

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 дробово-раціональних функцій

(тема)

Виконав:

студент II курсу, групи СКСм-20-1  
Ющенко С.В.  
(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 Чумаченко С.В.  
(підпис) (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 Ющенку Сергію Валерійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Моделі та методи проектування апаратного біт-потокowego обчислювача дробово-раціональних 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 Аналіз предметної області та постановка завдання

2 Математична модель обчислювача дробово-раціональних функцій

3 Архітектура обчислювача дробово-раціональних функцій

4 Апаратна реалізація обчислювача на основі цифрового автомату

5 Розробка автоматних HDL-моделей 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 обчислювача дробово-раціональних функцій	02.10.2021 - 15.10.2021	
5	Синтез архітектури обчислювача дробово-раціональних функцій	16.10.2021 - 30.10.2021	
6	Розробка апаратної моделі обчислювача 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, 9 таблиць, 14 джерел за переліком посилань.

БІТОВИЙ ПОТІК ДАНИХ, БІТ-ПОТОКОВИЙ ОБЧИСЛЮВАЧ, ФУНКЦІОНАЛЬНЕ ПЕРЕТВОРЕННЯ, ДРОБОВО-РАЦІОНАЛЬНИЙ МОДУЛЬ, АПРОКСИМАЦІЯ, АЛГОРИТМ, VHDL-МОДЕЛЬ, ВЕРИФІКАЦІЯ

В кваліфікаційній роботі досліджено та розроблено апаратний біт-потоківий обчислювач дробово-раціональних функцій. Об'єктом дослідження є обчислювачі дробово-раціональних функцій з бітовими потоками даних.

Досліджено математичний апарат на основі ступінчастої апроксимації, що покладений в основу роботи біт-потоківих обчислювачів визначеного класу, розроблено математичну модель пристрою відтворення заданої функції на основі системи різницевих нерівностей. Розроблено архітектуру обчислювача, що є синтезом біт-потоківого обчислювача поліноміальних функцій та пристрою обчислення лінійних функцій з використанням конвеєрної структури цифрових пристроїв. Апаратну реалізацію моделі обчислювача здійснено на основі кінцевого автомату моделі Мура. На основі графових моделей пристрою розроблено VHDL-описи, що моделюються та синтезуються інструментальними засобами САПР цифрових пристроїв XILINX ISE на ПЛІС. Здійснена верифікація проекту та імплементація пристрою в ПЛІС Xilinx Spartan.

## ABSTRACT

The explanatory note contains 77 pages, 17 figures, 9 tables, 14 sources by the list of references.

FUNCTIONAL CONVERSION, BITSTREAM DATA, BITSTREAM CALCULATOR, FRACTIONAL-RATIONAL MODULE, APPROXIMATION, ALGORITHM, VHDL-MODEL, VERIFICATION

In the qualification work a hardware bit-stream computer of fractional-rational functions was researched and developed. The object of the study are computers of fractional-rational functions with data bit-streams.

The mathematical apparatus on the basis of step approximation, which is the basis of bit-stream computers of a certain class, was researched, a mathematical model of the set function reproduction device on the basis of system of difference inequalities was developed. A computer architecture was developed, which is a synthesis of a bit-stream computer of polynomial functions and a device for calculating linear functions using the pipeline structure of digital devices. The hardware implementation of the computer model is based on a Moore finite state machine. Based on the graph model of the device, VHDL-descriptions were developed, which are modeled and synthesized by CAD tools of XILINX ISE digital devices on FPGA. Verification of the project and implementation of the device in FPGA Xilinx Spartan was done.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	11
1.1 Особливості перетворення бітових потоків даних при функціональній обробці сигналів.....	11
1.2 Застосування біт-потоків дробово-раціональних обчислювачів в розподілених системах управління.....	15
1.3 Постановка завдання.....	20
2 МАТЕМАТИЧНА МОДЕЛЬ БІТ-ПОТОВОГО ОБЧИСЛЮВАЧА ДРОБОВО-РАЦІОНАЛЬНИХ ФУНКЦІЙ.....	21
2.1 Математичне обґрунтування способу обчислення дробово-раціональних функцій.....	21
2.2 Математична модель біт-потоків обчислювача досліджуваної функції.....	26
2.3 Алгоритм обчислення поліноміальних функцій .....	31
3 АРХІТЕКТУРА БІТ-ПОТОВОГО ОБЧИСЛЮВАЧА ДРОБОВО-РАЦІОНАЛЬНИХ ФУНКЦІЙ.....	34
3.1 Узагальнена архітектура біт-потоків обчислювачів.....	34
3.2 Загальна архітектура дробово-раціонального обчислювача.....	36
3.3 Синтез архітектури досліджуваного обчислювача дробово-раціональних функцій .....	38
3.4 Вибір елементної бази для апаратної реалізації обчислювача.....	43
4 ЕКСПЕРИМЕНТАЛЬНА АПАРАТНА РЕАЛІЗАЦІЯ ДОСЛІДЖУВАНОВОГО ОБЧИСЛЮВАЧА.....	47
4.1 Етапи розробки експериментальної апаратної моделі обчислювача.....	47

4.2 Теоретичні розрахунки в компонентах пристрою.....	48
4.3 Граф-схема алгоритму та граф переходів керуючого автомату пристрою.....	54
4.4 Структурно-блокова схема пристрою.....	58
4.5. Опис проекту на МОА.....	62
4.6 Верифікація проекту.....	68
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

ІВС – інтелектуально-вимірювальна система

ІКС – інформаційно-керуюча система

ІОС – інформаційно-обчислювальна система

ІС – інтегральна схема

КМОП – комплементарна структура метал-оксид-напівпровідник

МОА – мова опису апаратури

НВІС – надвелика інтегральна схема

ПЛІС – програмована логічна інтегральна схема

САПР – система автоматизації проектування

ТЗ – технічне завдання

ЦАП – цифро-аналоговий перетворювач

ЦФПР – цифрові функціональні перетворювачі розгортуючого типу

CLB (Configurable Logic Block) – програмований логічний блок

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

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

## ВСТУП

При розробці розподілених систем управління виникає завдання, пов'язане з узгодженням сенсорів та датчиків з цифровими системами збору і обробки інформаційних сигналів. Одним з сучасних напрямків розробки систем управління є створення нових і удосконалення базових засобів обчислювальної техніки, що широко використовуються при вирішенні завдань, пов'язаних з перетворенням інформаційних сигналів, що приймають від великої кількості різномірних сенсорів. Структура подібних систем містить множину функціональних перетворювачів і обчислювачів для розв'язання відповідної множини функціональних задач, які можуть бути реалізовані за допомогою програмних засобів або допомогою алгоритмічного забезпечення, що відображено на спеціалізовані апаратні засоби.

В розподілених системах управління та інтелектуальних вимірювальних системах функціональні обчислювачі та перетворювачі, які мають вхідні дані, що представлені в тому числі, і бітовими (імпульсними) потоками. Функціональні обчислювачі, які є засобом нелінійної обробки цифрової інформації, можуть застосовуватись в якості периферійних процесорів функціональних розширювачів високоефективних обчислювальних систем.

Основною метою удосконалення структур апаратних біт-потоків функціональних обчислювачів, що виконують потокову обробку, є спрощення взаємодії блоків та компонентів структури.

В якості елементної бази для реалізації функціональних перетворювачів і обчислювачів застосовуються програмовані логічні інтегральні схеми (ПЛІС). Імплементация пристроїв у ПЛІС дозволяє отримати необхідні умови для розширення функціональних можливостей пристроїв, а також подальшого поліпшення швидкодії та забезпечення їх надійності.

Метою кваліфікаційної роботи є моделі та методи проектування біт-потокowego обчислювача дробово-раціональних функцій, що реалізований на технологічній платформі ПЛІС та програмних засобів, які реалізують принцип його роботи. Обчислення дробово-раціональних функцій у пристрої відбувається паралельно-послідовно в реальному часі по мірі надходження на його вхід бітового потоку. В досліджуваному пристрої спосіб обчислення дробово-раціональної функції здійснюється з використанням способу формування приростів ступінчастих функцій на основі системи різницевих нерівностей з мінімізацією похибки обчислень.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

В розділі проаналізовано особливості функціонального перетворення даних, що представляють собою бітові потоки, дано поняття бітових потоків даних, розглянуто застосування біт-потоків обчислювачів дробово-раціональних функцій в сучасних системах управління, сформульовано мету та постановку завдання дослідження.

### 1.1 Особливості перетворення бітових потоків даних при функціональній обробці сигналів

В даний час має місце підвищення складності завдань по здійсненню обчислень в розподілених системах, таких як системи управління, інтелектуальні вимірювальні системи, сенсорні системи, робототехніка, біоелектроніка, де завдання збору та перетворення інформації від великої кількості джерел – сенсорів і датчиків, є актуальним.

Розподілені комп'ютерні системи управління мають у своєму складі інформаційно-вимірювальну підсистему обробки потокової інформації, яка працює з поточними формами сигналів. Це пояснюється тим, що біт-поточна форма інформаційних сигналів ефективно застосовується як в сенсорних інтерфейсах при цифровій обробці сигналів, так і при формуванні керуючих сигналів в розподілених системах управління [1].

Поява нових методів розв'язання типових задач та створення нових базових елементів для побудови обчислювальних систем, як правило, стає актуальним тоді, коли з'являються нові архітектурні рішення або відбувається якісний скачок в області елементної бази. Сучасний етап розвитку обчислювальних систем характеризується наявністю обох чинників.

Одним з актуальних напрямків в даний час є розробка пристроїв, що виконують перетворення потоків, які складаються з імпульсів. В імпульсних потоках інформативні параметри пов'язані з часом.

Інформаційні потоки, що мають назву – бітові потоки, являють собою імпульси електричної природи. Вони мають одиничну, у яких інформативні параметри пов'язані з часом [2].

В інформаційних сигналах, що являють собою частотно-імпульсні сигнали, інформація передається у таких випадках:

- фізичний показник має періодичний характер дії (наприклад, частота обертання двигуна або виконавчого механізму тощо);

- датчик генераторного типу перетворює фізичний показник на електричний сигнал, частота якого пропорційна цьому показнику (наприклад, п'єзореzonансні датчики для вимірювання температури, тиску, вологості, концентрації хімічних речовин і т.д.);

- електричний сигнал, що відповідає контрольованому показнику і має інформаційний параметр амплітуду, перетворюється на електричний сигнал з інформаційним параметром частота (наприклад, частотно-імпульсна модуляція, що використовується в системах передачі інформації на значні відстані, а також у системах перетворення в цифровий код середнього за деякий часовий інтервал амплітудного значення сигналу).

У широтно-імпульсному вигляді інформація може передаватися у випадках, коли:

- для вимірювання фізіологічного або фізичного показника використовується пристрій знімання інформації з параметричним датчиком (індуктивним або ємнісним);

- для підвищення завадостійкості каналу передачі інформації на значні відстані використовується широтно-імпульсна модуляція сигналу, що відповідає контрольованому фізичному показнику;

- в імплантованих електрокардіостимуляторах між ступенем виснаженості джерела живлення та тривалістю стимулюючих імпульсів

передбачена функціональна залежність, що забезпечує ефективність стимуляції при зниженні напруги живлення. При цьому вимірювання тривалості стимулюючих імпульсів дозволяє у таких випадках визначити напругу живлення акумуляторної батареї.

Інформація про процес або об'єкт може бути представлена кількістю електричних імпульсів, коли для цього використовуються:

- витратомірні датчики, наприклад, датчик турбінного типу, що застосовується у приладах для дослідження функцій зовнішнього дихання в медицині;

- датчики іонізуючих випромінювань, наприклад, у радіонуклеїдних сканерах, гамма камерах, а також у лічильниках іонізуючого випромінювання;

- оптоелектронні датчики кутового або лінійного переміщення.

Вплив, який надається на процес або об'єкт також може бути представлений у вигляді імпульсної послідовності. До таких випадків, можуть бути віднесено в галузі медицини стимуляцію та збудження, які застосовуються для відновлення тканинами функцій, зокрема, електростимуляція нервово-м'язової системи опорно-рухового апарату, електродіагностика.

Якщо зміна стану процесу або об'єкта може здійснюватися регулюванням потужності електроенергії, що підводиться, то управління, використовуваними при цьому перетворювачами, також може розглядатися як завдання формування імпульсних послідовностей або послідовності із заданими параметрами:

- частота використовується для імпульсних перетворювачів постійного струму;

- ширина використовується для імпульсних перетворювачів постійного струму та безпосередніх перетворювачів частоти, перетворювачів змінної напруги;

- кількість імпульсів для систем керування кроковими двигунами.

В імпульсних сигналах інформація представляється безперервно у часі та легко перетворюється в дискретні форми представлення, наприклад, в двійкові коди.

Таким чином, електричні сигнали на вході та виході системи можуть являти собою імпульсні послідовності, зокрема, дискретними інформаційними параметрами.

Передача і обробка інформації, представленої біт-поточною формою сигналу здійснюється способом, що має особливість – це можливість послідовної обробки інформаційних потоків в темпі надходження одиничних імпульсів (біт). Крім того, бітові потоки даних характеризуються високою завадостійкістю, так як одиничний код є непозиційним [3].

Переваги біт-поточного подання інформаційних сигналів сприяють розробці відповідних обчислювачів та перетворювачів для розподілених систем управління.

Важливим напрямом підвищення точності та надійності вимірювально-керуючих систем є розвиток функціональних (інтелектуальних) можливостей перетворювачів, які здійснюють аналого-цифрове, цифро-аналогове перетворення, а також первинну та вторинну математичну переробку частотних і час-імпульсних сигналів. [4].

Передача та цифрова обробка інформаційних сигналів у вигляді потоків характеризуються здійсненням реалізації перетворення за рахунок використання послідовної обробки потоків і методів формування приростів в процесі надходження сигналів, що представляють собою бітові потоки на вхід функціональних пристроїв – перетворювачів та обчислювачів.

При використанні таких інформаційних сигналів, як частота і бітовий потік даних в функціональних обчислювачах, виникає можливість здійснення цифрового функціонального перетворення за певною елементарною функцією, в основі якого лежить цифрова функціональна розгортка, що здійснюється в реальному часі та є основною перевагою таких поширених обчислювальних операцій.

При цифровій функціональній розгортці, що відбувається в реальному часі, передбачено здійснення потокового методу обчислень.

Перетворення здійснюються послідовно, тобто відбувається послідовне обчислення значень відтворюваної функції, які виконуються для сусідніх значень аргументу.

При виконанні операцій функціонального перетворення використовується попередня послідовність обчислень, коли кожне наступне значення апроксимуючої функції, що реалізується у цифровому перетворювачі, обчислюють з урахуванням попереднього результату обчислень. При цьому перше обчислення здійснюється з урахуванням початкової ініціалізації компонентів, що мають бути встановлені в обчислювачі до початку процесу обчислення заданої функції. [5, 6].

