архітектурі ірраціонального обчислювача, модуль дільника чисел реалізовано на основі обчислювача лінійних функцій, що виконує ділення значення проміжної степеневі функції на число  $p$ .

До складу модуля входять: суматор SM\_RES та регістр RG3, а також елемент затримки DE4 і група логічних елементів &5, що порозрядно з'єднує регістр і суматор.

Ініціалізація компонентів модуля дільника чисел:

суматор SM\_RES ініціалізується числом  $2^i - p$  в додатковому коді;

регістр RG3 ініціалізується числом  $2^i - 2p$ .

Опишемо роботу дільника чисел.

На вхід дільника надходять біти потоку проміжної степеневі функції з виходу SM\_int степеневого модуля, де  $x = y_{\text{int}} := f(x)$ .

З приходом першого біту послідовності  $x$  на вхід SM\_RES, його вміст буде збільшений на число 2 у двійковому коді, тобто рівним  $2^i - p + 2$ , так як кожним бітом вхідної послідовності  $x$ , в SM\_RES потрібно вносити число 2. В результаті біт переповнення SM\_RES відсутній.

Перший біт переповнення SM\_RES з'явиться при надходженні на вхід пристрою біту з номером  $x_1$ , при якому виконається перша нерівність (2.25)

$$2x_1 \geq p.$$

Біт переповнення виконує дії:

- з виходу SM\_RES надходить на вихід обчислювача ірраціональних функцій;

- пройшовши через елемент затримки DE4, він відкриває групу & 5. Число, що зберігається в RG3  $2^i - 2p$  записується в суматор SM\_RES у додатковому коді та здійснюється підсумовування з його вмістом. В результаті вміст SM\_RES дорівнює  $2^i - 2p$ .

Надалі з надходженням чергових вхідних біт  $x$  описані обчислювальні процеси в схемі будуть повторюватися.

Другий біт переповнення SM\_RES з'явиться на виході пристрою при надходженні на вхід обчислювача біту з номером  $x_2$ . При цьому буде виконана друга нерівність системи (2.26).

Архітектура запропонованого обчислювача на основі степеневого модуля має переваги відносно розширення частотного діапазону пристрою у порівнянні з прототипом обчислювача ірраціональних функцій на основі дробово-раціонального модуля, в якому на виході SM\_int утворюються пачки вихідного бітового потоку проміжної функції, що обмежує частоту вхідного бітового потоку, яких подається на вхід обчислювача.

В досліджуваному обчислювачі на основі степеневого модуля після кожного чергового переповнення SM\_int в ньому залишається число в прямому двійковому коді, яке менше числа внесеного по ланцюгу зворотного зв'язку з боку SM2 в додатковому коді, тому другий біт переповнення SM\_int з'явитися не може. В результаті з надходженням першої пачки біт на вхід пристрою кожному вхідному біту відповідає тільки один біт переповнення SM\_int, що не суперечить рівності  $\sqrt{x_1^2} = x_1$ . А так як після введення в пристрій пачки  $x_1$  суматор SM1 встановлюється в початковий стан, тобто в

значення ініціалізації «1», і в  $SM\_int$  залишається записаним в додатковому коді число більше одиниці, то з надходженням кожної наступної пачки  $X_i$  на вхід пристрою частота проходження біт на виході  $SM\_int$  буде зменшуватися, тому даний пристрій може працювати на більш високих частотах у порівнянні з аналогічним пристроєм, що реалізований на основі дробово-раціонального модуля.

### 3.3 Вибір технологічної платформи для апаратної реалізації

При створенні пристроїв цифрової обробки сигналів в основу завжди ставиться елементна база. В ході експериментальних досліджень було здійснено синтез та імплементацію апаратної моделі ірраціонального обчислювача в платформу ПЛІС. ПЛІС є одним з напрямків розвитку елементної бази цифрової електроніки. Їх застосування дає розробнику можливість швидко та з малими витратами створювати складні пристрої, багаторазово змінювати та вдосконалювати їх функції під час налагодження шляхом перепрограмування функцій та зв'язків елементів.

ПЛІС – цифрові НВІС великого степеню інтеграції, що мають програмовану внутрішню структуру і призначені для проектування складних цифрових пристроїв. Вони володіють не тільки максимальними абсолютними показниками продуктивності, а й найкращим співвідношенням продуктивності і ціни та мають здатність паралельно виконувати десятки і навіть сотні операцій та рекомендовані к використанню при реалізації конвеєрних обчислень. При цьому можливість реконфігурації ПЛІС не обмежує використання обчислювальних ресурсів.

ПЛІС має важливі переваги для використання. Перевагами ПЛІС є невеликий період від початку процесу проектування пристроїв до випуску серійної продукції. Використання ПЛІС дозволяє випускати високотехнологічні електронні вироби, легше вирішуються завдання узгодження інтерфейсів мікросхем за рахунок можливості перепрограмування

ПЛІС. Налагодження та тестування апаратури на ПЛІС спрощується як за рахунок обов'язкової реалізації в них стандарту периферійного сканування JTAG, так і можливості вбудовування в проекти (на стадії їх налагодження) схем логічних аналізаторів. ПЛІС особливо ефективні при реалізації принципу масового паралелізму, наприклад, у пристроях цифрової обробки сигналів. Крім того, ПЛІС є зручним засобом для створення та дослідження прототипів та макетів пристроїв перед їх реалізацією у вигляді ВІС.

Платформи ПЛІС Xilinx широко застосовуються у різних галузях промисловості, виробництва, а також у системах та мережах, включаючи серверні платформи хмарних обчислень, SDN/NFV, відео обробки, машинного зору, робототехніки та багатьох інших областях.

Компанія Xilinx постачає розвинені засоби проектування та налагодження проектів, які включають, у тому числі, програмне забезпечення на основі мов опису апаратури з використанням інструментів бібліотек IP ядер.

За допомогою цього унікального поєднання програмних та апаратних засобів Xilinx пропонує інноваційні рішення, що задовольняють сучасним вимогам і попиту індустрії, що стрімко зростає. Все разом це дозволяє збільшувати в десятки разів пропускну здатність пристроїв, при скороченні затримок і потужності, що споживається.

Розробка проекту здійснювалась на технологічній платформі ПЛІС Xilinx Spartan-3E. Синтез проводився з використанням системи синтезу Synplify.

Для синтезу апаратної моделі ірраціонального обчислювача використана платформа ПЛІС Xilinx SPARTAN 3E XC3S500E.

FPGA має типову структуру вентиляційної матриці. ПЛІС типу FPGA фірми Xilinx, що виконані по SRAM КМОП технології, характеризуються високою гнучкістю структури. При цьому логіка реалізується за допомогою матриці LUT - таблиць (Look Up Table), а внутрішні міжз'єднання

реалізуються за допомогою розгалуженої ієрархії металевих ліній, комутованих спеціальними швидкодіючими транзисторами.

Застосування готових інструментальних засобів дозволяє значно скоротити сумарний час розробки проектного пристрою, уникнувши можливих помилок, що вносяться при виготовленні друкованої плати і монтажі компонентів. Крім того, налагоджені інструментальні модулі полегшують процес освоєння нових методів та програмних засобів розробки систем на основі ПЛІС, зокрема систем на основі конфігурованих мікропроцесорних ядер.

ПЛІС Xilinx характеризуються високою швидкодією, а також можливістю перепрограмування безпосередньо в самій системі. Забезпечений високий ступінь інтеграції, який дає можливість розмістити цифровий пристрій у єдиному кристалі. Результатом цього знижується час і витрати на трасування, а також виробництво плат.

Основою ПЛІС FPGA є матриця логічних комірок (logic cells), оточених блоками введення-виводу, що безпосередньо підключаються до зовнішніх виводів корпусу ПЛІС. Між окремими комірками, а також між матрицею комірок та блоками введення-виводу знаходяться програмовані трасувальні ресурси. Крім того, сучасні сімейства ПЛІС Xilinx Spartan містять на кристалі досить багато різноманітних спеціалізованих ресурсів, які часто вносять істотні корективи в процес проектування цифрових пристроїв. До таких пристроїв відносяться: виділені трасувальні ресурси, призначені для зменшення затримки розповсюдження сигналів, які потрібні для побудови типових вузлів цифрової схемотехніки, наприклад, ланцюги прискореного перенесення.

Системою наскрізного проектування є програмні засоби WebPACK ISE Система реалізує повний цикл проектування, розробки цифрових пристроїв на платформі ПЛІС.

Система включає різні етапи створення описів проекту, а також здійснення синтезу, моделювання. Далі виконується розміщення, трасування, а також програмування кристала.

Фірма Xilinx забезпечує проектувальників та розробників цифрових пристроїв та компонентів сучасне програмне забезпечення, що використовується для розробки і конфігурації кристалів.

Застосування ПЗ ISE дозволяє розробникам значно скоротити час розробки і підвищити рівень ефективності результатів.