В якості технологічної платформи для реалізації біт-потоків пристроїв використовуються програмовані логічні інтегральні схеми (ПЛІС). Застосування ПЛІС не обмежує складність розроблюваних перетворювачів та обчислювачів, що дозволяє при цьому досягти максимальної технологічної надійності та швидкодії.

## 1.2 Застосування біт-потоків функціональних обчислювачів в розподілених системах управління

Розподілені системи управління є системами реального часу. Реальний час розуміється у сенсі *soft real-time*, тобто коли поряд з коректністю функціонування необхідний результат має бути отриманий у заданий «прийнятний» час.

Будь-яка система, у якій істотну роль грає час генерації вихідного сигналу. Це зазвичай пов'язано з тим, що вхідний сигнал відповідає певним змінам у фізичному процесі, і вихідний сигнал повинен бути пов'язаний із самими змінами. Часова затримка від отримання вхідного сигналу до видачі вихідного сигналу має бути невеликою, щоб забезпечити прийнятний час

реакції, так як час реакції є системною характеристикою.

Системи зазвичай вважаються системами реального часу, якщо час їх реакції має порядок мілісекунд; діалоговими вважаються системи з часом реакції порядку кількох секунд, а системах пакетної обробки час реакції вимірюється годинами чи днями. Прикладами систем реального часу є системи управління фізичними процесами із застосуванням обчислювальних машин, системи торгових автоматів, автоматизовані системи контролю та автоматизовані випробувальні комплекси.

Систему управління реального часу приведено на рисунку 1.1.

Система включає в себе основні компоненти, а саме: підсистему управління, яка включає у себе обчислювальну систему, що здійснює керування об'єктом управління; датчики, сенсори та перетворювачі, що є підсистемою, яка сприймає сигнали фізичної природи та перетворює їх за допомогою первинних та вторинних перетворювачів до виду, зручному для подальшої обробки. Система управління містить виконавчі пристрої (актюатори). Актюатори виконують певні дії, спрямовані на об'єкт керування.



Рисунок 1.1 – Система управління

Завдяки успіхам мікроелектроніки цифрові методи подання інформації становлять нині основу обчислювальної, керуючої, телекомунікаційної та

вимірювальної техніки. Точність цифрових пристроїв синхронізації та перетворення інформації тем вище, ніж менше крок дискретизації часу, який зазвичай дорівнює тактовому періоду. Однак завжди залишаються завдання, вирішення яких вимагає кванта часу, значно меншого тактового періоду. До таких завдань належать синхронізація, усунення її перекосу та автотестування систем на кристалі, відновлення синхронізації та декодування даних у каналах їх передачі, радіолокація та супровід по дальності, оптичний зв'язок, вимір інтервалів часу у фізичних експериментах.

В сучасних інформаційних системах управління функціональні обчислювачі та перетворювачі працюють з різними формами сигналів. В даний час поточними формами даних, отриманих з сенсорів фізичних величин, зокрема сенсорів з частотним та час-імпульсним виходом.

У більшості випадків чутливі елементи формують аналогові вихідні сигнали, які необхідно перетворювати в цифрову форму. В розподілених системах сенсори і датчики розташовані на певному віддаленні від центрального обчислювального ядра. Тому доцільно в таких системах використовувати не аналого-цифрове перетворення, а на першому етапі здійснювати перетворення аналогового сигналу в частоту, а на другому етапі виконувати прямі обчислення, тобто, певні перетворення частотних сигналів за необхідною функцією при здійсненні лінеаризації сигналів. Використання сенсорів і датчиків з частотним та час-імпульсним виходом є переважним в розподілених системах управління реального часу. При цьому виконання обчислювальних операцій при функціональному перетворенні даних здійснюють в каналах передачі даних близько від датчиків та сенсорів фізичних величин або функціональні обчислювачі вбудовують в інтелектуальні датчики та сенсори.

Після отримання функціональним обчислювачем бітового потоку – імпульсного або частотного сигналу, що поступає на вхід пристрою, часто потрібно здійснювати лінеаризацію сигналу за певною функцією і далі

перетворений бітовий потік передається в центральне обчислювальне ядро для подальшої обробки сигналу [7].

Інтелектуальні вимірювальні системи часто передбачають первинну та вторинну обробку вимірювальної інформації – інформаційних сигналів, що поступають з датчиків та сенсорів фізичних величин для подальшої обробки сигналу з метою прийняття рішень про результати вимірювань для здійснення завдань управління.

При впливі на виконавчі механізми роботів та маніпуляторів виникає необхідність здійснювати плавні зміни сигналів управління. Для отримання згладжених сигналів управління використовуються перетворювачі, що відтворюють нелінійні функції, такі як степенева, логарифмічна, дробово-раціональна, ірраціональна, експоненціальна, тригонометричні.

Перетворювачі та обчислювачі біт-поточної інформації, що виконують нелінійні функціональні перетворення засновані на застосуванні апроксимуючих функцій, що апроксимують неперервні. При класичній реалізації перетворювачів, заснованих на методах розрахунку з наступними обчисленнями, можуть використовуватись різні наближення.

Для біт-поточкових функціональних перетворювачів, побудованих на основі структурних реалізацій виразів, найефективніше застосування апроксимуючих функцій, представлених відношенням поліномів, тобто дробово-раціональною функцією.

Досліджуваний біт-поточковий обчислювач дробово-раціональних функцій може знайти використання в сучасних розподілених системах управління в якості спеціалізованих вторинних перетворювачів для здійснення лінеаризації сигналів з метою перетворення сигналів з біт-поточною формою, що отримують від сенсорів фізичних величин з частотним та час-імпульсним виходом, а також в якості спеціальної апаратури спряження з виконавчими органами об'єкту керування.

Дробово-раціональні біт-поточкові обчислювачі використовуються:

а) при відтворенні траєкторій рухомих об'єктів в двовірному і тривимірному просторі, зокрема виконавчих механізмів маніпуляторів та роботів;

б) в архітектурах процесорів, що працюють з потоковими формами сигналів в якості зовнішніх апаратних функціональних перетворювачів бітових потоків;

в) при проведенні математичної обробки вторинної вимірювальної інформації в інтелектуальних вимірювальних системах;

г) при виконанні непрямих вимірювань в якості обчислювальних блоків та компонентів вимірювальних систем.

Програмовані логічні інтегральні схеми (ПЛІС) являються елементною базою для апаратної реалізації перетворювачів та обчислювачів елементарних математичних функцій, що призначені для реалізації складних проектів [8].

У різних пристроях і системах сьогодні широко використовуються ПЛІС як для вирішення «нетрадиційних» для універсальних процесорів завдань, наприклад, завдань, пов'язаних з цифровою обробкою сигналів і даних, шляхом реалізації «жорсткої логіки» та інших застосувань, так і для реалізації функціонування пристроїв і систем в динамічно поновлюваних умовах технологічних і користувальницьких вимог, наприклад, для портативних пристроїв масового виробництва.

Навіть незважаючи на те, що використання в таких пристроях спеціалізованих і замовних НВІС виявляється більш вигідним з причини більш високої швидкодії,

ПЛІС активно застосовуються в якості елементної бази спеціалізованих цифрових пристроїв, в тому числі і апаратних функціональних обчислювачів для вирішення різноманітних завдань.

Можливість створення динамічно реконфігурованих пристроїв, зменшення часу проектування є перевагою сучасних ПЛІС. Тому ПЛІС широко використовуються як альтернатива мікроконтролерам і чипсетам у

вбудованих системах з гнучко програмованим набором периферійних пристроїв, інтерфейсів різного типу.

### 1.3 Постановка завдання

Метою кваліфікаційної роботи є дослідження та розробка моделей та методів проектування спеціалізованого апаратного біт-потокowego обчислювача дробово-раціональних функцій на платформі ПЛІС.

Об'єктом дослідження є апаратні обчислювачі дробово-раціональних функцій з інформаційними сигналами, що являють собою бітові потоки даних.

Предмет дослідження – математична модель дробово-раціонального обчислювача, архітектури біт-потокowych обчислювачів на основі конвеєрних структур, апаратні моделі обчислювачів на основі цифрових автоматів.

На основі сформульованої мети визначені завдання дослідження:

- аналіз функціонального перетворення імпульсних потоків в апаратних обчислювачах;
- аналіз способу обчислення дробово-раціональних функцій з використанням способу формування приростів висхідних супінчастих функцій;
- розробка математичної моделі обчислювача дробово-раціональної функції;
- аналіз способу побудови конвеєрних архітектур біт-потокowych обчислювачів на основі алгоритму обчислення поліномів;
- розробка архітектури дробово-раціонального обчислювача;
- розробка апаратної моделі пристрою на основі цифрового автомату;
- розробка HDL-моделей обчислювача;
- верифікація та імплементація моделі пристрою з використанням САПР цифрових пристроїв XILINX ISE в платформу ПЛІС.

## 2 МАТЕМАТИЧНІ МОДЕЛІ БІТ-ПОТОКОВИХ ОБЧИСЛЮВАЧІВ ДРОБОВО-РАЦІОНАЛЬНИХ ФУНКЦІЙ

В даному розділі проаналізовано метод апроксимації відтворювання елементарних неперервних висхідних функцій, що використовується при отриманні математичних моделей біт-потоків обчислювачів. Проаналізовано та розроблено математичну модель біт-потоків обчислювача дробово-раціональних функцій на основі системи різницевої нерівностей, яка реалізується у пристрої. Розглянуто алгоритм обчислення поліноміальних функцій, що використовується в конвеєрних біт-потоків обчислювачах.

### 2.1 Математичне обґрунтування способу обчислення дробово-раціональних функцій

Найважливішою задачею проектування систем та їх компонентів є розробка їх адекватних математичних моделей, що описують процес їх роботи. При цьому мають бути прийняті до уваги початкові умови, допустимі області визначення вхідних величин та відповідна область значень вихідних величин.

Велику роль у процесі проектування відіграє правильний вибір методів аналізу математичної моделі, які дозволяють швидко і з достатньою точністю знайти конструктивні параметри проектного пристрою, що задовольняють вимогам технічного завдання на проектування.

При класичній реалізації функціональних перетворювачів та обчислювачів, заснованих на методах розрахунку з наступними обчисленнями, можуть використовуватись різні наближення.

Для біт-потоків функціональних перетворювачів, побудованих на основі структурних реалізацій виразів, найефективніше застосування апроксимуючих функцій, представлених відношенням поліномів, тобто дробово-раціональною функцією.

Одним з відомих методів відтворення елементарних математичних функцій, що застосовується при проектуванні біт-потоків функціональних обчислювачів є метод ступінчастої апроксимації висхідних неперервних функцій. Даний метод заснований на знаходженні апроксимуючих функцій, що є зворотними по відношенню до заданих неперервних функцій.

Для функцій, що не мають зворотних може бути застосований інший шлях, що передбачає спосіб формування приростів апроксимуючих функцій на основі визначення нерівностей, при реалізації яких на виході обчислювача формується значення ступінчастої функції з заданою похибкою обчислень. Абсолютна похибка обчислень впливає на результат апроксимації функцій та має бути раціональною, що забезпечить мінімізацію похибки обчислення функцій.

В даний час в розподілених системах управління знаходять застосування функціональні обчислювачі неперервних функцій  $y^* = f(x^*)$  та мають обмеження:

$$x^*, y^* \in \mathbb{R}; y \in X; y_k > y_{k-1}, \quad (2.1)$$

$$\frac{dy^*}{dx^*} > 0.$$

З урахуванням обмежень (2.1) елементарні функції є монотонно зростаючими, які належать першій чверті координатної площини.

В біт-потоківому функціональному обчислювачі вхідний  $x$  і вихідний  $y$  сигнали являють собою два бітових (імпульсних) потоки даних.

В [9] зазначено, що на виході обчислювача неперервна функція  $y^* = f(x^*)$  може бути відтворена ступінчастою функцією

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

де  $x$  – аргумент функції, що представляє собою бітовий потік, тобто  $x = 1, 2, 3, \dots$ ;

$f(x)$  – ступінчата апроксимуюча функція;

$|\delta_{\max}|$  – абсолютна похибка відтворення неперервної функції ступінчастою.

При цьому  $0,5 \leq |\delta_{\max}| < 1$  – граничне значення абсолютної похибки відтворення відповідних неперервних функцій.

Мінімальна похибка обчислень  $|\delta_{\max}| = 0,5$  є раціональною з точки зору точності обчислення елементарних неперервних функцій. Квадратні дужки у виразі (2.2) – ціла частина числа.

При способі формування приростів ступінчастих функцій, має місце процес округлення дробових значень ґратчастої функції в цілих точках. Такому обчисленню відповідає певний порядок проходження вихідних біт пристрою, який залежить від граничного значення  $|\delta_{\max}|$  абсолютної похибки обчислення.

Процес відтворення функції (2.2) може бути здійснений на основі визначення зворотної функції, якщо функція 2.2 має зворотну. При цьому значення номерів вибірок певних біт  $x_y$  з вхідного бітового потоку  $x$  визначають з нерівності:

$$\Psi(y - |\delta_{\max}|) \leq x_y < \Psi(y - |\delta_{\max}|) + 1, \quad (2.3)$$

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

Вибірки  $x_y$ , що обираються з вхідного бітового потоку, можуть бути знайдені шляхом послідовної підстановки  $y = 1, 2, 3, \dots, y_k$  в нерівність (2.3). Спосіб забезпечує неперервний процес відтворення функції (2.2) в реальному часі.

Обчислення у пристрої функції з абсолютною похибкою  $|\delta_{\max}| = 0,5 \epsilon$  мінімальною та раціональною.

Для біт-потоківих функціональних перетворювачів, побудованих на основі структурних реалізацій виразів, найефективніше застосування апроксимуючих функцій, що представлені відношенням поліномів, тобто дробово-раціональною функцією.

В [10] розглянуто математичні моделі обчислювачів дробово-раціональних функцій, що перетворюють бітові потоки.

Дробово-раціональні обчислювачі відтворюють неперервні функції:

$$y^* = \frac{\sum_{i=0}^n a_i x^{*i}}{\sum_{j=0}^1 b_j x^{*j}}, \quad (2.4)$$

де  $i, j, a_i, b_j$  – цілі числа.

В даний час при реалізації дробово-раціональних функцій найбільший практичний інтерес становлять схемні реалізації, у яких формування поліномів та операцію поділу здійснюють структурними методами без застосування окремих множинних, степеневих, підсумовуючих та ділильних пристроїв. При цьому для визначення коефіцієнтів у раціональному дробі (2.4), що забезпечують задану похибку апроксимації, використовуються відомі алгоритми наближення [7].

Досліджуваний біт-потоківий обчислювач дробово-раціональних функцій, що знаходить застосування в системах управління і контролю відтворює неперервну функцію:

$$y^* = \frac{\sum_{i=0}^n a_i x^{*i}}{m}, \quad (2.5)$$

де  $i$ ,  $a_i$ ,  $m$  – цілі числа.

Чисельник (2.5) – це поліном  $n$ -го порядку, а поліном в знаменнику – константа  $m$  ( $b_0 = m$ ), оскільки  $b_1 = b_{1-1} = b_{1-2} = \dots = b_1 = 0$ .

Дробово-раціональна функція, що апроксимує неперервну (2.5) має вигляд:

$$y = \left[ \frac{\sum_{i=0}^n a_i x^i}{m} + |\delta_{\max}| \right]. \quad (2.6)$$

де  $x$  – аргумент функції. Він є бітовим потоком даних;

$i$ ,  $a_i$  – цілі додатні числа;

$|\delta_{\max}|$  – значення абсолютної похибки ділення  $\sum_{i=0}^n a_i x^i$  на константу  $m$ .

Абсолютна похибка обчислення функції  $|\delta_{\max}| = 0,5$ , є раціональною, тому (2.6) з урахуванням заданої мінімальної похибки:

$$y = \left[ \frac{\sum_{i=0}^n a_i x^i}{m} + 0,5 \right]. \quad (2.7)$$

При обчисленні функції (2.7) необхідно виконання двох етапів, що

мають бути здійсненні в досліджуваному пристрої:

- обчислення поліноміальної функції  $\sum_{i=0}^n a_i x^i$ , що є в чисельнику

функції (2.7);

- ділення поліному на число  $m$  з похибкою  $|\delta_{\max}| = 0,5$ .

На графіку (рис. 2.1) зображено неперервну 1 та апроксимуючу 2 дробово-раціональні функції.

Апроксимуюча ступінчаста функція відтворює безпервну з абсолютною похибкою  $|\delta_{\max}| = 0,5$ .

Зображені функції задовольняють умовам (2.1).

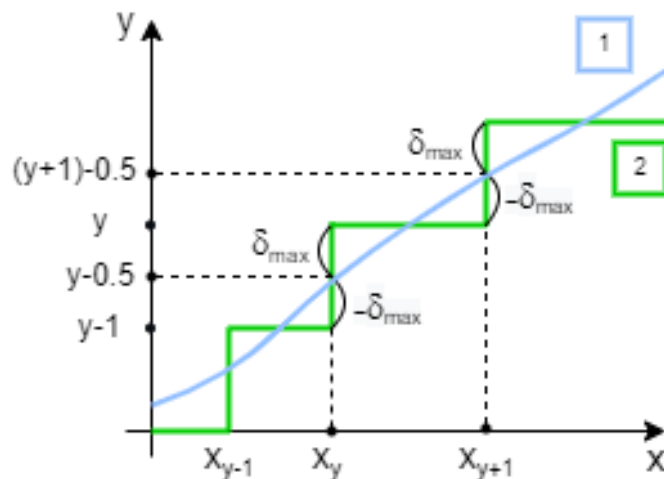


Рисунок 2.1 –Дробово-раціональна функція

2.2 Математична модель біт-потокowego обчислювача досліджуваної функції

В апаратних біт-потокowych обчислювачах реалізація ступінчастих функцій, що апроксимують безпервні здійснюється на основі методу відтворення зворотних функцій. Але дробово-раціональна функція не має

зворотної їй функції, тому отримання математичної моделі біт-потокowego обчислювача дробово-раціональної функції здійснюється за допомогою способу формування приростів висхідних функцій на основі системи різницевих нерівностей [10].

Так як функція (2.7) не має зворотної їй функції, було отримано нерівність:

$$\frac{\sum_{i=0}^n a_i x^i}{m} \geq \left(y - \frac{1}{2}\right), \quad (2.8)$$

Після виконання перетворень нерівність (2.8) набуває вигляду:

$$2 \sum_{i=0}^n a_i x_y^i \geq m(2y_k - 1), \quad (2.9)$$

Нерівність (2.9) представляє собою нерівність, що реалізується в обчислювачі дробово-раціональних функцій.

З нерівності (2.9) випливає, що при подачі на вхід пристрою вхідного бітового потоку  $x$ , що є імпульсною послідовністю, перший ( $y_1=1$ ) біт вихідного потоку  $y$  пристрою з'явиться на виході пристрою при подачі на вхід біту  $x_1$  вхідного потоку  $x = 1, 2, 3, \dots$ . При цьому може бути записана нерівність:

$$2 \sum_{i=0}^n a_i x_1^i \geq m, \quad (2.10)$$

Якщо перейти до різницевих нерівностей, то другий ( $y_2=2$ ) і наступні біти вихідного бітового потоку обчислювача з'являться на виході пристрою при виконанні нерівності, що має вигляд:

$$2\left(\sum_{i=0}^n a_i x_y^i - \sum_{i=0}^n a_i x_{y-1}^i\right) + \Delta_{y-1} \geq 2m, \quad (2.11)$$

Нерівність (2.11) що може бути застосована при отриманні математичної моделі обчислювача у вигляді різницевих нерівностей:

Вибіркові значення  $x_y$ , що обираються з вхідного потоку визначаються при обчисленні приростів гратчастої функції  $2\sum_{i=0}^n a_i x_y^i$ . При цьому здійснюється її порівняння з приростами функції  $m(2y_k - 1)$  з урахуванням різниці  $\Delta_{y-1}$ , яка отримана при попередньому кроці обчислення.

Система нерівностей, що описує математичну модель біт-потокowego обчислювача дробово-раціональних функцій на основі (2.9), приведено нижче:

$$\begin{aligned} 2\sum_{i=0}^n a_i x_1^i &\geq m \\ 2\left(\sum_{i=0}^n a_i x_2^i - \sum_{i=0}^n a_i x_1^i\right) + \Delta_1 &\geq 2m \\ 2\left(\sum_{i=0}^n a_i x_3^i - \sum_{i=0}^n a_i x_2^i\right) + \Delta_2 &\geq 2m \\ \dots &\dots \\ 2\left(\sum_{i=0}^n a_i x_y^i - \sum_{i=0}^n a_i x_{y-1}^i\right) + \Delta_{y-1} &\geq 2m, \end{aligned} \quad (2.13)$$

де  $\Delta_{y-1}$  – значення різниці, яка отримана при виконанні попередньої нерівності математичної моделі в результаті порівняння приростів поточних

значень функцій  $2 \sum_{i=0}^n a_i x^i$  и  $m(2y-1)$  між двома сусідніми вузлами апроксимації функції.

В (2.13) різниці  $\Delta_1$  і  $\Delta_{y-1}$  визначаються за формулами:

$$\Delta_1 = 2 \sum_{i=0}^n a_i x_1^i + \Delta_0 - m, \quad (2.14)$$

$$\Delta_{y-1} = 2 \left( \sum_{i=0}^n a_i x_y^i - \sum_{i=0}^n a_i x_{y-1}^i \right) + \Delta_{y-2} - 2m. \quad (2.15)$$

На виході обчислювача буде сформований вихідний біт  $u_k$  при виконанні кожної нерівності системи (2.13).

Відповідно до завдання дослідження на основі формул (2.3), (2.9) було отримано математичну модель біт-потокowego обчислювача для заданої дробово-раціональної функції, що має у своєму аналітичному виразі поліном другого порядку:

$$y = \left[ \frac{\sum_{i=0}^2 a_i x^i}{m} + 0,5 \right] \quad (2.16)$$

де абсолютна похибка обчислення  $|\delta_{\max}| = 0,5$ .

При обчисленні функції (2.16) необхідно виконання двох етапів, що мають бути здійсненні в досліджуваному пристрої:

- обчислення поліноміальної функції  $\sum_{i=0}^2 a_i x^i$ ;
- ділення поліному на число  $m$  з похибкою  $|\delta_{\max}| = 0,5$ .

Опустивши квадратні дужки в (2.16) та виконавши перетворення отримано нерівність:

$$2 \sum_{i=0}^2 a_i x_y^i \geq m(2y_k - 1). \quad (2.17)$$

На підставі формули (2.14), отримано математичну модель біт-потокowego біт-потокowego обчислювача дробово-раціональної функції (2.16) що має вигляд:

$$\begin{aligned} 2 \sum_{i=0}^2 a_i x_1^i &\geq m \\ 2 \left( \sum_{i=0}^2 a_i x_2^i - \sum_{i=0}^2 a_i x_1^i \right) + \Delta_1 &\geq 2m \\ 2 \left( \sum_{i=0}^2 a_i x_3^i - \sum_{i=0}^2 a_i x_2^i \right) + \Delta_2 &\geq 2m \\ \dots &\dots \\ 2 \left( \sum_{i=0}^2 a_i x_y^i - \sum_{i=0}^2 a_i x_{y-1}^i \right) + \Delta_{y-1} &\geq 2m. \end{aligned} \quad (2.18)$$

При цьому різниці  $\Delta_1$  і  $\Delta_{y-1}$  визначаються за формулами:

$$\Delta_1 = 2 \sum_{i=0}^2 a_i x_1^i + \Delta_0 - m \quad (2.19)$$

$$\Delta_{y-1} = 2 \left( \sum_{i=0}^2 a_i x_y^i - \sum_{i=0}^2 a_i x_{y-1}^i \right) + \Delta_{y-2} - 2m. \quad (2.20)$$

Систему нерівностей (2.19) можна реалізувати обчислювачем, шляхом одночасного обчислення приростів ґратчастих функцій  $2 \sum_{i=0}^2 a_i x_y^i \geq m(2y_k - 1)$

і послідовного формування бітів вихідного бітового потоку обчислювача при виконанні кожної нерівності системи (2.18).

### 2.3 Алгоритм обчислення поліноміальних функцій

В досліджуваному обчислювачі застосовано конвеєрну обробку біт-потоківих даних.

Для функції (2.6) необхідно обчислювати поліном  $n$ -го порядку, що є в чисельнику функції.

Поліном  $n$ -го порядку має вигляд:

$$y = \sum_{i=0}^n a_i x^i, \quad (2.21)$$

де  $n, a_i$  – цілі числа [11].

В поліномі з цілочисельними коефіцієнтами  $a_i$  існує зв'язок та відповідність послідовності його значень функції  $U_0, U_1, U_2, U_3, \dots, U_i$ , що відповідають значенням аргументу  $x_i = 0, 1, 2, 3, \dots, i$ .

При підстановці у (2.21)  $x_i = 0, 1, 2, 3, \dots, i$  отримано числову послідовність  $U_i$ .

В результаті утворений ряд значень функції  $U_0, U_1, U_2, U_3, \dots, U_i$  є арифметичним рядом  $n$  – го порядку.

Синтезу конвеєрного біт-потоківого обчислювача поліномів здійснюється методом пониження порядку різниць.

Значення функції  $U_i$  для полінома (2.21) визначаються як:

$$y_i = f(i + 1) - f(i), \quad (2.22)$$

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

Різницеве рівняння першого порядку має вигляд:

$$\Delta_i = y_{i+1} - y_i. \quad (2.23)$$

Різницеве рівняння другого порядку має вигляд:

$$\Delta_i^2 = \Delta_{i+1} - \Delta_i, \quad (2.24)$$

Різницеве рівняння n-го порядку визначається як:

$$\Delta_i^n = \Delta_{i+1}^{n-1} - \Delta_i^{n-1}, \quad (2.25)$$

де  $i = 0, 1, 2, 3, \dots$ .

В результаті отримано систему рівнянь:

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

Система рівнянь (2.26) представляє собою математичну модель конвеєрного біт-потокowego обчислювача поліномів.

Розглянута методика використовується при проектуванні архітектури біт-потокowego пристрою обчислення дробово-раціональних функцій.

Перед подачею на вхід бітового потоку компоненти архітектури ініціалізуються числами  $y_0, \Delta_0, \Delta_0^2, \dots, \Delta_0^n$  відповідно.

Досліджуваний пристрій обчислює поліном другого порядку:

$$y = \sum_{i=0}^2 a_i x^i . \quad (2.27)$$

При підстановці  $x=1, 2, 3, \dots$  у (2.27) визначають арифметичний ряд 2-го порядку значень функції.

На наступному етапі визначаються значення різниць у рядах різниць 1-го і 2-го порядків за допомогою формул (2.23), (2.24).

В результаті обчислень арифметичний ряд різниць 2-го порядку містить константу  $2a$  або першу похідну першого члена полінома другого порядку.

При розробці досліджуваного пристрою було використано алгоритм обчислення поліноміальної функції.

### 3 АРХІТЕКТУРА БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА ДРОБОВО-РАЦІОНАЛЬНИХ ФУНКЦІЙ

В розділі розглянуто узагальнену архітектуру біт-потоків обчислювачів. Розроблено архітектуру обчислювача дробово-раціональних функцій, що є синтезом двох блоків: блоку конвеєрної архітектури обчислювача поліноміальних функцій та блоку обчислювача лінійних функцій.

#### 3.1 Узагальнена архітектура біт-потоків обчислювачів

При проектуванні функціональних обчислювачів виконання обчислювальних процедур уніфікованим способом при реалізації математичних моделей обчислювачів дозволяє застосовувати в архітектурах використання уніфікованих вузлів та блоків.

Математична модель обчислювача ірраціональних функцій, представляє собою систему різницевих нерівностей. На виході обчислювача вихідний біт апроксимуючої функції з'являється при виконанні кожної нерівності системи. При виконанні умови кожної нерівності математичної моделі обчислювача (2.18) відбувається порівняння значень відтворюваних функцій, що записані в правій та лівій частинах (2.18).

В [11] розглянуто узагальнену архітектуру біт-потоків обчислювачів елементарних функцій, яку наведено на рис. 3.1.

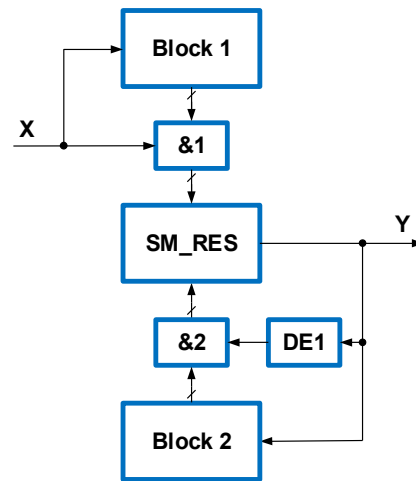


Рисунок 3.1 – Узагальнена архітектура біт-потокowego обчислювача

Узагальнена архітектура дозволяє здійснювати синтез структур біт-потокowych обчислювачів, де  $x$ ,  $y$  – бітові потоки даних.

Узагальнена архітектура містить компоненти: перший Block1 і другий Block2. У блоках формуються відповідно прирости ґратчастих функцій лівої та правої частин нерівностей (2.18).

Функціональне перетворення розгортуючого типу може бути здійснено за допомогою базової архітектури біт-потокowego обчислювача поліноміальних функцій, яке використовується при побудові конвеєрних структур. Блок Block1 представляє собою конвеєрну архітектуру на основі біт-потокowego поліноміального обчислювача.

Перевагами архітектур, що розглядаються, є простота побудови архітектури, швидкодія та завадостійкість пристроїв.