Модуль iMPACT, що входить до складу пакету, може застосовуватися для конфігурації практично всіх кристалів, що випускаються фірмою Xilinx.

Технічні характеристики обраного кристалу приведено нижче у таблиці 3.1.

Таблиця 3.1 - Параметри кристала типу FPGA Spartan 3E серії XC3S500E

Device	Logic Cells	System Gates (Logic and RAM)	CLB Array (C*R)	Total CLBs	Maximum Available User I/O	Total Distributed RAM Bits	Total Block RAM Bits
XC3S500E	10476	500000	34*46	1164	232	73K	360K

Логічний блок (CLB) є базовим логічним елементом FPGA, що конфігурується, утворений з декількох секцій-slice, які в свою чергу складаються зазвичай з двох логічних комірок.

Кожен CLB-блок в серії Spartan-3E містить чотири логічних комірки, які чітко організовані у вигляді двох повністю однакових секторів (Slice). Функціональні генератори, що є у FPGA, реалізовані як 4-х входів функціональні таблиці (LUT).

Програмована користувачем вентилярна матриця Spartan-3E охоплює: конфігуровані логічні блоки (configurable logic blocks - CLBs) і блоки вводу-виводу (IOBs). CLBs блоки служать для створення функціональних логічних елементів, а блоки Input / Output (I/O) створюють інтерфейс між контактами мікросхеми і CLBs блоками.

## 4 АПАРАТНА РЕАЛІЗАЦІЯ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА ІРРАЦІОНАЛЬНИХ ФУНКЦІЙ

В результаті розробленої математичної моделі і архітектури ірраціонального обчислювача було здійснено експериментальне автоматизоване проектування пристрою. Розроблено апаратну реалізацію обчислювача на основі кінцевого автомату, розроблено граф-схему алгоритму реалізації ірраціональної функції, на підставі якої отримано граф переходів керуючого автомату моделі Мура, розроблено автоматні HDL-моделі пристрою, здійснено верифікацію отриманого рішення та імплементацію моделі в ПЛІС Xilinx Spartan 3E.

### 4.1 Результати обчислювального процесу в компонентах пристрою

В процесі експериментального дослідження розроблено апаратну реалізацію обчислювача, що відтворює досліджувану ірраціональну функцію:

$$y = \left[ \frac{1}{p} \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right] + 0,5 \right], \quad (4.1)$$

де  $x_i$  – пакки біт вхідного бітового потоку;

$p$  – ціле позитивне число.

На вхід обчислювача подається бітова послідовність, що представляє собою пакки біт (імпульсів)  $x_i$  – інформаційній сигнал.

На виході пристрою генерується вихідний бітовий потік  $y$ , що відтворює апроксимуючу ірраціональну функцію (4.1.) з похибкою 0,5 одиниці

молодшого біту  $x$  при виконанні операцій добування кореня квадратного і операції ділення на константу  $p$ .

У таблиці 4.1 наведено вихідні дані для проведення експерименту: аргументи  $X_1, X_2, X_3, X_4$ , що задані кількістю бітів у пачці  $x$ , цілочисельного коефіцієнта  $p$ , числа бітових пачок  $n$  виразу відтворюваної функції  $y$ .

Таблиця 4.1 – Вихідні експериментальні дані

$X_1$	$X_2$	$X_3$	$X_4$	$n$	$p$
7	4	5	6	4	2

У таблицях 4.2 і 4.3 приведено вхідні та вихідні параметри та сигнали для проектування досліджуваного пристрою.

Таблиця 4.2 - Вхідні параметри та сигнали

Входи	Розрядність	Опис
Reset	1 біт	Установка в початковий стан пристрою
Clk	1 біт	Сигнал синхронізації
Data	1 біт	Інформаційні дані – вхідний бітовий потік

Таблиця 4.3 - Вихідні параметри та сигнали

Виходи	Розрядність	Опис
Ready	1 біт	Готовність пристрою до прийому нових інформаційних даних
DataOut	1 біт	Вихідний бітовий потік

Розглянемо обчислення функції з біт-потоківими даними.

У відповідності до вихідних даних таблиці 4.1 досліджувана функція має вигляд:

$$y = \left[ \frac{1}{p} \left[ \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2} + 0,5 \right] + 0,5 \right], \quad (4.2)$$

$|\delta_{1\max}| = |\delta_{2\max}| = 0,5$  – абсолютні похибки обчислення при виконанні математичних операцій добування кореня квадратного і операції ділення на константу  $p$ . Константа дорівнює  $p = 2$ .

На вхід обчислювача надходить 4 пачки біт розмірністю  $x_1 = 7$ ,  $x_2 = 4$ ,  $x_3 = 5$ ,  $x_4 = 6$ , в результаті відтворювана функція набуває вигляду:

$$y = \left[ \frac{1}{2} \left[ \sqrt{7^2 + 4^2 + 5^2 + 6^2} + 0,5 \right] + 0,5 \right]. \quad (4.3)$$

Функція (4.3) є експериментальною, розглянуто подачу на вхід пристрою чотирьох пачок бітових послідовностей. На практиці при визначенні середньоквадратичних значень кількість пачок буде вищою.

Для того, щоб обчислювати суму квадратів у математичному виразі (4.3), необхідно після обчислення кожної пачки, що поступила на вхід пристрою, приводити суматор SUM2 в модулі степеневого обчислювача проміжної функції в initial state (начальний стан) перед подачею на вхід пристрою наступної пачки бітового потоку [14].

1. Обчислення заданої функції. Результати обчислення ірраціональної функції наведено для чотирьох пачок бітових послідовностей.

У таблиці 4.4 приведено результати обчислення заданої функції (4.3) для першої пачки вхідної бітової послідовності  $x_1 = 7$ , при підстановці у формулу

Таблиця 4.4. Результати обчислення ірраціональної функції для першої пачки біт вхідного бітового потоку

№ пачки	Значення аргументу X	Значення функції Y
1 пачка	1	$y = [\frac{1}{2}[\sqrt{1^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [1.5] + 0.5] = 1$
	2	$y = [\frac{1}{2}[\sqrt{2^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [2.5] + 0.5] = 1$
	3	$y = [\frac{1}{2}[\sqrt{3^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [3.5] + 0.5] = 2$
	4	$y = [\frac{1}{2}[\sqrt{4^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [4.5] + 0.5] = 2$
	5	$y = [\frac{1}{2}[\sqrt{5^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [5.5] + 0.5] = 3$
	6	$y = [\frac{1}{2}[\sqrt{6^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [6.5] + 0.5] = 3$
	7	$y = [\frac{1}{2}[\sqrt{7^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [7.5] + 0.5] = 4$

У таблиці 4.5 приведено результати обчислення функції для другої пачки вхідної бітової послідовності .

Таблиця 4.5. Результати обчислення ірраціональної функції для другої пачки біт вхідного бітового потоку

№ пачки	Значення аргументу X	Значення функції Y
2 пачка	1	$y = [\frac{1}{2}[\sqrt{7^2 + 1^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [7.5] + 0.5] = 4$
	2	$y = [\frac{1}{2}[\sqrt{7^2 + 2^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [7.7] + 0.5] = 4$
	3	$y = [\frac{1}{2}[\sqrt{7^2 + 3^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [8.1] + 0.5] = 4$
	4	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [8.5] + 0.5] = 4$

У таблиці 4.6 приведено результати обчислення функції для третьої пачки вхідної бітової послідовності.

Таблиця 4.6. Результати обчислення ірраціональної функції для третьої пачки біт вхідного бітового потоку

№ пачки	Значення аргументу X	Значення функції Y
3 пачка	1	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 1^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [8.6] + 0.5] = 4$
	2	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 2^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [8.8] + 0.5] = 4$
	3	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 3^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [9.1] + 0.5] = 5$
	4	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 4^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [9.5] + 0.5] = 5$
	5	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [9.9] + 0.5] = 5$

У таблиці 4.7 приведено результати обчислення функції для четвертої пачки вхідної бітової послідовності.

Таблиця 4.7. Результати обчислення ірраціональної функції для четвертої пачки біт вхідного бітового потоку

№ пачки	Значення аргументу X	Значення функції Y
4 пачка	1	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2 + 1^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [10] + 0.5] = 5$
	2	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2 + 2^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [10.2] + 0.5] = 5$
	3	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2 + 3^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [10.4] + 0.5] = 5$
	4	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2 + 4^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [10.8] + 0.5] = 5$
	5	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2 + 5^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [11.2] + 0.5] = 6$
	6	$y = [\frac{1}{2}[\sqrt{7^2 + 4^2 + 5^2 + 6^2} + 0.5] + 0.5] = [\frac{1}{2} \cdot [11.7] + 0.5] = 6$

2. Алгоритм обчислення поліномів. Нерівність, яка реалізується в

пристрої має вигляд

$$2^2 x_y^2 \geq (2y - 1)^2. \quad (4.4)$$

Підставляючи в нерівність значення  $x$  вхідного бітового потоку максимальної довжини з чотирьох пачок, тобто  $x = 1, 2, 3, \dots, 7$ , та  $y = 1, 2, 3, \dots, 7$ , у праву частину нерівності (4.4) отримано арифметичні ряди 2-го порядку.