Основне обчислювальне ядро біт-потокowego обчислювача – паралельний суматор SM\_RES, що має зворотний зв'язок. Він застосовується як елемент порівняння паралельних двійкових кодів приростів ґратчастих функцій.

Прирости двох одночасно відтворюваних ступінчастих функцій, які містять нерівності математичної моделі пристрою (2.18) порівнюються у суматорі.

Біти переповнення суматору  $u$  є відповідними бітами початку формування кожної чергової сходинки апроксимуючої функції.

### 3.2 Загальна архітектура дробово-раціонального обчислювача

Приведена у попередньому підрозділі узагальнена архітектура обчислювача може бути використана для побудови загальної архітектури дробово-раціонального обчислювача. Загальна архітектура пристрою, що обчислює дробово-раціональну функцію (2.7) представлена на рис. 3.2 [12].

Архітектура є синтезованою структурою. Структура включає блоки: Block 1 - обчислювач поліноміальних функцій, який сприймає на вхід

бітовий потік сигналу. Пристрій відтворює функцію  $2 \sum_{i=0}^n a_i x_y^i$ , Block 2 являє

собойо дільник чисел. Пристрій побудовано на основі обчислювача лінійних функцій. Він реалізує функцію  $m(2y_k - 1)$  нерівності (2.9).

На виході суматора результату SUM\_RES формуються сходинки відтворюваної функції (2.7). з Block 1 в SUM\_RES поступають прямі коди чисел за допомогою біт вхідного потоку  $x$ . При цьому додаткові коди чисел поступають з Block 2 за допомогою вихідних біт потоку  $y$ , що поступають у зворотний зв'язок блоку Block 2.

Дробово-раціональний обчислювач в своєму складі містить:

- суматор результату SUM\_RES;
- $n-1$  суматорів SUM1, ..., SUM $n-1$ ;
- регістри RG1, RG2;
- елементи затримки DE'1, DE1, DE2, ..., DE $n-1$ ;
- групи елементів &'1, &1, &2, ..., & $n$ .

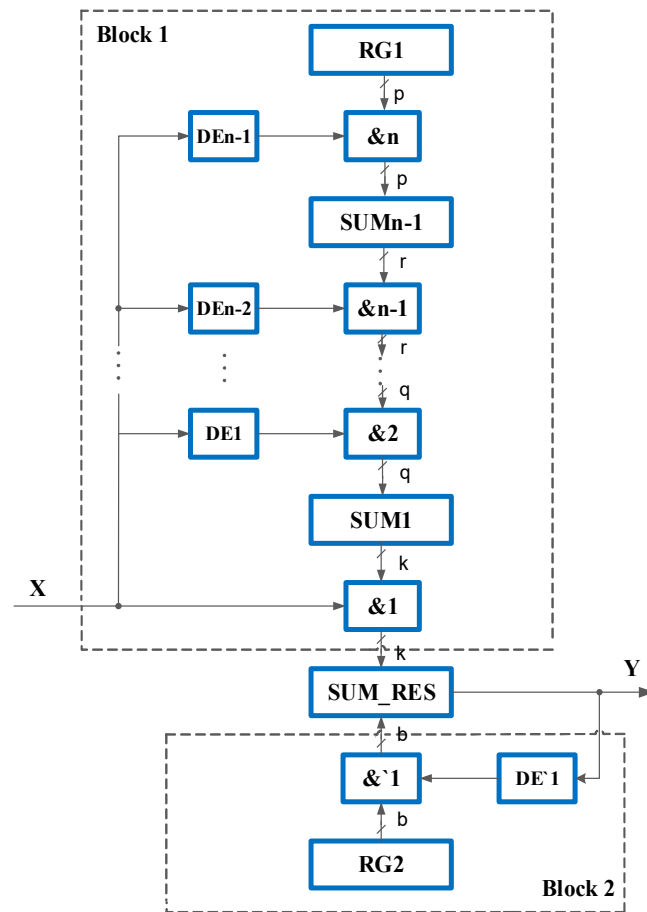


Рисунок.3.2 – Загальна архітектура біт-поточкового обчислювача дробово-раціональних функцій

Архітектури біт-поточкових обчислювачів, що будуються на основі конвеєрних структур передбачають реалізацію потокового методу обчислень, що здійснюється в реальному часі при подачі на вхід пристроїв бітового потоку. При цьому одночасно виконується паралельно-послідовне функціональне перетворення біт вхідного потоку.

В пристрою послідовне обчислення значень функції здійснюється для значень аргументу  $x=1, 2, 3, \dots$ . Попередній результат обчислення функції враховується при обчисленні кожного наступного значення функції.

У пристрою перше обчислення функції виконується з урахуванням ініціалізації компонентів обчислювача, що ініціалізуються перед подачею на

вхід пристрою вхідної бітової послідовності. Абсолютна похибка є раціональною з точки зору точності обчислення функцій та дорівнює 0,5.

### 3.3 Синтез архітектури досліджуваного обчислювача дробово-раціональних функцій

На основі загальної архітектури біт-потокowego обчислювача дробово-раціональних функцій було розроблено архітектуру, що є синтезом двох архітектур: архітектури біт-потокowego обчислювача поліномів та архітектури обчислювача лінійних функцій. Створена архітектура реалізує досліджувану функцію (2.16).

Розглянемо детально архітектуру запропонованого біт-потокowego пристрою обчислення дробово-раціональних функцій та принцип його роботи (рис. 3.3).

На виході біт-потокowego обчислювача відтворюється апроксимуюча функція (2.16) с з похибкою  $|\delta_{\max}| = 0,5$ .

Обчислення функції (3.1) в пристрою здійснюється виконанням двох операцій над вхідним бітовим потоком  $x$ :

- обчислення полінома другого порядку  $\sum_{i=0}^2 a_i x^i$ ;
- поділу отриманого результату на число  $m$ .

Абсолютна похибка має місце при поділу чисельника функції (2.16) на знаменник  $m$ , тому реалізація обчислень врахована з раціональною мінімальною похибкою.

Вираз функції (2.16) має вигляд:

$$y = \left[ \frac{a_2 x^2 + a_1 x + a_0}{m} + 0,5 \right] . \quad (3.1)$$

Досліджуваний обчислювач, має у своєму складі: регістри RG1, RG2; суматори SUM1, SUM2; елементи затримки DE1, DE2; групи елементів &1, &2, &3.

В архітектурі перший блок Block 1 являє собою конверсний біт-поточковий обчислювач поліноміальних функцій, що відтворює функцію

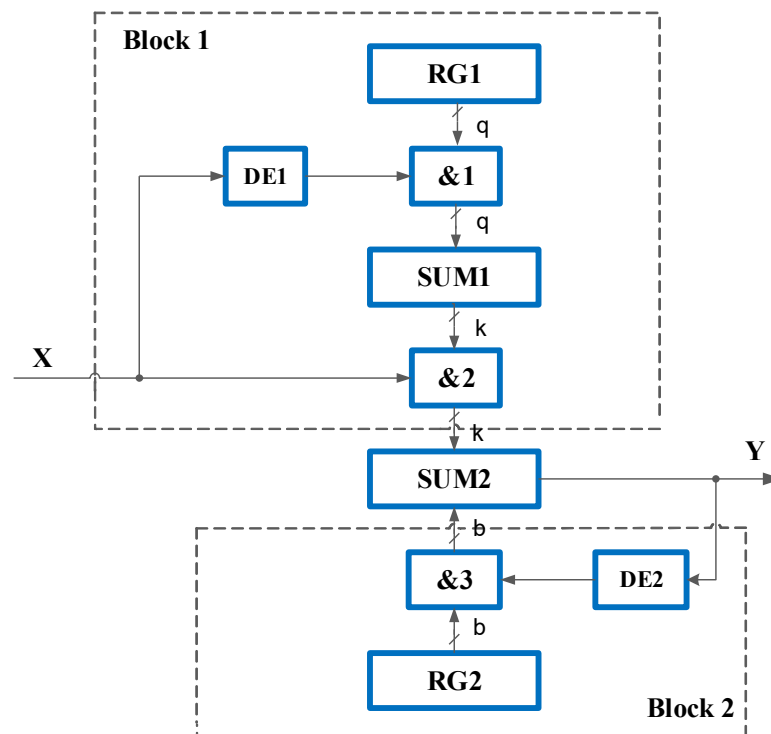
полінома другого порядку  $\sum_{i=0}^2 a_i x^i$ . Біт-поточковий обчислювач поліномів

обчислює поліном другого порядку і називається квадратор.

Блок обчислювача поліноміальної функції має конверсну структуру і містить суматори SUM1, SUM2, регістр RG1, групи логічних елементів &1 та &2, елемент затримки DE1.

Другий блок Block 2 являє собою обчислювач лінійної функції і використовується як дільник чисел. Блок призначений для здійснення

операції – ділення полінома  $\sum_{i=0}^2 a_i x^i$  на константу знаменника  $m$  (2.18).



### Рисунок 3.3 – Структура досліджуваного біт-потокowego обчислювача

Блок дільника чисел має у складі наступні компоненти: суматор SUM2, регістр RG2, групу елементів &3, елемент затримки DE2.

На виході SUM2 буде формуватись вихідна бітова послідовність, що представляє результат функції  $u$ .

При обчисленні досліджуваної функції (2.16) пристрій працює на початковому етапі в режимі вибірки біт з вхідної бітової послідовності, а надалі переходить в режим генерації групи біт вихідної послідовності, що формуються на кожен вхідний біт потоку.

У компонентах архітектури в початковому стані ініціалізуються певними числами перед подаванням на вхід обчислювача вхідного потоку  $x$ .

Компоненти обчислювача ініціалізуються числами, що визначені заздалегідь. Нижче приведено пояснення щодо ініціалізації елементів пристрою

У суматорі SUM1 ініціалізація здійснюється числом – значення суми коефіцієнтів  $a_2 + a_1$  з врахуванням знаку при  $a_1$ . Те ж значення буде отримано при визначенні першого члена арифметичного ряду різниць 1-го порядку  $\Delta_0$ .

У регістрі RG1 встановлюється значення константи в арифметичному ряду різниць 2-го порядку  $\Delta_0^2$ .

У регістрі RG2 встановлюється значення  $-2m$ , тобто подвоєне значення знаменника  $m$ , що є у математичній моделі пристрою, яке записане у в правій частині нерівностей (2.18).

Суматор SUM2 ініціалізується числом  $-m \pm a_0$ . В цьому значенні враховано початкову ініціалізацію SUM2, в якому з боку поліноміального обчислювача записано число  $\pm a_0$ , а у дільнику чисел SUM2 має бути ініціалізований числом  $-m$ .

Суматор SUM2 у структурі – це спільний компонент, який застосований для здійснення операції порівняння паралельних кодів чисел лівої та правої частин нерівностей (2.18).

При здійсненні ініціалізації суматора SUM2 необхідно записувати число  $a_0$  в прямому коді, якщо число позитивне. Якщо число  $-a_0$  – негативне, необхідно ініціалізувати SUM2 числом у додатковому коді  $-a_0$ .

За допомогою групи елементів &2 числа, що формуються у SUM1 переносяться у SUM2 в прямому двійковому коді із зсувом на один розряд вправо, тобто переноситься подвійне значення числа із SUM1.

Отже, в паралельному суматорі SUM2 надійде значення  $2 \sum_{i=0}^2 a_i x^i$  у паралельному двійковому коді під час подачі на вхід обчислювача  $x$  біт вхідного потоку.

Регістр RG1 з'єднаний порозрядно з рахунковими входами SUM1 через групу елементів &1. В результаті при надходженні на вхід пристрою кожного біту вхідного бітового потоку  $x$ , з регістра RG1 в SUM1 переноситься константа, що утворена в ряду різниць другого порядку, тобто  $\Delta_0^2$  і виконується підсумовування з вмістом суматора SUM1.

У блоку Block 2 здійснюється ділення поліному на константу  $m$ . При цьому з регістра RG2 у суматор SUM2 переноситься паралельний двійковий додатковий код числа  $-2m$ , що зберігається у RG2. Значення  $-2m$  відповідно до правої частини нерівностей в (2.18).

За допомогою групи елементів &3, кожен біт вихідного бітового потоку  $y$  в зворотному зв'язку переносить число  $-2m$  в додатковому двійковому коді з RG 2 в SUM2.

Якщо в початковому стані в SUM2 записано число  $-m + a_0$  в процесі роботи пристрою на виході SUM2 буде формуватися результат обчислення функції (2.16) у вигляді вихідного бітового потоку  $y$ .

Розглянемо детально роботу обчислювача.

З приходом першого біту бітового потоку  $x$  на вхід пристрою виконуються наступні операції:

- за допомогою групи елементів  $\&2$  прямий бінарний код числа  $a_2 + a_1$  із SUM1, переноситься в SUM2 зсувом на один розряд в бік старших розрядів і підсумовується з його вмістом. В результаті вміст SUM2 стане рівним  $(-m + a_0) + 2(a_2 + a_1) = \Delta_1$ .

- Якщо число в прямому коді, що поступає з SUM1 в SUM2 більше від'ємного числа в SUM2, то на виході SUM2 з'явиться біт переповнення і в SUM2 залишиться число  $\Delta_1$ ;

- через елемент затримки DE1 за допомогою групи елементів  $\&1$  прямий бінарний код числа  $2a$ , записаного в RG1, переноситься в SUM1 і підсумовується з його вмістом.

- В результаті вміст SUM1 стане рівним  $a_2 + a_1 + 2a$ . Далі при надходженні на вхід пристрою бітів вхідного бітового потоку  $x$  в SUM1 будуть утворюватися члени арифметичного ряду різниць 1-го порядку.

Перший біт переповнення SUM2 буде мати місце при надходженні на вхід пристрою біту з номером  $x_1$ . При цьому виконається перша нерівність

системи (2.18), а саме,  $2 \sum_{i=0}^2 a_i x_1^i \geq m$ .

Перший біт переповнення SUM2:

- з виходу SUM2 надходить на вихід в канал  $u$ ;
- пройшовши елемент затримки DE2, відкриває групу елементів  $\&3$ . При цьому число, яке зберігається в регістрі RG2  $-2m$  переноситься в SUM2, тобто в додатковому коді і підсумовується з вмістом SUM2.

В результаті в SUM2 утворюється число  $-2m + \Delta_1$ . Так як  $\Delta_1 < 2m$ , біт переповнення на виході SUM2 буде відсутній.

Надалі з надходженням чергових вхідних біт аргументу описані процеси в схемі будуть циклічно повторюватися.

При надходженні на вхід обчислювача біту  $x_2$  при якому реалізується наступна нерівність системи (2.18) на виході SUM2 сформується другий біт переповнення.

В результаті при подачі на вхід пристрою вхідної бітової послідовності на виході буду формуватись біти вихідної послідовності  $y$ , що є результатом обчислення досліджуваної дробово-раціональної функції (2.16).