Для функції лівої частини нерівності (4.4)  $P = x_y^2$  арифметичний ряд 2-го порядку:

$$P: \quad 1, 4, 9, 16, 25, 36, 49.$$

Арифметичні ряди різниць першого і другого порядків відповідно:

$$\Delta: \quad 3, 5, 7, 9, 11, 13, 15, 17, 19$$

$$\Delta^2: \quad 2, 2, 2, 2, 2, 2, 2, 2.$$

Підставляючи у функцію правої частини нерівності (4.4)  $Q = (2y - 1)^2$  значення  $y = 1, 2, 3, \dots, 7$ , отримано послідовність значень функції  $u$ , яка є арифметичним рядом 2-го порядку:

$$Q: \quad 1, 9, 25, 49, 81, 121, 169.$$

Арифметичні ряди різниць першого і другого порядків для цієї послідовності:

$$\Delta: \quad 8, 16, 24, 32, 40, 48,$$

$$\Delta^2: \quad 8, 8, 8, 8, 8.$$

3. Ініціалізація компонентів пристрою. В архітектурі компоненти мають бути ініціалізовані першими членами арифметичних рядів та їх різниць. Числа ініціалізації компонентів архітектури наступні: SM\_int ініціалізується числом  $2^i - 1$ , SM1:  $1^2 = 1$ , SM2:  $3^2 - 1^2 = 8$ ; RG1:  $2!$ , тобто 2; RG2: константою, числом 8; SM\_RES:  $2^i - p$ , тобто  $2^i - 2$ ; RG3:  $2^i - 2p$ , тобто  $2^i - 4$ .

4. Обчислювальний процес в компонентах пристрою. У таблиці 4.8 приведено

обчислювальний процес у компонентах.

Таблиця 4.8 – Обчислювальний процес у компонентах пристрою

№ пачки	x	SM_int	Біт на виході SM_int	SM1	SM2	SM_RES	Біт на виході SM_RES	CNT
1 Пачка	1	-1+4=3	1	1+2=3	8+8=16	-2+2=0	1	1
		3-8=-5				0-4=-4		
	2	-5+12=7	1	3+2=5	16+8=24	-4+2=-2		
		7-16=-9						
	3	-9+20=11	1	5+2=7	24+8=32	-2+2=0	1	2
		11-24=-13				0-4=-4		
	4	-13+28=15	1	7+2=9	32+8=40	-4+2=-2		
		15-32=-17						
	5	-17+36=19	1	9+2=11	40+8=48	-2+2=0	1	3
		19-40=-21				0-4=-4		
	6	-21+44=23	1	11+2=13	48+8=56	-4+2=-2		
		23-48=-25						
	7	-25+52=27	1	13+2=15	56+8=64	-2+2=0	1	4
		27-56=-29				0-4=-4		
2 Пачка	1	-29+4=-25		1+2=3	64	-4		
	2	-25+12=-13		3+2=5	64	-4		
	3	-13+20=7	1	5+2=7	64+8=72	-4+2=-2		
		7-64=-57						
4	-57+28=-29		7+2=9	72	-2			
3 Пачка	1	-29+4=-25		1+2=3	72	-2		
	2	-25+12=-13		3+2=5	72	-2		
	3	-13+20=7	1	5+2=7	72+8=80	-2+2=0	1	5
		7-72=-65				0-4=-4		
	4	-65+28=-37		7+2=9	80	-4		
5	-37+36=-1		9+2=11	80	-4			
4 Пачка	1	-1+4=3	1	1+2=3	80+8=88	-4+2=-2		
		3-80=-77						
	2	-77+12=-65		3+2=5	88	-2		
	3	-65+20=-45		5+2=7	88	-2		
	4	-45+28=-17		7+2=9	88	-2		
	5	-17+36=19	1	9+2=11	88+8=96	-2+2=0	1	6
19-88=-69		0-4=-4						
6	-69+44=-25		11+2=13	96	-4			

При виконанні обчислень в компонентах пристрою враховано перенесення чисел. Перенесення числа з суматора SUM1 в суматор SUM2

виконується зі зсувом на 2 розряди вправо, що відповідає операції множення на число  $2^2$  в нерівності (4.4).

Результати розрахунків обчислювального процесу в компонентах пристрою співпадають з результатами розрахунків значення ірраціональної функції для кожної пачки бітового потоку та по всім вхідним бітам.

#### 4.2 Граф-схема алгоритму та граф переходів керуючого автомату обчислювача

При розробці апаратної реалізації біт-потокowego обчислювача ірраціональних функцій апаратну модель пристрою було представлено на основі кінцевого автомату. Такий підхід до проектування пристрою дозволив забезпечити чіткість та несуперечливість алгоритму обчислення ірраціональної функції.

Кінцевий автомат складається з керуючого та операційного автоматів. ГСА операційного автомату реалізації ірраціональних функцій побудовано на основі розробленої математичної моделі обчислювача і містить операторні вершини з мікроопераціями, що відповідають конвеєрним обчисленням в компонентах пристрою.

Досліджуваний обчислювач працює за наступним алгоритмом (рис. 4.1).

1. При скиданні пристрою по сигналу «reset» регістри встановлюються в значення ініціалізації компонентів пристрою.

2. ГСА містить умовну вершину «impulse». При надходженні чергового біта на вхід пристрою, виконуються операції:

- значення регістра  $SM\_int$  збільшується на значення регістра суматора  $SM1$ ;
- значення регістра суматора  $SM1$  збільшується на значення регістра  $RG1$ .

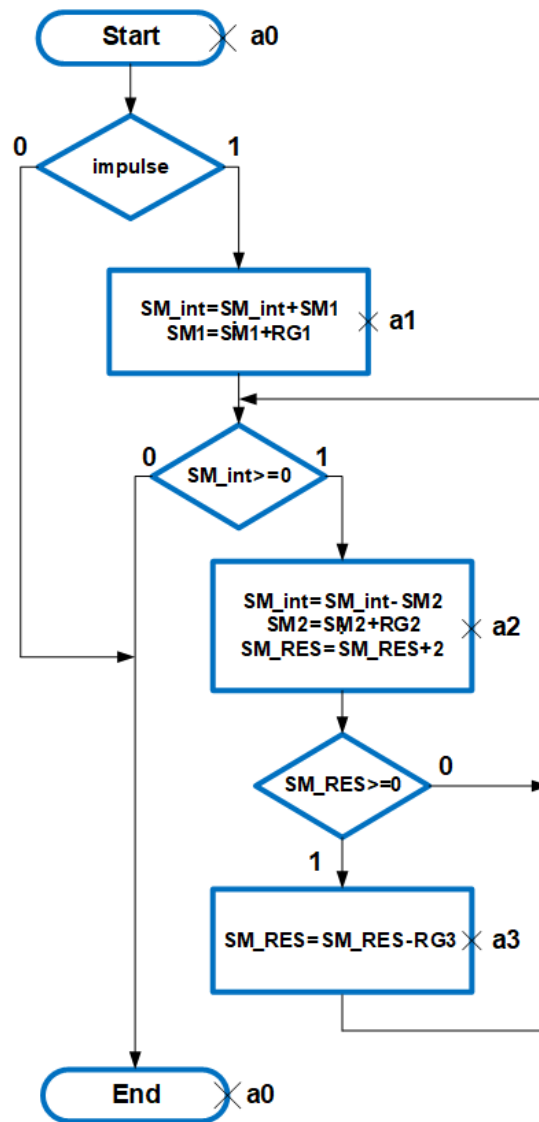


Рисунок 4.1 – Змістовна граф-схема алгоритму роботи пристрою

3. Якщо значення регістра  $SM\_int \geq 0$  виявляється невід’ємне, тоді на виході суматора  $SM\_int$  степеневого модуля з’явиться вихідний біт – результат обчислення проміжної степеневі функції  $y_{int} := f(x)$ . При цьому від значення регістра  $SM\_int$  віднімається значення регістра  $SM2$ , так як значення регістра суматора  $SM2$  переноситься в регістр суматора  $SM\_int$  у додатковому коді. При цьому паралельно з деякою затримкою значення регістра суматора  $SM2$  збільшується на значення регістра  $RG2$ . До значення регістру суматора  $SM\_RES$  додається число 2 у прямому двійковому коді.

Оскільки кожний вхідний біт з виходу  $SM\_int$  надходить у суматор  $SM\_RES$  зі зсувом на один розряд вправо.