### 3.2 Вибір елементної бази для апаратної реалізації обчислювача

Апаратний обчислювач досліджуваної функції реалізовано на платформі ПЛІС. Проектування цифрових пристроїв на основі програмованої логіки надає низку можливостей. Головною відмінністю програмованої логіки є можливість їх гнучкої настройки на виконання заданих функцій самим користувачем. Мікросхеми програмованої логіки вдало поєднують гнучкість замовних ВІС та доступність стандартної логіки.

Процес проектування цифрових пристроїв на основі НВІС програмованої логіки полягає в описі його функціонування на вхідній мові програмного засобу, що використовується, виконанні автоматизованого синтезу, проведення моделювання, верифікації та налаштування архітектури кристала програмованої логіки.

Сучасні програмовані логічні пристрої мають низьку вартість, високу швидкодію (професійні серії мікросхем працюють на частоті 500 МГц), широкі функціональні можливості, багаторазовість програмування, низьку споживану потужність, компактність.

Технології Xilinx представляють можливість скоротити час розробки пристроїв, проектувати і займатися відладкою унікальних пристроїв на базі ПЛІС Xilinx. Крім того, оскільки ПЛІС Xilinx представляють собою готові до програмування стандартні вироби.

В даний час продукція Xilinx знаходить саме широке застосування: від телекомунікацій до автомобільного і промислового обладнання.

ПЛИС FPGA Spartan фірми Xilinx призначено для недорогої серійної продукції. Сучасна платформа ПЛИС Xilinx Spartan, створена для реалізації недорогих рішень. Завдяки ПЛИС Spartan-3 Xilinx проникають в ті сегменти ринку продукції масового попиту, які раніше вважалися оптимальними для застосування ASIC середнього рівня.

Ефективне використання переваг технології пропонують користувачам функціональні FPGA з низькою вартістю. Xilinx Spartan користується великою популярністю серед виробників електроніки в різних сферах ринку, таких як споживча електроніка, цифрове відео, промислове, медичне та комунікаційне обладнання. Останнім по часу появлення являється сімейство Spartan-3E, що забезпечує логічну ємність до 1,6 млн системних вентилів, до 376 програмованих виводів і гнучку архітектуру. Сімейство Spartan, розроблене для досягнення малого рівня енергоспоживання.

При використанні ПЛИС можна забезпечити зниження споживчої потужності. дозволяє зменшити габарити пристрою і підвищити надійність пристроїв і системи в цілому.

Програмовані вентильні матриці (Field Programmable Gate Arrays - FPGA) вперше були розроблені фірмою Xilinx у 1985р. В даний час спостерігається бурхливий розвиток архітектури FPGA. З'являються нові покоління цих пристроїв, що мають підвищену швидкодію і функціональну складність.

Використання програмованої логіки особливо актуальне на етапі проектування, розробки дослідного зразка, при коригуванні проекту в процесі розробки або реконфігурації структури пристрою під час його функціонування. На сьогодні інтенсивно розвивається виробництво інструментальних моделей різного призначення сімейства Xilinx: програмних засобів, призначених для проектування, налагодження, синтезу пристроїв, заснованих з урахуванням кристалів сімейства Xilinx. Застосування готових

інструментальних засобів дозволяє скоротити час розробки проектного пристрою, уникнувши можливих помилок, що вносяться при виготовленні друкованої плати. Крім того, налагоджені інструментальні модулі полегшують процес освоєння нових методів та програмних засобів систем на основі конфігурованих мікропроцесорних ядер. Інтерес для ефективного вивчення наскрізного проектування мікропроцесорних систем на основі ядер сімейства PicoBlaze представляє інструментальний комплект Spartan 3E. Крім широкого спектру ПЛІС фірма Xilinx надає сучасне програмне забезпечення, яке є необхідним для розробки та конфігурування кристалів.

Застосування програмних засобів ISE дозволяє значно скоротити час розробки та підвищити рівень ефективності результатів за рахунок застосування удосконалених методів проектування, алгоритмів синтезу, розміщення та трасування проектних рішень на кристалі.

САПР включає синтез та моделювання на HDL, розміщення та трасування проектів, а також програмування кристалів за JTAG-інтерфейсом.

ISE WebPACK забезпечує просту в освоєнні та повнофункціональне середовище проектування

Таким чином, компанією Xilinx розроблено зручний для роботи засіб проектування. ISE Foundation – повнофункціональна версія програмного забезпечення ISE, що поєднує підтримку всіх провідних сімейств ПЛІС FPGA та CPLD фірми Xilinx. Це найповніший засіб проектування для пристроїв з програмованою логікою, забезпечує оптимальне управління споживаною потужністю, має найкращу по галузі службу технічної підтримки Xilinx.

ISE Foundation також підтримує методологію часткового реконфігурування, що дозволяє зменшити розмір проектного пристрою, його масу, споживану потужність та вартість.

І ISE Foundation, і ISE WebPACK у даний час підтримують технологію Fmax, яка забезпечує підвищення продуктивності для ПЛІС

Алгоритми синтезу нового покоління передбачають технологію, що визначається терміном *retiming*; відображення проекту на фізичні ресурси з урахуванням заданих часових параметрів; глобальну оптимізацію та оптимізацію за підсумками розміщення.

Для реалізації використаний кристал типу FPGA сімейства Spartan-3E серії XC3S500E фірми Xilinx, технічні характеристики якого наведено у таблиці 3.1. Таблиця містить основні характеристики кристалу (логічні осередки, число системних вентилів, розмірність масиву CLB, у CLB).

Таблиця 3.1 – Параметри кристала типа 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

## 4 ЕКСПЕРИМЕНТАЛЬНА АПАРАТНА РЕАЛІЗАЦІЯ ДОСЛІДЖУВАНОВОГО ПРИСТРОЮ

У розділі представлено результати проектування апаратної моделі біт-потокowego обчислювача досліджуваної функції, а також верифікація отриманого рішення. Апаратна модель обчислювача розроблена мовою VHDL з використанням автоматної моделі опису. Для верифікації проекту використовувалося автоматизоване тестове середовище (test-bench). Модель обчислювача імплементовано в програмовану логічну інтегральну схему Xilinx Spartan.

### 4.1 Етапи розробки експериментальної апаратної моделі обчислювача

На основі розробленої математичної моделі обчислювача та його запропонованої структури було виконано етапи схемотехнічного проектування з використанням САПР цифрових пристроїв.

Розробка апаратної моделі обчислювача передбачає здійснення певних етапів, що приведено нижче, а отже:

- складання специфікації пристрою, що проектується та проведення підготовчих теоретичних розрахунків, зокрема, обчислення значень досліджуваної функції при заданій довжині вхідного бітового потоку  $x = x_{\max}$ ;

- на основі отриманої математичної моделі обчислювача дробово-раціональних функцій обчислити квадратичний поліном в чисельнику дробово-раціональної функції при значеннях вхідного бітового потоку  $x_1, x_2, \dots, x_{\max}$ , знайти значення арифметичного ряду 2-го порядку і арифметичних рядів різниць 1-го та 2-го порядків відповідно;

- на наступному етапі визначення початкових значень ініціалізації компонентів пристрою;

- виконання теоретичних розрахунків в основних елементах блоків 1, 2 обчислювача з урахуванням значень ініціалізації компонентів, записати у таблицю результати обчислень для заданих значень бітового потоку даних, а саме, для значень  $x_1, x_2, \dots, x_{\max}$ ;
- розробка граф-схеми алгоритму (ГСА) арифметичного блоку операційного автомата реалізації досліджуваної дробово-раціональної функції;
- здійснення вибору моделі кінцевого автомату, і на основі розробленої ГСА розробка графа переходів керуючого автомату обчислювача;
- розробка HDL-моделі досліджуваного обчислювача з використанням VHDL –шаблонів на основі автоматного опису пристрою;
- здійснення верифікації проекту за допомогою testBench та імплементації спроектованого пристрою в ПЛІС Xilinx.

#### 4.2 Теоретичні розрахунки в компонентах пристрою

При виконанні кваліфікаційної роботи була створена експериментальна апаратна модель обчислювача дробово-раціональної функції. Вхідний сигнал пристрою є бітовим потоком.

Дробово-раціональна функція, що досліджується має вигляд:

$$y = \left[ \frac{a_2 x^2 + a_1 x + a_0}{m} + 0,5 \right], \quad (4.1)$$

де  $a_2 = 1, a_1 = 2, a_0 = 5, m = 10$ .

Для наочної демонстрації прикладу роботи моделі обчислювача приведено результати експерименту дослідження функціонування пристрою.

інформаційний сигнал, який подається на вхід пристрою є біт-потоковою послідовністю  $x$ .

Вихідний інформаційний сигнал, що формується на виході обчислювача представляє біт-потоківу послідовність  $u$ .

Вихідний бітовий потік  $u$  відтворює дробово-раціональну функцію, що апроксимує безперервну з похибкою 0,5 половини одиниці молодшого розряду аргументу  $x$ .

Вихідні дані коефіцієнтів  $a_2$ ,  $a_1$ ,  $a_0$  та константи  $m$  функції (2.16) та довжини вхідного бітового потоку  $x_{\max}$  приведені у таблиці 4.1.

Таблиця 4.1 – Вихідні дані

$a_2$	$a_1$	$a_0$	$m$	$x_{\max}$
1	2	5	10	10

Вхідні та вихідні сигнали досліджуваного пристрою обчислення дробово-раціональної функції наведено відповідно у таблицях 4.2 і 4.3.

Таблиця 4.2 - Вхідні сигнали обчислювача

Входи	Розрядність	Опис
Reset	1 біт	Установка в початковий стан пристрою
Clk	1 біт	Сигнал синхронізації
Data	1 біт	Вхідний бітовий потік даних

Таблиця 4.3 - Вихідні сигнали обчислювача

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

Обчислення досліджуваної функції є необхідним підготовчим етапом при проведенні експерименту.

У відповідності до вихідних даних функція представлена із заданими коефіцієнтами

$$y = \left[ \frac{x^2 + 2x + 5}{10} + 0,5 \right], \quad (4.2)$$

I. Обчислення значень функції, обчислення поліному, ініціалізація компонентів досліджуваного пристрою для проведення експерименту.

Обчислення поліному другого порядку.

В чисельнику функції (4.2) записаний поліном другого порядку  $y = x^2 + 2x + 5$ . Він реалізований у в блоку квадратора пристрою.

Для поліному  $y = x^2 + 2x + 5$  необхідно порахувати значення функції для значень  $x = 0, 1, 2, 3, \dots, 10$ . При обчисленні отримано арифметичний ряд 2-го порядку, що є послідовністю значень функції  $y$ :

$$Y: 5, 8, 13, 20, 29, 40, 53, 68, 85, 104.$$

Для визначеної послідовності арифметичні ряди різниць 1-го і 2-го порядків мають вигляд:

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

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

Довжина вхідного бітового потоку, що поступає на вхід пристрою дорівнює  $x_{\max} = 10$ .

Результати обчислення заданої функції наведено для перших десяти значень аргументу вхідного бітового потоку:  $x=1, 2, \dots, 10$ .

Дробово-раціональна функція приймає значення з урахуванням округлення результату обчислення (таблиця 4.4).

Таблиця 4.4 Результати обчислення дробово-раціональної функції

Значення аргументу X	Значення функції Y
$x = 1$	$y = \left[ \frac{1^2 + 2 \cdot 1 + 5}{10} + 0,5 \right] = [1,3] = 1$

$x = 2$	$y = \left[ \frac{2^2 + 2 \cdot 2 + 5}{10} + 0,5 \right] = [1,8] = 1$
$x = 3$	$y = \left[ \frac{3^2 + 2 \cdot 3 + 5}{10} + 0,5 \right] = [2,5] = 2$
$x = 4$	$y = \left[ \frac{4^2 + 2 \cdot 4 + 5}{10} + 0,5 \right] = [3,4] = 3$
$x = 5$	$y = \left[ \frac{5^2 + 2 \cdot 5 + 5}{10} + 0,5 \right] = [4,5] = 4$
$x = 6$	$y = \left[ \frac{6^2 + 2 \cdot 6 + 5}{10} + 0,5 \right] = [5,8] = 5$
$x = 7$	$y = \left[ \frac{7^2 + 2 \cdot 7 + 5}{10} + 0,5 \right] = [7,3] = 7$
$x = 8$	$y = \left[ \frac{7^2 + 2 \cdot 7 + 5}{10} + 0,5 \right] = [7,3] = 7$
$x = 9$	$y = \left[ \frac{9^2 + 2 \cdot 9 + 5}{10} + 0,5 \right] = [10,9] = 10$
$x = 10$	$y = \left[ \frac{10^2 + 2 \cdot 10 + 5}{10} + 0,5 \right] = [12,5] = 12$

Ініціалізація компонентів. Перед подачею бітового потоку на вхід обчислювача компоненти пристрою мають бути ініціалізовані початковими значеннями, які необхідно визначити.

Числа ініціалізації, якими мають бути ініціалізовані компоненти архітектури наступні:

$$\text{SUM1} \quad a_2 + a_1 = 1 + 2 = 3;$$

$$\text{SUM2} \quad -m + a_0 = -10 + 5 = -5;$$

$$\text{RG1} \quad 2\mathbf{a}_2 = 2\mathbf{4} = 2;$$

$$\text{RG2} \quad -2 \cdot m = -2 \cdot 10 = -20.$$

При апаратній реалізації було створено утиліту для обчислення полінома другого порядку, отримано результати розрахунку (лістинг 4.1), що підтвердили обчислення значення функції, яка є арифметичним рядом 2-го порядку.

Арифметичні ряди різниць 1-го та 2-го порядків, а також числа, якими ініціалізуються компоненти пристрою приведено нижче.

#### Лістинг 4.1 Результати обчислення поліноміальної функції

$$y = x^2 + 2x + 5$$

$y = a \cdot x^2 + b \cdot x + c$

Enter a: 1

Enter b: 2

Enter c: 5

Begin calculate  $y=f(x)$  from (0 by default): 0

Number of calculated  $y=f(x)$  (10 by default): 10

$y=(1) \cdot x^2+(2) \cdot x+(5)$  where  $x$  is [0, 10]

$y = [5, 8, 13, 20, 29, 40, 53, 68, 85, 104]$

$\text{delta1} = [3, 5, 7, 9, 11, 13, 15, 17, 19]$

$\text{delta2} = [2, 2, 2, 2, 2, 2, 2, 2]$

entity kvadrator:

COUNTER\_INITIAL:=3

COUNTER\_STEP:=2

SUMMATOR\_INITIAL:=5

component counter:

STEP:=2

INITIAL:=3

component summator:

INITIAL:=5

II. Результати обчислень, які відбуваються у компонентах дробово-раціонального обчислювача.

На вхід обчислювача подається бітовий потік довжиною 10 біт  $x_{\max} = 10$ .

У компонентах пристрою відбуваються операції підсумовування, зсуву, інкременту, перенесення з регистрів RG1 і RG2 констант, що видно із таблиці 4.5.

Процес обчислень показано в основних компонентах модуля: SUM2, SUM1.

В результатах обчислення в SUM2 враховано виконання операції перенесення чисел, що підсумовуються в SUM1 в SUM2 із зсувом на 1 розряд вправо.

Результати обчислень у компонентах співпадають з результатами розрахунку значень функції (4.2) при підстановки у формулу по всіх вхідних бітах  $x$ .

Таблиця 4.5 – Результати обчислювального процесу у компонентах

$x$	SUM2	Біт переповнення SUM2	SUM1
1	$-5 + 6 = 1$ $1 - 20 = -19$	1	$3+2=5$
2	$-19 + 10 = -9$		$5+2 = 7$
3	$-9 + 14 = 5$ $5 - 20 = -15$	1	$7+2 = 9$
4	$-15 + 18 = 3$ $3 - 20 = -17$	1	$9+2=11$
5	$-17 + 22 = 5$ $5 - 20 = -15$	1	$11+2=13$
6	$-15 + 26 = 11$ $11 - 20 = -9$	1	$13+2 = 15$
7	$-9 + 30 = 21$ $21 - 20 = 1$ $1 - 20 = -19$	1 1	$15+2=17$
8	$-19 + 34 = 15$ $15 - 20 = -5$	1	$17+2=19$
9	$-5 + 38 = 33$ $33 - 20 = 13$ $13 - 20 = -7$	1 1	$19+2=21$
10	$-7 + 42 = 35$ $35 - 20 = 15$ $15 - 20 = -5$	1 1	$21+2=23$

4.3 Граф-схема алгоритму та граф переходів керуючого автомату пристрою

При розробці апаратної реалізації біт-потокowego обчислювача апаратну модель пристрою було представлено на основі цифрового автомату, який складається з керуючого та операційного автоматів.

Реалізація пристрою на цифровому автоматі дозволила забезпечити надійність роботи пристрою і чітко послідовне виконання обчислювальних операцій в його компонентах.

Операційний автомат має обчислювальні стани, в яких виконуються мікрооперації – перетворення даних, які ініціалізуються сигналами керування керуючого автомату [13].

Порядок слідування сигналів керування здійснює керуючий автомат. Крім того керуючий автомат визначає виконання мікрооперацій у певній послідовності на базі граф-схеми алгоритму.

Операційний автомат генерує і сигналів сповіщення при керуванні процесом.

В біт-потоквому обчислювачі арифметичний блок обчислювача реалізований на основі кінцевого автомату и є композицією операційного і керуючого автоматів.

ГСА операційного автомату обчислювача представляє собою орієнтований граф. У графі є початкова та кінцева вершини, він містить умовні, операторні вершини [14].

ГСА операційного автомату реалізації дробово-раціональних функцій побудовані на основі математичних моделей обчислювачів і містять операторні вершини з мікроопераціями, що відповідають обчислювальним процесам, які відбуваються в компонентах архітектури.

ГСА операційного автомату реалізації дробово-раціональної функції містить конвеєрні обчислення в операторних вершинах.

В ГСА операційного автомату пристрою відтворення заданої функції: при початковому скиданні пристрою сигналом «reset» усі регістри приводяться у початковий стан, тобто у встановлюються значення ініціалізації блоків обчислювача.

За математичною моделлю обчислювача та архітектурою пристрою було розроблено граф-схему алгоритму (ГСА), що описує принцип його роботи (рис. 4.1).

1 Модель пристрою працює за наступним алгоритмом при надходженні чергового біта (імпульсу) на вхід пристрою:

2а) пристрій приводять в початковий стан сигналом сигнал  $reset=1$ . При скиданні пристрою регістри встановлюються у значення ініціалізації:  $SUM1 = 3$ ,  $SUM2 = -5$ ,  $RG1=2$ ,  $RG2=-20$ ;

3б) виконуються паралельні операції: значення регістра  $SUM2$  збільшується на подвоєне значення вмісту  $SUM1$  при подачі біту вхідної послідовності  $x$  на вхід обчислювача, а значення регістра  $SUM1$  збільшується на значення  $RG1$ ;

в) якщо значення регістра  $SUM2$  невід'ємне,  $SUM\_RES \neq 0$ , то на виході обчислювача уде згенерований біт  $y$ .

В результаті буде операція віднімання: із значення регістра  $SUM\_RES$  буде відніматися значення регістра  $RG2$ , а саме, число у додатковому двійковому коді  $2^k - 2m$ .

г) наступна дія: пункт в) буде повторюватися поки вміст регістра  $SUM2$  є невід'ємним.

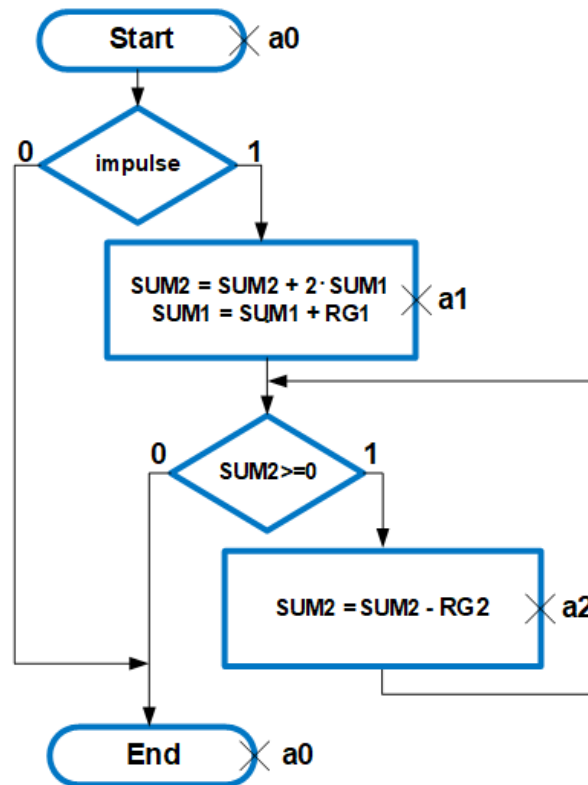


Рисунок 4.1 – Змістовна граф-схема алгоритму реалізації дробово-раціональної функції

Керуючий автомат арифметичного блоку обчислювача описується графом переходів, який розроблений на основі розмітки ГСА. Було обрано кінцевий автомат моделі Мура. Автомат цієї моделі є більш зручним по відношенню до автомата моделі Мілі для виконання обчислювальних операцій у обчислювачі, так як обчислювальні операції здійснюються у операторних вершинах, а не на переходах, як в автоматах Мілі.

Для керування обчислювальними операціями та при розмітці ГСА визначено три стани: стан  $a_0$  – початковий і кінцевий стан, та стани  $a_1$ ,  $a_2$  (рис. 4.2).

Керуючий автомат переходить у початковий стан  $a_0$  коли подається сигнал reset, що дорівнює «1» .

Автомат знаходиться у стані  $a_0$  до моменту появи сигналу impulse, який надходить із вхідного буфера пристрою. З приходом сигналу impulse автомат переходить в стан  $a_1$ .

У стані  $a1$  керуючий автомат додає до суми вмісту  $SUM2$  подвоєне значення регістру суматора  $SUM1$  із виконанням зсуву на 1 розряд вправо. У стані  $a1$  також значення регістра  $SUM1$  збільшується на значення регістра  $RG1$ , тобто на число 2.

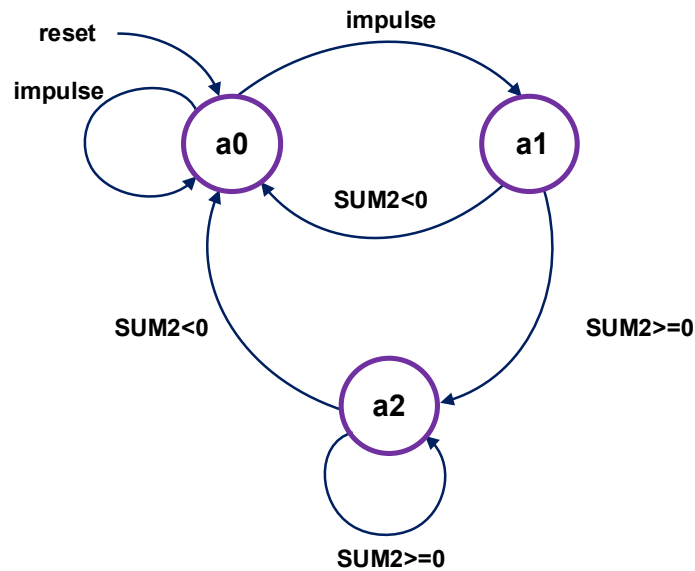


Рисунок 4.2 – Граф переходів автомата Мура арифметичного блоку  
обчислювача

Керуючий автомат перейде у стан наступний  $a2$ , якщо вміст регістру  $SUM2$  стане невід’ємним, тоді з’явиться біт переповнення на виході  $SUM2$  і автомат перейде у наступний стан  $a2$ , в іншому випадку автомат перейде у стан  $a0$ .

В стані  $a2$  від значення регістра  $SUM2$  віднімається значення регістра  $RG2$ , так як значення регістра  $RG2$  переноситься в регістр  $SUM2$  в додатковому коді, а саме,  $-20$ .

Також у стані  $a2$ , автомат видає сигнал вхідному буферу для формування біту на виході обчислювача.

#### 4.4 Структурно-блокова схема пристрою

При апаратній реалізації було створено структурно-блокову схему пристрою представлена на рис. 4.3.

Структурно-блокова схема містить блоки:

- блок детектору вхідного біту;
- блок спеціалізованого обчислювача;
- блок вихідного буферу.

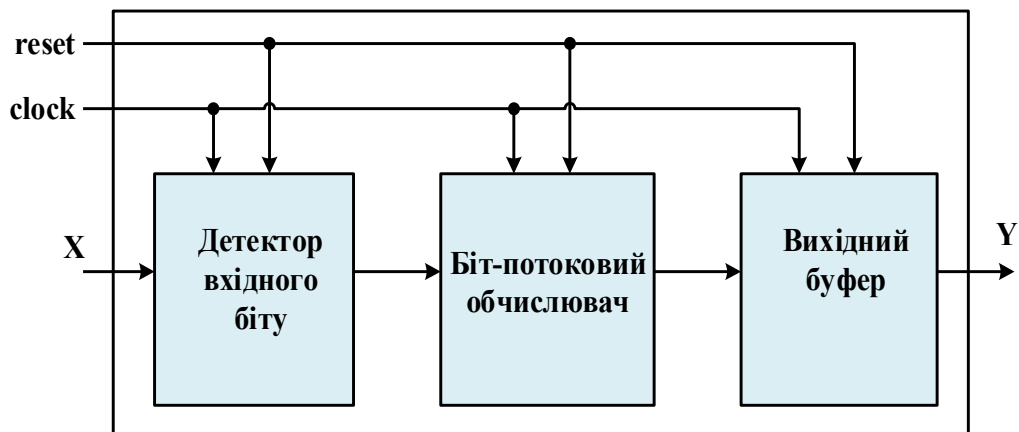


Рисунок 4.3 – Структурно-блокова схема обчислювача

Нижче розглянуто призначення блоків схеми обчислювача.

I. Детектор вхідного біту. Робота даного блоку полягає у детектуванні біт вхідного бітового потоку, що надходять на вхід пристрою. Вхідний сигнал (біт) може бути задетектований по двом тактам синхросигналу. Якщо, на одному такті, сигнал  $x$  має значення, що дорівнює логічному «0», а на другому такті змінює своє значення, та становиться логічною «1», тоді на виході детектору з'являється біт. Детектор встановлює сигнал  $impulse = 1$  на його виході. Саме цей сигнал отримує обчислювач для подальшої обробки.

На рис. 4.4 представлена часова діаграма детектування вхідного біту.

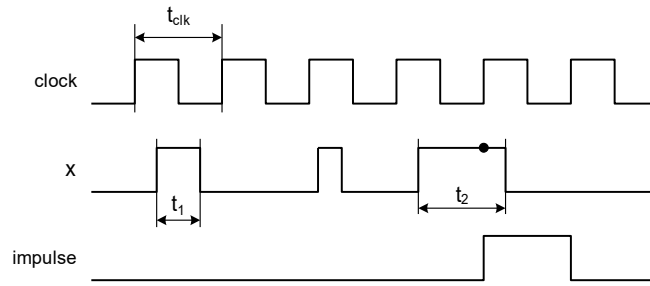


Рисунок 4.4 – Процес детектування біту

Потрібно, щоб період сигналу  $t_{clk}$  був у два рази меншим, ніж період вхідного біту. Якщо тривалість вхідного біту  $t_1$  менше періоду синхронізації, детектор не може відреагувати на вхідний біт, тобто задетектувати біт для яких  $t_1 < t_{clk}$ .

Таким чином, для виявлення вхідного біту необхідно, щоб період сигналу синхронізації  $t_{clk}$  був менший або дорівнював періоду вхідного біту. Якщо період вхідного біту  $t_1$  менше періоду синхронізації, такі імпульси не зможуть бути задетектовані детектором вхідного біту ( $t_1 < t_{clk}$ ).

У таблиці 4.6 наведено входи і виходи детектора структурно-блокової схеми.

Таблиця 4.6 – Вхідні і вихідні параметри детектору

Data	Вхідний бітовий потік	IN
Reset	Установка в початковий стан пристрою	IN
Clk	Сигнал синхронізації	IN
DataOut	Вихідний бітовий потік	OUT
Ready	Сигнал готовності з арифметичного блоку	IN

Фрагмент опису інтерфейсу детектора приведений у лістингу 4.2.

## Лістинг 4.2 – Інтерфейс блоку «Вхідний буфер».

```

entity aprox_inputbuffer is
  generic(
    width: natural := 8);
  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 aprox_inputbuffer;

```

II. Блок спеціалізованого пристрою – біт-потокowego обчислювача. Блок «Aproximator» містить обчислювач поліномів – квадратор та блок дільника чисел. Цей блок здійснює операції обчислення полінома другого порядку  $x^2 + 2x + 5$  та операція поділу результату обчислення поліному на число 10 с урахуванням абсолютної похибки обчислення.

Вхідні та вихідні сигнали блоку «Aproximator» наведено в таблиці 4.7.

Таблиця 4.7 – Вхідні та вихідні сигнали для блоку «Aproximator»

Data	Вхідний бітовий потік	IN
Reset	Установка пристрою в початковий стан	IN
Clock	Сигнал синхронізації	IN
Ready	Сигнал готовності пристрою прийняти на вхід наступний біт на обробку	OUT
DataOut	Вихідний бітовий потік, який є результатом обчислення функції $y = \left[ \frac{a_2 x^2 + a_1 x + a_0}{m} + 0,5 \right]$	OUT
Count	Значення функції в двійковому N-розрядному коді на виході суматора	OUT

В результаті роботи блоку біт-потокowego обчислювача на його виході генерується вихідний бітовий потік. Сигнал Ready, який з'являється на виході арифметичного блоку обчислювача, є сигналом готовності

арифметичного блоку сприйняти наступний біт на обчислювальну обробку. сигнал Ready дорівнює логічній «1». Результат обчислення функції .