4. Якщо при обчисленнях вміст регістра  $SM\_RES \geq 0$  стає невід'ємним, тоді на виході суматора  $SM\_RES$  формується вихідний біт у, що є вихідним бітом обчислювача ірраціональних функцій. За допомогою вихідного біта по ланцюгу зворотного зв'язку до регістра суматора  $SM\_RES$  додається значення регістра  $RG3$ , тобто значення у додатковому коді  $2^k - 2p$ .

При створенні апаратної моделі обчислювача був розроблений керуючий автомат. Було запропоновано використовувати автоматну модель Мура та розроблено граф переходів керуючого автомату Мура біт-потокowego обчислювача.

Керуючий автомат обчислювача, що описується графом переходів (рис.4.2), був отриманий в результаті розмітки ГСА арифметичного блоку обчислювача.

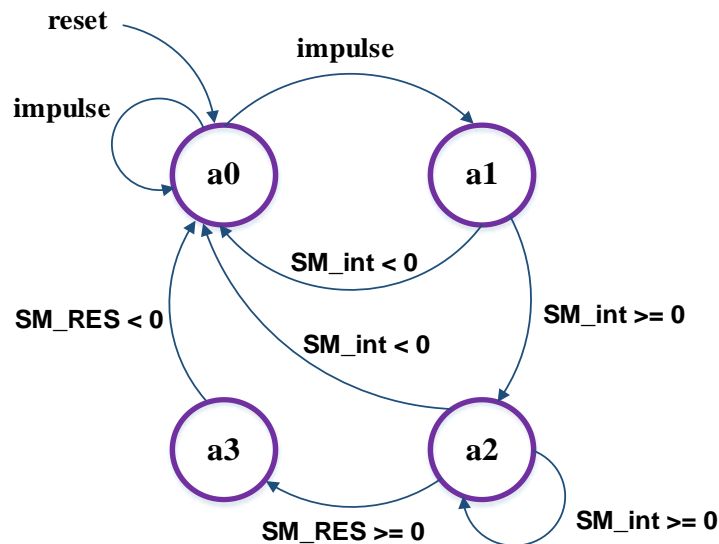


Рисунок 4.2 –Граф переходів керуючого автомату пристрою

На основі ГСА операційного автомату реалізації ірраціональної функції розроблено граф переходів керуючого автомата обчислювача за моделлю Мура, що представлений на рисунку 4.7.

Граф переходів автомата має чотири стани:  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_3$ .

1. У стані  $a_1$  значення  $SM\_int$  збільшується на значення регістра  $SM_1$  зі зсувом на 2 розряди вправо, що відповідає множенню на  $2^2$  вмісту регістра  $SM\_int$ , а значення регістра  $SM_1$  збільшується на 2, так як значення  $RG_1 = 2$ .

2. Автомат переходить у стан  $a_2$ , якщо значення регістра  $SM\_int$  стає невід'ємним, на виході  $SUM_1$  буде згенерований вихідний біт.

У стані  $a_2$  від значення регістра  $SM\_int$  віднімається значення регістра  $SM_2$ , так як значення регістра  $SM_2$  переноситься в регістр  $SM\_int$  в додатковому коді, значення  $SM_2$  збільшується на 8, так як значення регістра  $RG_2 = 8$ , і до значення регістру  $SM\_RES$  додається 2.

3. Автомат переходить у стан  $a_3$ , якщо зміст регістра  $SM\_RES$  стає невід'ємним, тоді на виході  $SM\_RES$  буде згенерований вихідний біт.

4. У стані  $a_3$  в від значення регістра  $SM\_RES$  віднімається значення регістра  $RG_3$ , яке дорівнює  $-2^p$ . Також в цьому стані формується біт на виході пристрою, що є вихідним бітом ірраціонального обчислювача і результатом обчислення заданої функції.

#### 4.3 Структурно-блокова схема пристрою

При виконанні апаратної реалізації було розроблено структурно-блокову схему досліджуваного пристрою.

Пристрій містить два блоки (рис.4.3) :

- детектор імпульсу,
- блок біт-потокowego обчислювача.

Блок детектора імпульсу призначений для детектування біт вхідного потоку  $x$  і на виході встановлює відповідний сигнал  $impulse = 1$ , який буде отриманий арифметичним блоком обчислювача для подальшої обробки [15].

Блок біт-потокowego обчислювача містить арифметичний блок  $Spressalc$ , який виконує операції зведення аргументу  $x$  в дробовий ступінь із заданою абсолютною похибкою обчислення та ділення результату на константу.

Арифметичний блок видає Сигнал Ready, який означає, що блок готовий прийняти наступний біт на обробку.

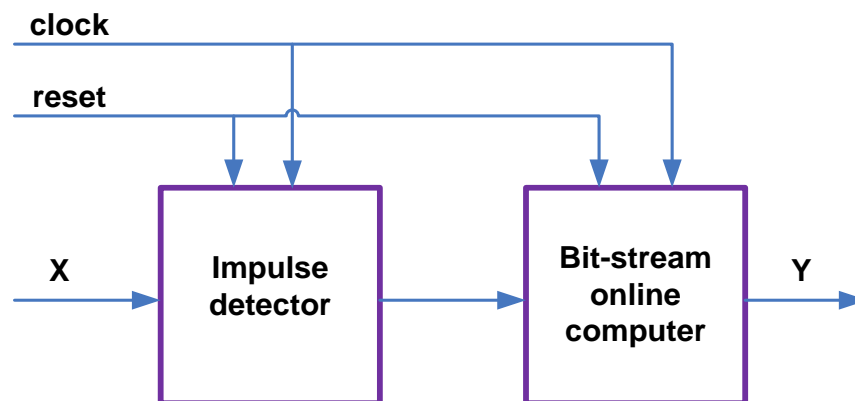


Рисунок 4.3 – Структурно-блокова схема пристрою

Результатом роботи блоку є вихідний біт-поточковий сигнал  $y$ , що є результатом обчислення досліджуваної функції.

Часова діаграма детектування вхідного імпульсу показана на рис. 4.4.

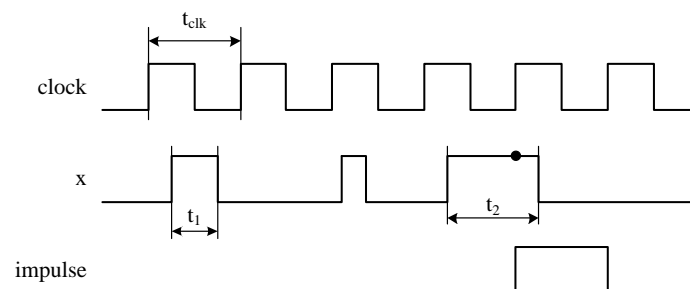


Рисунок 4.4 – Часова діаграма детектування вхідного імпульсу

Вхідний імпульс бітової послідовності детектується за двома сусідніми тактами сигналу  $clock$ . Якщо на черговому такті значення сигналу  $x = 0$ , а на наступному такті значення  $x = 1$ , то вхідний буфер детектує імпульс і на виході встановлює відповідний сигнал  $impulse = 1$ . Цей сигнал буде отримано пристроєм обчислювача для подальшої обробки. Таким чином видно, що для

детектування вхідного імпульсу потрібно, щоб період сигналу clock  $t_{clk}$  був менше або дорівнював періоду вхідної імпульсної (бітової) послідовності. Якщо період вхідної імпульсної послідовності  $t_1$  менше періоду синхронізації clock, то вхідний буфер не зможе детектувати такі імпульси ( $t_1 < t_{clk}$ ).

Коли сигнал ready = 0, арифметичний блок не готовий приймати імпульси (ще не закінчена обробка попередніх імпульсів).

Вхідний буфер накопичує кількість імпульсів в регістрі counter.

Коли сигнал ready = 1, арифметичний блок готовий прийняти вхідний імпульс.

Опис входів та виходів детектора вхідного імпульсу зазначено в таблиці 4.9.

Таблиця 4.9 - Вхідні і вихідні параметри детектора вхідного імпульсу

Сигнал	Опис	Режим
x_i	Біт-потоківна вхідна послідовність	IN
reset_i	Установка в початковий стан пристрою	IN
clock_i	Сигнал синхронізації	IN
y_o	Біт-потоківна вихідна послідовність	OUT
ready_i	Сигнал готовності з арифметичного блоку	IN

Блок «Спеціалізований обчислювач» містить блок «Spssalc». Вхідні та вихідні параметри для даного блоку наведено у таблиці 4.10.

Блок «Spssalc» виконує операції обчислення суми квадратів аргументу, що представлений пачками біт потоку x, добуває квадратний корінь з похибкою  $|\delta_{max1}|=0,5$ , а також ділить отриманий результат на число p з похибкою  $|\delta_{max2}|=0,5$ , тобто результат добування кореня та ділення округляється до найближчих цілих чисел.

На виході біт-потоківного обчислювача формується вихідна бітова послідовність y після появи сигналу Ready на виході арифметичного блоку,

що дорівнює '1'. Даний блок видає сигнал готовності для прийняття обчислювачем наступного вхідного біту на обробку.