Сигнал DataOut є результатом обчислення досліджуваної функції.

У лістингу 4.3. приведено фрагмент опису, що містить інтерфейс блоку «Aproximator» Розрядність даного компоненту обчислювача 8 біт.

Лістинг 4.3 – Інтерфейс блоку «Aproximator.vhdl».

```
entity aproximator is
  generic(
    width: natural := 8;
    a: integer := 1;
    b: integer := 2;
    c: integer := 5;
    m: integer := 10
  );
  port(
    x_i: in std_logic;
    ready_o: out 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 aproximator;
```

III. Блок схеми «Вихідний буфер». Вихідний буфер призначений, щоб формувати на виході обчислювача сигнал вихідного бітового потоку у.

Вхідні та вихідні параметри блоку «Вихідний буфер» описані у таблиці 4.8.

Таблиця 4.8 – Вхідні та вихідні параметри блоку

Enable	Сигнал дозволу	IN
Reset	Установка пристрою в початковий стан	IN
Clock	Сигнал синхронізації	IN
DataOut	Вихідний бітовий потік, який є результатом обчислення заданої функції	OUT

У лістингу 4.4. приведено опис інтерфейсу блоку «Вихідний буфер».

#### Лістинг 4.4 – Інтерфейс блоку «Вихідний буфер».

```
entity aprox_outputbuffer is
    port(
        clock_i: in std_logic;
        reset_i: in std_logic;
        en_i: in std_logic;
        y_o: out std_logic);
end aprox_outputbuffer;
```

#### 4.5 Опис проекту на МОА

Система Active-HDL є продуктом, що орієнтований на максимальну гнучкість розробки складних проектів багаторівневої ієрархії з урахуванням будь-яких типів сучасних ПЛІС різних фірм-виробників. Важливо те, що окрім розвинених вбудованих засобів моделювання та налагодження Active-HDL також надає розробникам єдине середовище для побудови власних маршрутів проектування ПЛІС залежно від характеру розв'язуваних завдань та програмних засобів. Програмний пакет Active-HDL виступає, як система проектування, а також моделювання проектів, що підтримують всі сімейства ПЛІС.

Використання [мов опису апаратури](#), зокрема, Active-HDL, а також структурних схем та математичних моделей пристроїв дає можливість здійснити розробку HDL-моделі дробово-раціонального обчислювача.

Графова модель обчислювача дробово-раціональних функцій керуючого автомату, що має всього три стани дозволяє здійснювати керування конвеєрними обчисленнями в операторних вершинах ГСА операційного автомату. HDL-модель обчислювача розроблена на основі автоматного програмування, в якому головним є поняття стану автомату.

Для того, щоб здійснити моделювання проекту та його верифікацію було обрано програмний пакет Active-HDL. Програмний пакет представляє виконувати автоматизацію процесу проектування пристрою і створити testbench, аналізувати результати, проводити верифікації створеного

проектного рішення. Щоб здійснити синтез проекту було обрано продукт Synplify. Продукт представляє собою інструмент синтезу RTL, що був розроблений для ПЛІС там включає у себе бібліотеки сучасної елементної бази.

Для написання проекту на МОА було обрано мову VHDL, яка дає можливість високої адаптації до того, що потребують цілі проекту. Мова VHDL має ефективне середовище моделювання, зокрема, Active-HDL, та середовище для синтезу проектів, яке використовує продукт VHDL, зокрема, Synplify.

При виконанні опису проекту мовою VHDL проект представлено у вигляді ієрархічної структури (рис. 4.5), в якій все більше деталізується опис проекту на нижніх рівнях, що спрощує процес написання HDL-моделей окремих компонентів проекту, а також їх налагодження.

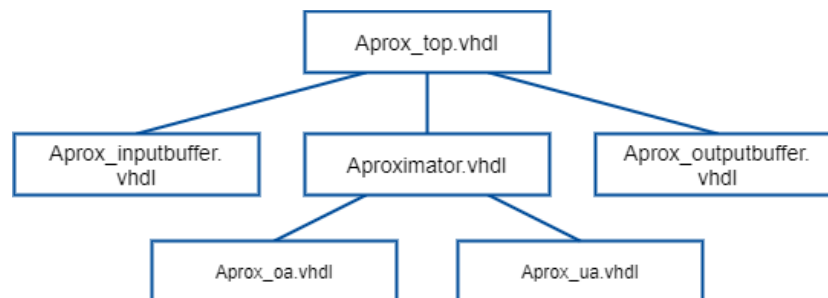


Рисунок 4.5 – Ієрархія проекту для експериментальної реалізації обчислювача

Проект складається з файлів, що мають ієрархічну структуру. Вони об'єднуються за допомогою top-файлу Aprox\_top.vhdl, що являє собою самий верхній рівень ієрархічної структури проекту. Даний файл містить компоненти, що пов'язані між собою операторами port map, що виконують зв'язок файлів нижнього рівня ієрархії. HDL-файл Aprox\_top.vhdl, що реалізований, є специфікацією всього пристрою. Він містить entity, в якому

дано опис всіх вхідних і вихідних сигналів розробленої апаратної моделі пристрою.

Інтерфейс файлу Aprox\_top.vhdl наведено у лістингу 4.5.

Лістинг 4.5 – Інтерфейс файлу верхнього рівня Aprox\_top.vhdl.

```
entity aprox_toplevel is
  generic(
    width: natural := 8;
    a: integer := 1;
    b: integer := 2;
    c: integer := 5;
    m: integer := 10);
  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 aprox_toplevel;
```

Компоненти другого рівня ієрархії наведено у лістингу 4.6: Aprox\_inputbuffer.vhdl, Aproximator.vhdl, Aprox\_outputbuffer.vhdl. У лістингу наведено фрагмент HDL-коду, що містить задекларовані змінні та їх значення.

Лістинг 4.6 – Опис компонентів другого рівня ієрархії проекту.

```
architecture struct of aprox_toplevel is
  -- Component declaration of the "aproximator(struct)" unit defined in
  -- file: "./src/aproximator.vhd"
  component aproximator
    generic(
      width : NATURAL := 8;
      a: integer := 1;
      b: integer := 2;
      c: integer := 5;
      m: integer := 10);
    port(
      x_i : in std_logic;
      ready_o : out 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 aprox_inputbuffer
  generic(
    width : NATURAL := 8);
  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;

-- Component declaration of the "outputbuffer(beh)" unit defined in
-- file: "./src/outputbuffer.vhd"
component aprox_outputbuffer
  port(
    clock_i : in std_logic;
    reset_i : in std_logic;
    en_i : in std_logic;
    y_o : out std_logic);
end component;

```

З'єднання блоків у файлі `Aprox_top.vhdl` описано за допомогою операторів `port map` (лістинг 4.7). Вони пов'язують файли другого рівня ієрархії проекту.

#### Лістинг 4.7 – Опис з'єднання блоків `Aprox_top.vhdl`

```

signal inputbuffer_out, aprox_out, ready, y_t: std_logic;
signal count_t: std_logic_vector(width-1 downto 0);
signal sum_t: std_logic_vector(width-1 downto 0);
begin
  y_o <= y_t;
  InputBuffer_1 : aprox_inputbuffer
  generic map(
    width => width
  )
  port map(
    x_i => x_i,
    clock_i => clock_i,
    reset_i => reset_i,
    ready_i => ready,
    y_o => inputbuffer_out
  );
  Aproximator_1 : aproximator
  generic map(
    width => width,
    a => a,

```

```

        b => b,
        c => c,
        m => m
    )
port map(
    x_i => inputbuffer_out,
    ready_o => ready,
    clock_i => clock_i,
    reset_i => reset_i,
    y_o => aprox_out,
    sum_o => sum_t
);
OutputBuffer_1 : aprox_outputbuffer
port map(
    clock_i => clock_i,
    reset_i => reset_i,
    en_i => aprox_out,
    y_o => y_t
);

```

Компонентами третього рівня ієрархії є компоненти керуючого та операційного автоматів `Aprox_ua.vhdl`, `Aprox_oa.vhdl` відповідно, які наведено у лістингу 4.8.

Лістинг 4.8 – Фрагмент HDL-коду опису компонентів третього рівня ієрархії проекту

```

architecture struct of aproximator is
    -- Component declaration of the "aproximator_ua(beh)" unit defined in
    -- file: "./src/aproximator_ua.vhd"
    component aproximator_ua
        port(
            clock_i : in std_logic;
            reset_i : in std_logic;
            x_i : in std_logic;
            y_o : out std_logic;
            ready_o : out std_logic;
            sum_less_zero_i : in std_logic;
            sum_plus_a_o : out std_logic;
            sum_minus_b_o : out std_logic;
            states: out std_logic_vector(1 downto 0));
    end component;

    -- Component declaration of the "aproximator_oa(beh)" unit defined in
    -- file: "./src/aproximator_oa.vhd"
    component aproximator_oa
        generic(
            width : NATURAL := 8;
            a: integer := 1;
            b: integer := 2;
            c: integer := 5;

```

```

        m : integer := 10);
port(
    clock_i : in std_logic;
    reset_i : in std_logic;
    sum_plus_a_i : in std_logic;
    sum_minus_b_i : in std_logic;
    sum_less_zero_o : out std_logic;
    sum_o: out std_logic_vector(width-1 downto 0));
end component;

```

Обчислювальний процес, що відбувається в операційному автоматі блоку «Aprox\_oa.vhdl» , приведений у фрагменті HDL-коду (лістинг 4.9).

Лістинг 4.9 – Фрагменті HDL-коду, що описує роботу операційного автомату «Aprox\_oa.vhdl»

```

process (clock_i, reset_i)
begin
    if (reset_i = '1') then
        sum <= CONV_STD_LOGIC_VECTOR(c - m, width);
        counter <= CONV_STD_LOGIC_VECTOR(a+b, width);
    else
        if (falling_edge(clock_i)) then
            if (sum_plus_a_i = '1') then
                sum <= sum + counter + counter;
                counter <= counter + 2*a;
            else
                if (sum_minus_b_i = '1') then
                    sum <= sum - 2*m;
                end if;
            end if;
        end if;
    end if;
end process;

```

Фрагмент HDL-коду, що описує роботу керуючого автомату за допомогою оператора case приведений у лістингу 4.10.

Лістинг 4.10. Фрагмент HDL-коду, що описує роботу керуючого автомату

```

process(state, x_i, sum_less_zero_i)
begin
    case (state) is
        when a_0 =>
            if x_i = '1' then
                next_state <= a_1;
            end if;
    end case;
end process;

```

```

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

```

#### 4.5 Верифікація проекту

Після розробки HDL-моделей обчислювача з метою верифікації апаратної моделі біт-потокowego обчислювача дробово-раціональних функцій було розроблено тестове оточення (TestBench). TestBench представляє собою модуль, написаний мовою опису апаратури VHDL.

У модулі відсутній інтерфейс, але TestBench містить примірник тестового модуля, що подає тестові впливи та аналізує відповідні реакції тестованого пристрою. У Додатку Б приведено лістинг програмного коду модуля TestBench.

На рисунку 4.6 показано взаємодію TestBench та тестованого модуля.

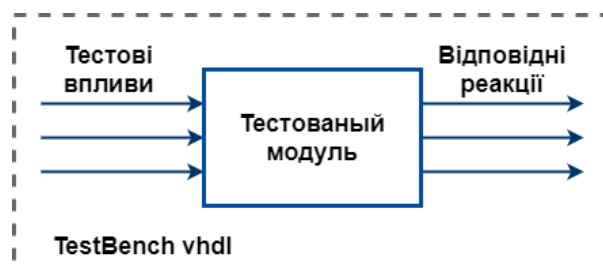


Рисунок 4.6 – Відношення TestBench і тестованого модуля

На рис. 4.7 представлено результати моделювання поведінкової моделі досліджуваного пристрою. Часова діаграма показує, що результати тестування поведінкової моделі співпадають з теоретичними розрахунками експериментальних даних. Результати тестування моделі обчислювача підтверджують правильність його функціонування.

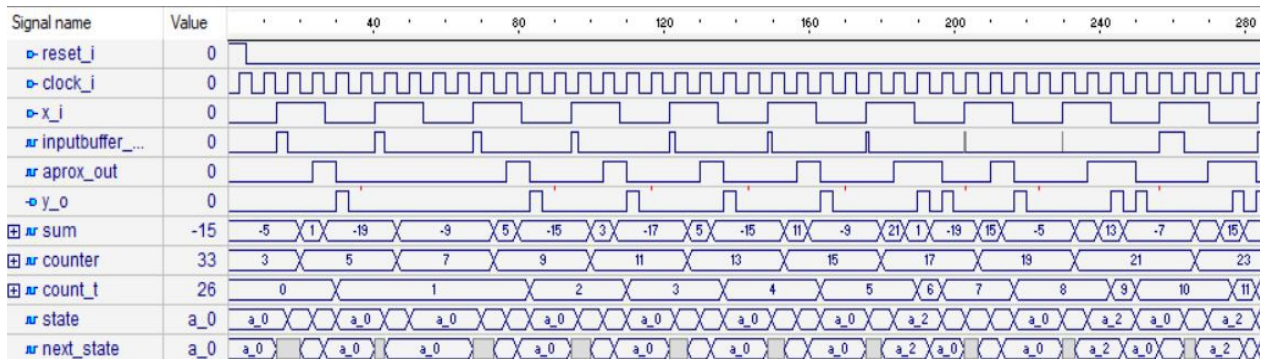


Рисунок 4.7 – Часова діаграма моделювання поведінкової моделі досліджуваного обчислювача

На рис. 4.8 зображено часову діаграму, на якій показано числа, що з'являються при роботі компонентів пристрою.

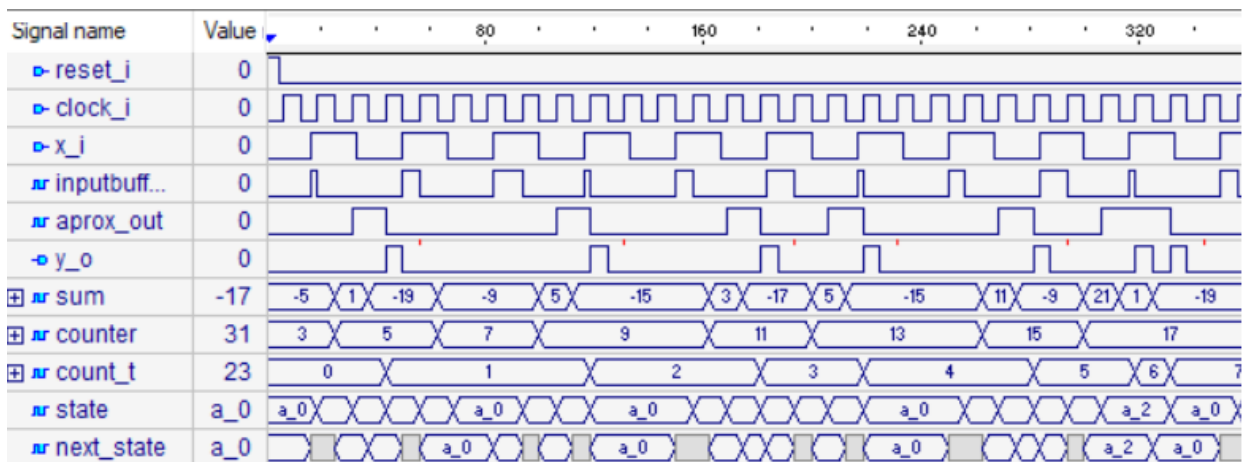


Рисунок 4.8 – Результати обчислень у компонентах апаратної моделі

Отримані значення у регістрах компонентів такі ж самі, як у результатах розрахунків, що представлені у таблиці 4.5, тобто результати експерименту співпадають з результатами теоретичних обчислень.