Результатом роботи обчислювача є сигнал DataOut – вихідний сигнал, який являє собою результат обчислення заданої ірраціональної функції. Інтерфейс арифметичного блоку на мові VHDL наведено у лістингу 4.1.

Таблиця 4.10 - Вхідні та вихідні параметри для блоку «Speccalc»

Data	Біт-потокова вхідна послідовність	IN
Reset	Установка в початковий стан пристрою	IN
Clock	Сигнал синхронізації	IN
DataOut	Біт-потоківий вихідний сигнал, який є результатом обчислення ірраціональної функції $y = \left[ \frac{1}{p} \left[ \sqrt{\sum_{i=1}^n x_i^2} + 0,5 \right] + 0,5 \right]$	OUT
Count	Значення функції в двійковому 16-розрядному коді на виході SM_RES	OUT
Ready	Сигнал готовності з арифметичного блоку	IN
Get_next_bundle	Сигнал готовності прийняти наступну пачку імпульсів	OUT
Next_bundle	Сигнал початку формування пачки	IN
Done	Сигнал закінчення формування пачки імпульсів	IN

Лістинг 4.1 - Інтерфейс арифметичного блоку обчислювача «Speccalc»

```
entity speccalc is
  generic(
    p: natural := 3;
    a: natural := 2;
    width: natural := 16);
  port(
    x_i: in std_logic;
```

```

x_imp_o: out std_logic;
ready_o: out std_logic;
done_i: in std_logic;
get_next_bundle: out std_logic;
next_bundle: in std_logic;
clock_i: in std_logic;
reset_i: in std_logic;
y_o: out std_logic;
sum_o: out std_logic_vector(width-1 downto 0));
end entity speccalc;

```

#### 4.4 Опис проекту на МОА

Як ефективне середовище моделювання та верифікації проектів на платформі FPGA обрано програмний пакет Active-HDL, що дозволяє автоматизувати процес введення розробленого проекту в САПР, а також ефективно здійснювати аналіз проектних рішень. Апаратна модель пристрою розроблена мовою VHDL із використанням автоматної моделі опису.

Для синтезу проекту вибрано програмний продукт фірми Synplify Synplify 7.0.2, так як він має певні переваги, зокрема, високу швидкість синтезу, візуальне представлення результатів синтезу, наявність бібліотек, що складають сучасну елементну базу. Введення проекту за допомогою опису на мові опису апаратури (МОА) є гнучким варіантом, незважаючи на те, що потребує більше знань специфіки застосованої МОА.

В якості МОА для опису автоматних моделей пристрою обрано мову VHDL. Такий вибір обумовлений високою адаптацією засобів мови для потреб проектувальника. Крім названої переваги, є інші переваги – це наявність ефективного середовища моделювання Active-HDL та ефективного середовища синтезу, яке зручно використовує VHDL (Synplify). Проект моделі пристрою представляє собою складний проект у вигляді ієрархічної структури, все більш деталізуючи його опис на нижчих рівнях. Тим самим спрощується процес написання VHDL-моделей окремих компонентів проекту, а також їх налагодження [16].

Файлова структура проекту містить: `speccalc_top.vhd`, `speccalc.vhd`, `speccalc_oa.vhd`, `speccalc_ua.vhd`, `speccalc_imp.vhd`, `speccalc_tb.v`, `config_packages.vhd`, `traceability_unit.vhd`.

Весь проект складається з множини файлів, які об'єднані за допомогою одного `Speccalc.top.vhdl`, він є верхньою частиною ієрархії проекту. Даний файл складається з компонентів, які об'єднані у файл за допомогою операторів `port map`, що є сполучними файлами нижньої ієрархії. Вищевказаний файл є реалізацією на мові опису апаратури специфікації всього пристрою, він складається з `entity`, в якому описані всі вхідні і вихідні сигнали пристрою, задекларовані компоненти нижнього рівня компонент детектору імпульсів `Speccalc_imp.vhdl` та компонент обчислювача `Speccalc.vhdl` (Лістинг 4.4).

Лістинг 4.4 – Фрагмент коду, що описує компоненти `Speccalc_imp.vhdl`, `Speccalc.vhdl`.

```
architecture struct of speccalc_toplevel is
  -- Component declaration of the "aproximator(struct)" unit defined in
  -- file: "./src/aproximator.vhd"
  component speccalc
    port(
      x_i: in std_logic;
      x_imp_o: out std_logic;
      ready_o: out std_logic;
      done_i: in std_logic;
      get_next_bundle: out std_logic;
      next_bundle: in std_logic;
      clock_i: in std_logic;
      reset_i: in std_logic;
      y_o: out std_logic;
      sum_o: out std_logic_vector(width-1 downto 0));
  end component;

  -- Component declaration of the "inputbuffer(beh)" unit defined in
  -- file: "./src/inputbuffer.vhd"
  component speccalc_imp
    port(
      x_i : in std_logic;
      clock_i : in std_logic;
      reset_i : in std_logic;
      ready_i : in std_logic;
      y_o : out std_logic);
  end component;
```

Фрагмент HDL-коду з'єднання блоків "Speccalc.top.vhdl", приведений у лістингу 4.5.

Лістинг 4.5 – Фрагмент HDL-коду з'єднання блоків "Speccalc.top.vhdl"

```

signal imp_out, speccalc_out, ready : std_logic;
signal count_t: std_logic_vector(width-1 downto 0);
signal sum_t: std_logic_vector(width-1 downto 0);
signal x_imp_o, done, get_next_bundle, next_bundle: std_logic;
begin
  Impulse_Detector: speccalc_imp
  port map(
    x_i => x_i,
    clock_i => clock_i,
    reset_i => reset_i,
    ready_i => ready,
    y_o => imp_out
  );
  spec_calc: speccalc
  port map(
    x_i => imp_out,
    x_imp_o => x_imp_o,
    ready_o => ready,
    done_i => done,
    get_next_bundle => get_next_bundle,
    next_bundle => next_bundle,
    clock_i => clock_i,
    reset_i => reset_i,
    y_o => speccalc_out,
    sum_o => sum_t
  );
  trace_unit: traceability_unit
  port map (
    reset_i => reset_i,
    x_imp_i => x_imp_o,
    done => done,
    get_next_bundle => get_next_bundle,
    next_bundle => next_bundle);

```

У лістингу 4.5 приведено фрагмент HDL-моделі керуючого автомату обчислювача ірраціональних функцій.

Лістинг 4.5 – Фрагмент HDL-моделі керуючого автомату обчислювача

```

process(state, x_i, sum1_above_zero_i, sum2_above_zero_i)
begin
  case (state) is
    when a_0 =>
      if x_i = '1' then

```

```

        next_state <= a_1;
    else
        next_state <= a_0;
    end if;
    when a_1 =>
    if sum1_above_zero_i = '1' then
        next_state <= a_2;
    else
        next_state <= a_0;
    end if;
    when a_2 =>
    if sum2_above_zero_i = '1' then
        next_state <= a_3;
    else
        next_state <= a_0;
    end if;
    when a_3 =>
    next_state <= a_0;
    when others =>
    next_state <= a_0;
end case;
end process;

```

HDL-модель операційного автомату обчислювача створюються для конкретної ірраціональної функції. Нижче наведено фрагмент коду, що описує роботу операційного автомата блоку «Spessalc.oa», в якому описано реалізацію конвеєрних обчислень в компонентах пристрою. Повну HDL-модель операційного автомату online-обчислювача наведено в додатку А.

#### Лістинг 4.6 – Опис операційного автомата Spessalc.oa

```

process (clock_i, reset_i)
begin
    if (reset_i = '1') then
        counter <= CONV_STD_LOGIC_VECTOR(1, width);
        sum1 <= CONV_STD_LOGIC_VECTOR(-1, width);
        sum2 <= CONV_STD_LOGIC_VECTOR(8, width);
        sum3 <= CONV_STD_LOGIC_VECTOR(-p, width);
        get_next_bundle <= '0';
    else
        if (falling_edge(clock_i)) then
            if (x_imp_i = '1') then
                sum1 <= sum1 + CONV_INTEGER(counter(width-3 downto 0)& "00");
                if (next_bundle = '1') then
                    counter <= CONV_STD_LOGIC_VECTOR(1, width);
                    get_next_bundle <= '1';
                else
                    counter <= counter + step_cnt;
                    get_next_bundle <= '0';
                end if;
            end if;
        end if;
    end if;
end process;

```

```

        end if;
    elsif (sum1_ovf_i = '1') then
        sum1 <= sum1 - sum2;
        sum2 <= sum2 + step_sum;
        sum3 <= sum3 + step_cnt;

        elsif (sum3_ovf_i = '1') then
            sum3 <= sum3 - 2*p;
        end if;
    end if;
end process;
end beh;

```

У додатку А наведено HDL-код програми-обгортки Wrapper.VHDL, який необхідний для успішної інтеграції з індикацією на платі.

У лістингу 4.7 наведено HDL-код опису пачок, що відповідає за управління пачками імпульсів, які подаються на вхід обчислювача.

Лістинг 4.7 – Фрагмент HDL-коду опису пачок файлу config\_package.vhdl

```

package cfg_pkg is
    constant width: natural := 16;
    --same as max in "bundles"
    constant max_Xi: natural := 10;
    type t is array (integer range <>) of integer range 1 to max_Xi;
    constant bundle_count: natural := 4; -- i
    constant bundles: t(0 to bundle_count-1) := (7, 4, 5, 6); -- Xi
    constant step_cnt: natural := 2;
    constant step_sum: natural := 8;
    constant p: natural := 2;
end package;

```

#### 4.5 Верифікація поведінкової моделі обчислювача

Верифікація поведінкової моделі досліджуваного обчислювача ірраціональної функції виконувалась з використанням САПР Active-HDL. Для верифікації поведінкової моделі пристрою було розроблено test bench – модуль, що описаний мовою VHDL. Він має примірник тестового модуля та

здійснює подачу тестових сигналів і аналіз тестових реакцій модуля, фрагмент якого наведено у лістингу 4.8.

#### Лістингу 4.8 – Фрагмент модуля test bench

```
entity testbench is
    generic(
        width: natural := 16);
end entity;
architecture testbench_1 of testbench is
    component speccalc_toplevel
    port(reset_i: in std_logic;
        clock_i: in std_logic;
        x_i: in std_logic;
        y_o: out std_logic;
        count: out std_logic_vector(width-1 downto 0));
    end component;
    signal reset_i, clock_i, x_i, y_o : std_logic;
    signal count: std_logic_vector(width-1 downto 0);
    begin
    UUT : speccalc_toplevel
    port map (reset_i=>reset_i, clock_i=>clock_i, x_i=>x_i, y_o=>y_o, count=>count);
    process
    begin
        clock_i<='0', '1' after 5 ns;
        wait for 10 ns;
    end process;
    process
    begin
        reset_i<='1';
        wait for 10ns;
        reset_i<='0';
        wait for 1000ns;
    end process;
    process
    begin
        x_i<='0', '1' after 10 ns; wait for 20 ns;
    end process;
    end testbench_1;
```

Приклад моделювання роботи досліджуваного обчислювача представлений на рис. 4.5 і 4.6. На вхід пристрою було подано 3 пакки по 4, 10 та 8 імпульсів відповідно. В результаті моделювання пристрій згенерував на виході 4 імпульси, що збігаються з результатами обчислювального процесу в компонентах пристрою (таблиця 4.8), що підтверджує правильність його роботи.

Отже, на часовій діаграмі можна бачити, що результати експерименту співпадають з результатами обчислень, наведеними в таблиці 4.8.

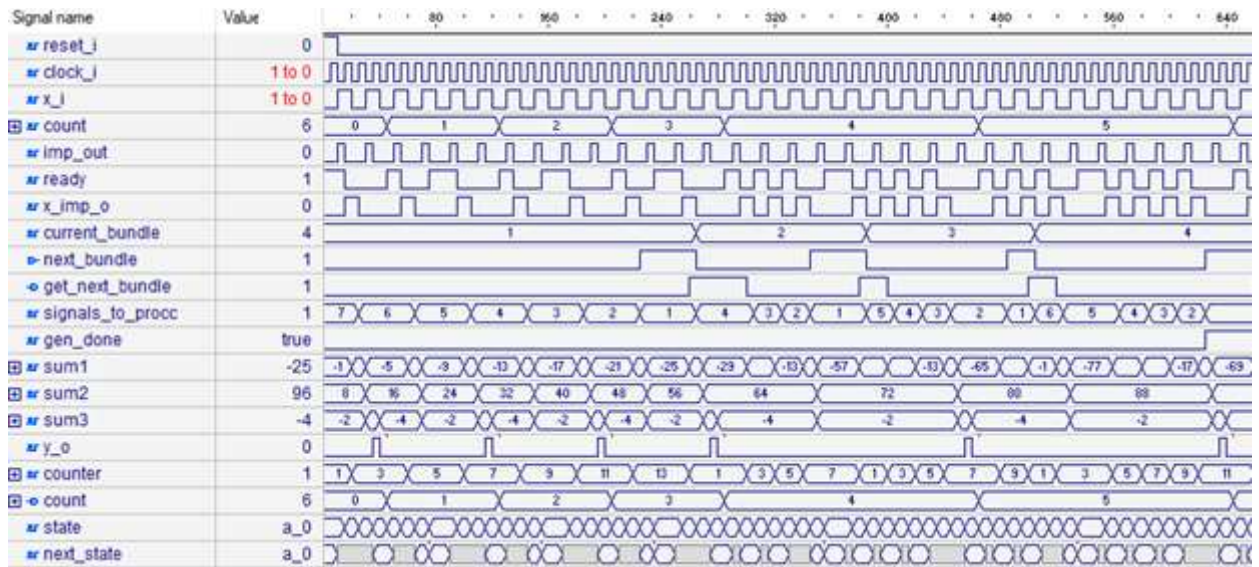


Рисунок 4.5 – Часова діаграма верифікації поведінкової моделі обчислювача

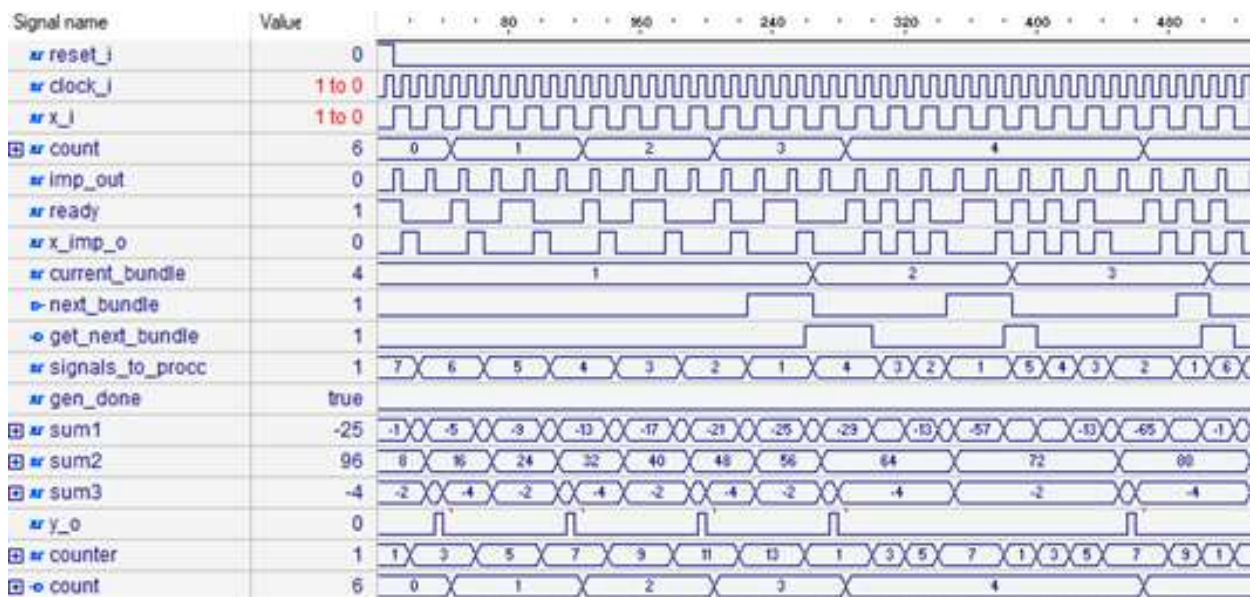


Рисунок 4.6 – Часова діаграма верифікації роботи обчислювача

Перевіривши режими верифікації експериментального обчислювача, і переконавшись, що обчислювач працює коректно, далі було виконано синтез та імплементацію проекту в ПЛІС.

Синтез проводився з використанням системи синтезу Synplify 7.0 Pro. В результаті синтезу було отримано RTL схему вентильного рівня, схему Technology View і файл звіту. Було здійснено імплементацію досліджуваного пристрою в технологічну платформу ПЛІС. Попередньо створений UCF- файл приведено в лістингу 4.9.

#### Лістинг 4.9 – UCF- файл

```
NET "rst" LOC = "H13" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "St" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "LED<7>" LOC = "F9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<6>" LOC = "E9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<5>" LOC = "D11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<4>" LOC = "C11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<3>" LOC = "F11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<2>" LOC = "E11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<1>" LOC = "E12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<0>" LOC = "F12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;

NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
```

Звіт про мінімальний період сигналу синхронізації, його максимальну частоту та використання ресурсів кристалу приведений нжче:

Minimum period: 7.511ns (Maximum Frequency: 133.136MHz)

Number of External IOBs	11 out of 232	4%
Number of External Input IOBs	3	
Number of External Input IBUFs	3	
Number of LOCed External Input IBUFs	3 out of 3	100%
Number of External Output IOBs	8	
Number of External Output IOBs	8	
Number of LOCed External Output IOBs	8 out of 8	100%
Number of External Bidir IOBs	0	
Number of BUFGMUXs	1 out of 24	4%
Number of Slices	46 out of 4656	1%
Number of SLICEMs	0 out of 2328	0%

У звіті показано, що при синтезі пристрою в мікросхему Xilinx Spartan 3E серії xc3s500e було використано приблизно 4% ресурсів, що пов'язані з логічними елементами.

Таким чином, розроблений обчислювач є економічним по відношенню

до займаних ресурсів, обрана мікросхема дозволяє розмістити до 18 екземплярів подібних пристроїв.

На рисунках 4.7 – 4.10 наведено RTL-схеми пристрою, що був імплементований у ПЛІС.

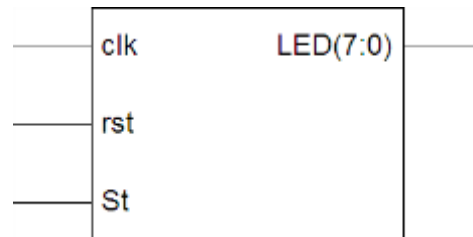


Рисунок 4.7 – RTL-схема інтерфейсу пристрою

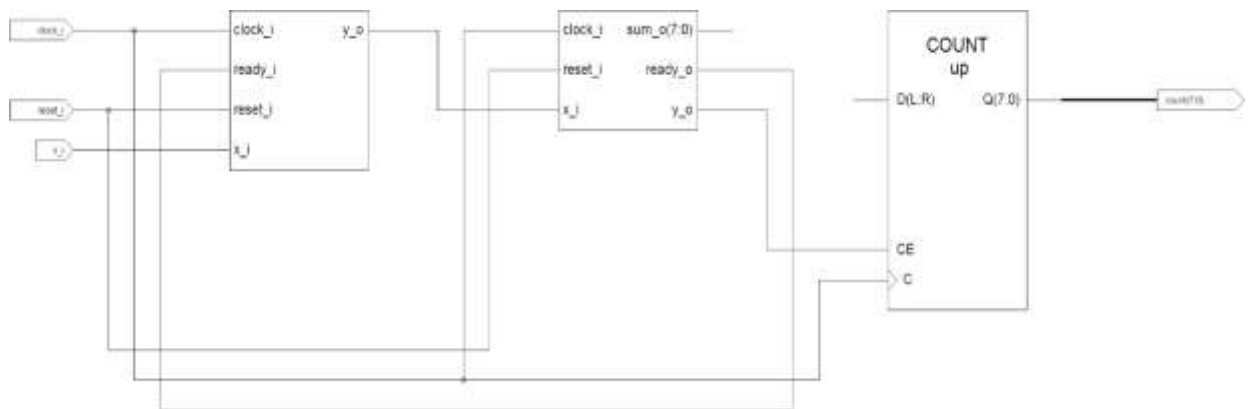


Рисунок 4.8 – RTL схема SpecCalc.top

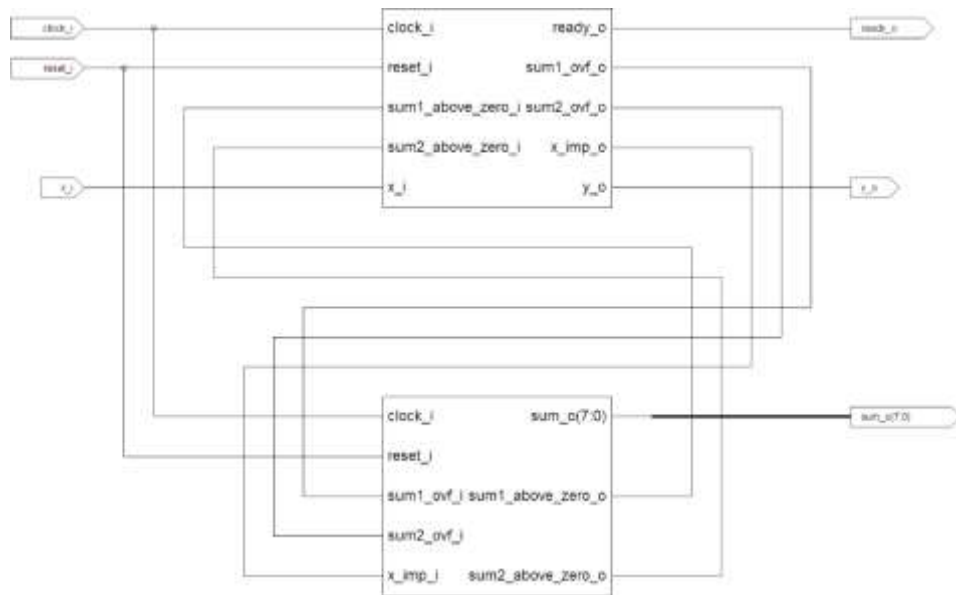


Рисунок 4.9 – RTL схема Spresalc

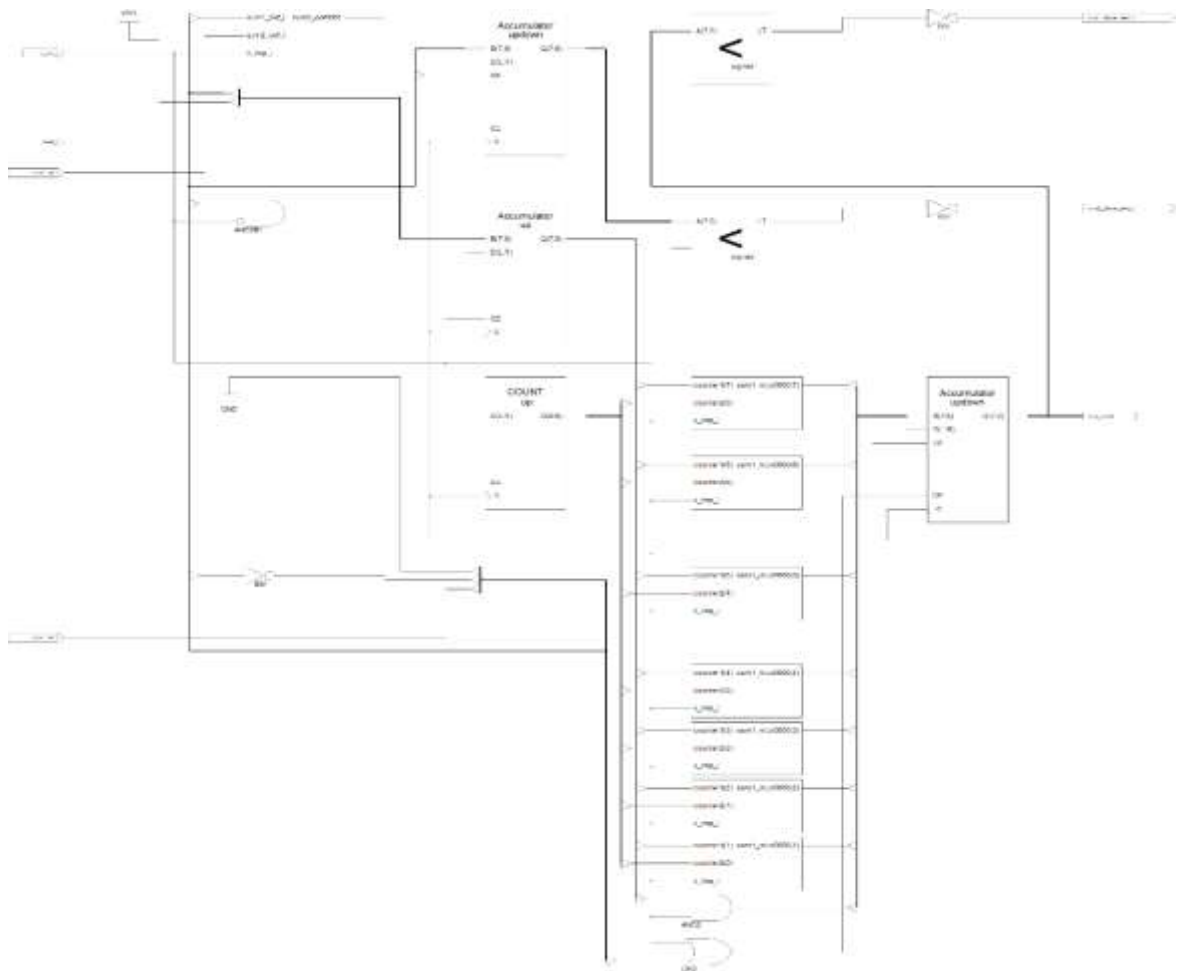


Рисунок 4.10 – RTL схема Spresalc.oa

#### 4.6 Порівняльний аналіз частотних характеристик ірраціональних обчислювачів

При проведенні експериментальних досліджень апаратної реалізації обчислювача ірраціональних функцій при імплементації в ПЛІС було отримано Synthesis reports, в якому зазначено максимальну частоту роботи пристрою. У порівнянні з прототипом, що реалізує ірраціональну функцію на основі дробово-раціонального модуля, досліджуваний ірраціональний обчислювач на основі степеневого модуля має переваги відносно максимальної частоти роботи обчислювача.

На рис. 4.11 приведено порівняльний аналіз частотних характеристик ірраціональних обчислювачів у відповідності до таблиці 4.11.

Таблиця 4.11 Порівняльна таблиця частотних характеристик

Біт-потоківий обчислювач	$T_F$ , ns	$F_{max}$ , MHz
Ірраціональний (1)	7,7 ns	133 MHz
Ірраціональний (2)	10,7 ns	93 MHz

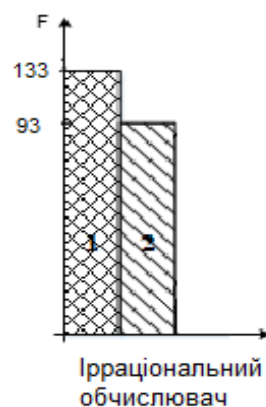


Рисунок 4.11 – Порівняльний аналіз частотних характеристик ірраціональних обчислювачів

Аналізуючи дані таблиці 4.11 можна зробити висновок, що максимальна частота роботи ірраціонального обчислювача (1) на основі степеневого модуля на 28% вища за максимальну частоту роботи ірраціонального обчислювача (2) на основі дробово-раціонального модуля. Це підтверджує теоретичні висновки щодо розширення частотного діапазону ірраціонального обчислювача, побудованого на основі степеневого модуля.

## ВИСНОВКИ

В розподілених системах управління знаходять широке використання біт-потоківі функціональні обчислювачі, що здійснюють лінеаризацію сигналів за певною функцією – функціональне перетворення бітових потоків, які отримують від сенсорів фізичних величин. Бітові потоки представляють собою частотні або час-імпульсні сигнали одиничної амплітуди, що забезпечують завадостійкість при передачі потоків по каналам зв'язку при відсутності надмірності.

Кваліфікаційна робота присвячена розробці та дослідженню моделей та методів проектування біт-потоківі обчислювача ірраціональних функцій на основі технологічної платформи ПЛІС з використанням САПР на основі мов опису апаратури.

Проаналізовано місце та застосування біт-потоківі обчислювачів в системах управління. Проаналізовано метод ступінчастої апроксимації на основі зворотних функцій, що реалізує формування приростів висхідних ступінчастих функцій визначеного класу. На основі методу отримано математичну модель обчислювача досліджуваної ірраціональної функції, що представлена декомпозицією математичних моделей біт-потоківі обчислювачів степеневих та лінійних функцій.

Проаналізовано узагальнену архітектуру обчислювача ірраціональних функцій. Розроблено та описано архітектурну модель обчислювача досліджуваної функції на підставі узагальненої архітектури. Архітектура досліджуваного пристрою синтезована на основі біт-потоківі конвеєрного степеневі обчислювача та дільника чисел. Запропонований обчислювач реалізує апроксимуючу ірраціональну функцію з мінімальною абсолютною похибкою апроксимації, що не перевищує половину одиниці молодшого біту аргументу.

Основним обчислювальним компонентом в структурі обчислювача є паралельний нагромаджуючий суматор зі зворотним зв'язком.

Була виконана експериментальна апаратна реалізація досліджуваної моделі обчислювача, виконані необхідні розрахунки. Ірраціональний обчислювач реалізовано на основі кінцевого автомату моделі Мура. Було розроблено граф-схему алгоритму та граф переходів керуючого автомату пристрою.

Було здійснено опис проекту для введення в САПР. Розроблено графові HDL-моделі пристрою з використанням шаблонів коду. При здійсненні верифікації проекту отримано поведінкову модель пристрою, та шляхом наочної демонстрації прикладу його роботи було проказано, що результати моделювання поведінкової моделі обчислювача ірраціональних функцій співпадають з теоретичними розрахунками. Модель пристрою синтезована засобами САПР Xilinx. В якості платформи для імплементації досліджуваного обчислювача обрано кристал типу FPGA Spartan 3E серії XC3S500E.

Апаратний біт-потоківий обчислювач ірраціональних функцій може бути застосований в системах управління, контролю, інформаційно вимірювальних системах в якості вторинних функціональних перетворювачів бітових потоків, що отримують з датчиків фізичних величин з частотним виходом.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Буренева О.И. Бит-потокное устройство извлечения квадратного корня / О.И. Буренева, О.А. Жирнова // Известия ЛЭТИ № 2, – 2019. С. 26 – 32.
2. A.I. Gulin, N.M. Safyannikov, O.I. Bureneva, A.Yu. Kaydanovich. Assurance of Fault-Tolerance in Bit-Stream Computing Converters // Proceeding of 16th IEEE East-West Design & Test Symposium (EWDTS'2018). Kazan, Russia, September 14 – 17, 2018. – pp. 418 – 421.
3. Dhafer Al-Makhles, Nitish Patel, Akshya Swain. Bitstream control system: Stability and experimental application // Intern. Conf. on Appl. Electronics. Czech Republic, Pilsen, 2013. P. 1–6.
4. Нетеренко Н.В. Применение ПЛИС для построения вычислительных систем. / Н.В. Нетеренко, В.В. Ересько, Ю.С. Яковлев // Математичні машини і системи – 2016. – № 1. – С. 3-14. Електронний ресурс: <http://dspace.nbu.gov.ua/handle/123456789/113577>.
5. Юревич Е. И. Теория автоматического управления. / Е. И. Юревич. – 4-е изд. – Санкт-Петербург: БХВ-Петербург. – 2016. – 560 с.
6. Al-Makhles D. Conventional and hybrid bit-stream in real-time system. / D. Al-Makhles, N. Patel, A. Swain // Proceedings of the 11th Workshop on Intelligent Solutions in Embedded Systems (WISES), Pilsen, Czech Republic. – 2013. –pp.1-6.
7. Стахів М.Ю. Автореф. дисертації. Цифрові функціональні перетворювачі розгортуючого типу з покращеними характеристиками // Поліграф. Центр Видавництва Національного університету "Львівська політехніка" – 2013. – 22 С.
8. Буренева О.И., Жирнова О.А. Многофункциональный бит-потокный преобразователь // Известия ЛЭТИ, 2019, №10,. С. 46 – 53.

9. Сафьянников Н.М., Буренева О.И. Следящий потоковый вычислительный преобразователь для интеллектуальных измерительных систем // Международная конференция по мягким вычислениям. 2019. Т.1. С 263-266.

10. Ларченко Л.В. Функціональне перетворення імпульсних потоків в апаратних обчислювачах математичних функцій / Л.В. Ларченко, Е.М. Кулак, Б.Д. Ларченко // Радіоелектроніка та інформатика. №3. – 2019. – С 27-34.

11. Ларченко Б.Д. Декомпозиція математичної моделі біт-потокowego обчислювача ірраціональних функцій / Б.Д. Ларченко // Радіоелектроніка та інформатика. №4. – 2019. – С 34-39.

12 Ларченко Б.Д. Апаратний біт-потокowy online обчислювач дробово-раціональних функцій. [Текст] / О.С. Шкіль, Б.Д. Ларченко, Л.В. Ларченко, І.В. Філіппенко, С.В. Ющенко // Радіоелектроніка та інформатика. – 2020. – № 3. – С. 55-63. 13.

13. Larchenko Bogdan. Mathematical model of bit-stream online computer of irrational functions. / Larchenko Bogdan, Kuznichenko Taras. // Innovation in science and technology. 15th International scientific and practical conference, January 25-26, Boston, USA, 2021.– P. 82-86.

14. Ларченко Б.Д. Математична модель біт-потокowego online-обчислювача лінійних функцій. / Ларченко Б.Д., Кузніченко Т.Б. // Лютневі наукові читання, LX Міжнародна науково-практична інтернет-конференція. – м. Київ, 01 лютого, 2021. С. 116-121.

15. Larchenko B.D. Bit-Stream Power Function Online Computer / A.S. Shkil, L.V. Larchenko, B.D. Larchenko. // 18 IEEE East-West Design & Test Symposium (EWDTS'2020), Sept. 4-7, Varna, Bulgaria, 2020. – P. 423-428. (Входить до міжнародних наукометричних баз Scopus, IEEE Xplore).

16. Баранов В. А. Анализ и синтез последовательностных устройств: учебное пособие / В. А. Баранов. – Екатеринбург: УрГУПС. – 2016. – 462 с.