Після перевірки режимів верифікації досліджуваного пристрою було виконано наступний етап проектування – імплементацию моделі пристрою в ПЛІС та тестування після імплементации.

При апаратній реалізації для імплементации розробленої експериментальної моделі використаний кристал Xilinx Spartan 3E.

При синтезі проекту використано систему Synplify 7.0 Pro.

Як результат синтезу було отримано RTL схеми вентильного рівня, схема Technology View і файл звіту.

У лістингу 4.11 приведено UCF - файл, що розроблений для імплементации апаратної моделі у ПЛІС.

#### Лістинг 4.11 – UCF- файл

```
NET "rst" LOC = "H13" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "pack" LOC = "V4" | 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 ;
```

Приведено характеристики кристалу, на основі якого виконувався синтез:

```
Target Device : xc3s500e
Target Package : cp132
Target Speed : -5
```

В роботі наведено звіт про мінімальний період сигналу синхронізації та максимальну частоту сигналу:

Minimum period: 10.647ns (Maximum Frequency: 93.921MHz).

Приведений звіт про використання ресурсів кристалу:

Design Summary Report:

Selected Device : 3s500efg320-5

Number of Slices:	65 out of 4656	1%
Number of Slice Flip Flops:	91 out of 9312	0%
Number of 4 input LUTs:	122 out of 9312	1%
Number of IOs:	11	
Number of bonded IOBs:	11 out of 232	4%
Number of GCLKs:	1 out of 24	4%

Звіт містить дані, які показують, що при синтезі апаратної моделі у ПЛІС Xilinx Spartan при застосуванні серії xc3s500e використано близько 1% ресурсів, що пов'язані з логічними елементами схеми обчислювача.

Таким чином, розроблена апаратна модель пристрою є економічною відносно використаних ресурсів і може дозволити розмістити до 25 подібних обчислювачів.

На рис. 4.9 - 4.12 наведено RTL схема пристрою, RTL схема wrapper, а також наведено RTL схему арифметичного блоку на основі цифрового автомату: операційного та керуючого.

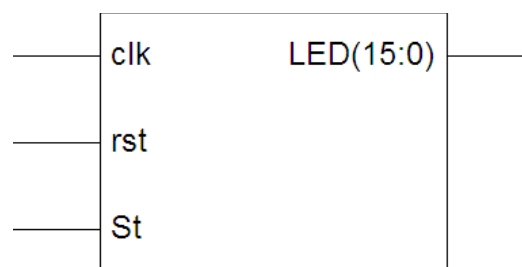


Рисунок 4.9 – RTL схема пристрою

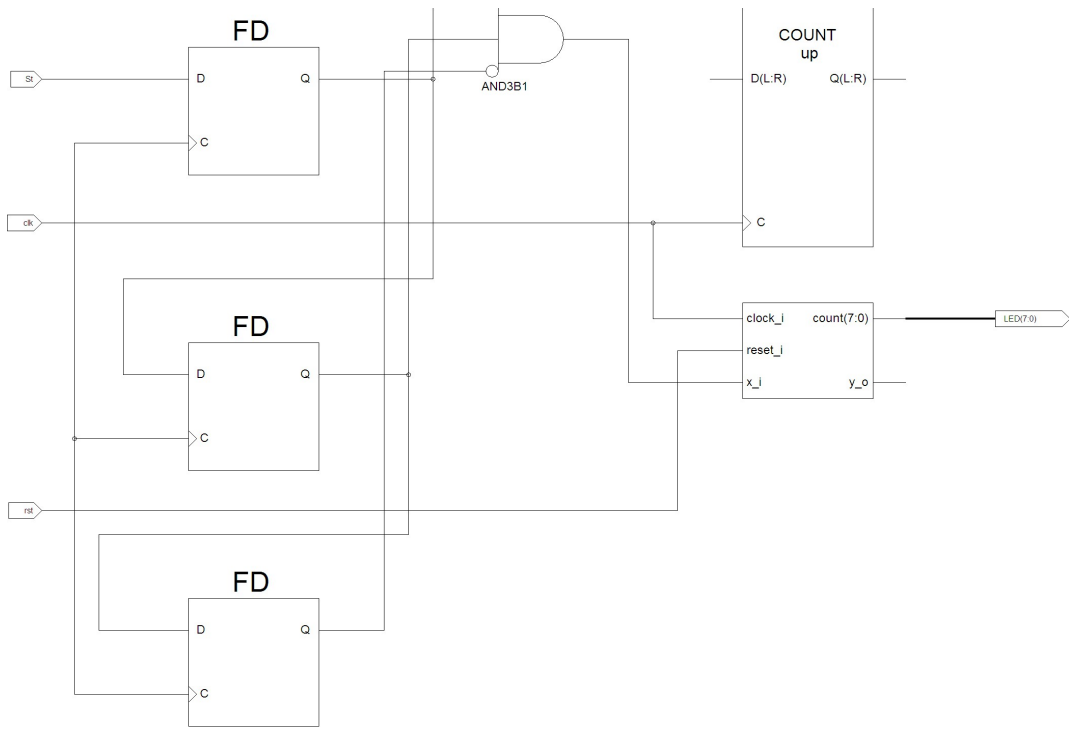


Рисунок 4.10 – RTL схема wrapper

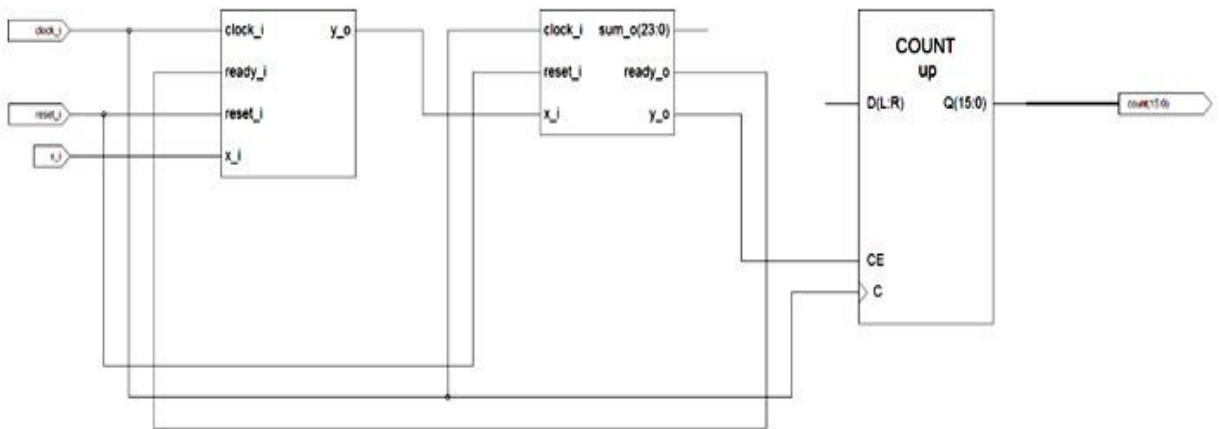


Рисунок 4.11 – RTL схема пристрою нижнього рівня

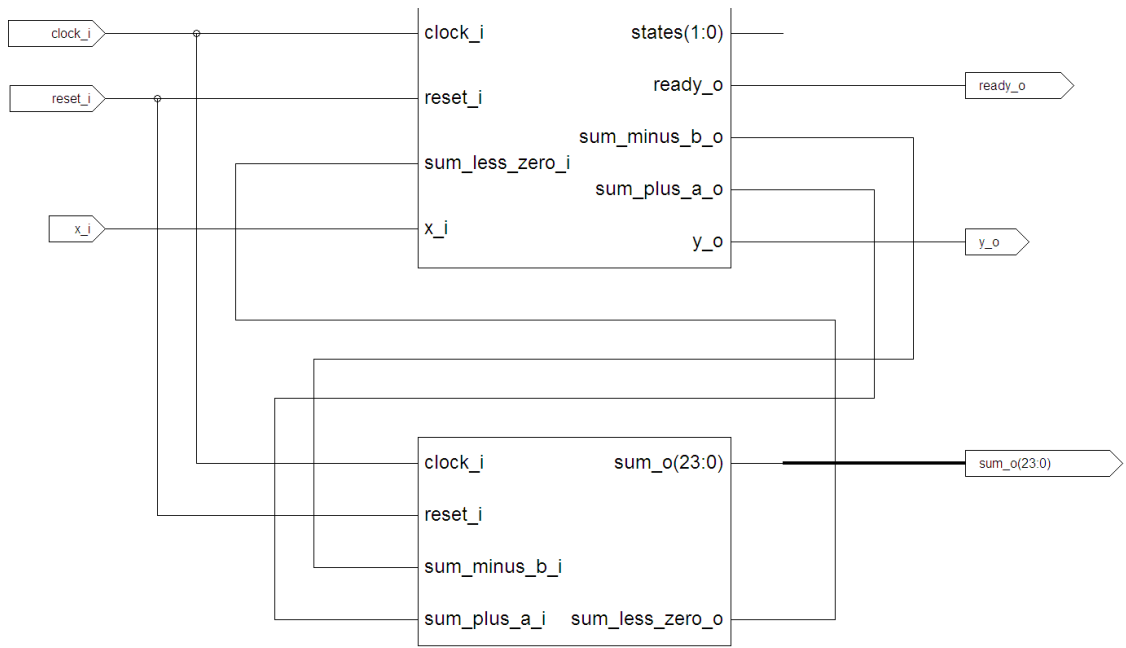


Рисунок 4.12 – RTL схема блоку на основі цифрового автомату

## ВИСНОВКИ

Кваліфікаційна робота присвячена проектуванню та дослідженню обчислювача дробово-раціональних функцій з біт-поточною формою подання аргументу.

В даний час широке застосування знаходить завдання створення розподілених систем управління, що можуть обробляти данні від великої кількості різномірних сенсорів фізичної природи. При цьому актуальним завданням є розробка зовнішніх модулів потокової обробки даних, що включають вторинні функціональні перетворювачі та обчислювачі для вирішення завдань лінеаризації сигналів.

Основною метою розвитку архітектур біт-поточкових обчислювачів є спрощення взаємодії між компонентами, шляхом використання бітових потоків для послідовної передачі даних. Важливою вимогою до таких пристроїв є необхідність виконання обробки бітових потоків даних в реальному масштабі часу, що накладає жорсткі обмеження на час вирішення завдань.

На основі аналізу способу формування приростів ступінчастих елементарних функцій отримано математичну модель досліджуваного обчислювача дробово-раціональної функції з мінімальною похибкою обчислення, що представляє собою систему різницевих нерівностей та яка покладена в основу роботи пристрою.

Проведено синтез архітектури біт-поточкового обчислювача дробово-раціональних функцій з використанням базової конвеєрної структури, що забезпечує високу швидкодію пристрою на її основі. Архітектура пристрою є синтезом структур біт-поточкового конвеєрного квадратора та дільника чисел. Основним обчислювальним компонентом в архітектурі пристрою є паралельний нагромаджуючий суматор зі зворотним зв'язком, в якому

здійснюється порівняння приростів двох одночасно відтворюваних висхідних ступінчастих функцій математичної моделі.

В результаті проведеного дослідження була створена експериментальна апаратна реалізація досліджуваної моделі обчислювача, виконані необхідні розрахунки. Обчислювач реалізовано на цифровому автоматі моделі Мура, що забезпечило надійність роботи пристрою. Розроблено граф-схему алгоритму та граф переходів керуючого автомату.

Було здійснено опис проекту для введення в САПР. За граф-схемою алгоритму та графом переходів з використанням шаблонів коду, було розроблено модель пристрою мовою опису апаратури VHDL.

Для верифікації проекту було розроблено test-bench та отримано поведінкову модель обчислювача. Показано, що результати моделювання поведінкової моделі досліджуваного пристрою співпадають з результатами обчислень в його компонентах. Модель синтезована засобами САПР Xilinx. В якості платформи для імплементації пристрою обрано кристал типу FPGA Spartan 3E серії XC3S500E.

Біт-потоківий обчислювач дробово-раціональних функцій може бути застосований в системах управління, контролю, інформаційно вимірювальних системах в якості функціонального перетворювача частотних сигналів, отриманих з датчиків фізичних величин, що потребують здійснення функціональної обробки, а також при розв'язанні задач управління при впливі на виконавчі механізми роботів, маніпуляторів та біонічних протезів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Dhafer Al-Makhles. Bitstream control system: Stability and experimental application / Dhafer Al-Makhles, Nitish Patel, Akshya Swain. // Intern. Conf. on Appl. Electronics. Czech Republic, Pilsen, 2013. P. 1–6.
2. Bureneva O. Hardware Module for Pulse Interference Mitigation in Digital Signal Streams / O. Bureneva, D. Sobolev, A. Pustovoitova // 2021 10th Mediterranean Conference on Embedded Computing (MECO), 2021, pp. 1-4.
3. Буренева О.И. Бит-потокное устройство извлечения квадратного корня / О.И. Буренева, О.А. Жирнова // Известия ЛЭТИ № 2, – 2019. С. 26 – 32.
4. A.I. Gulin, N.M. Assurance of Fault-Tolerance in Bit-Stream Computing Converters. / A.I. Gulin, N.M. Safyannikov, O.I. Bureneva, A.Yu. Kaydanovich // Proceeding of 16th IEEE East-West Design & Test Symposium (EWDTS'2018). Kazan, Russia, September 14 – 17, 2018. – pp. 418 – 421.
5. Стахів М.Ю. Автореф. дисертації. Цифрові функціональні перетворювачі розгортуючого типу з покращеними характеристиками // Поліграф. Центр Видавництва Національного університету "Львівська політехніка" – 2013. – 22 С.
6. Сафьянников Н.М. Следящий потокный вычислительный преобразователь для интеллектуальных измерительных систем/Сафьянников Н.М., Буренева О.И. // Международная конференция по мягким вычислениям. 2019. Т.1. С 263-266.
7. Попов А. Ю. Проектирование цифровых устройств с использованием ПЛИС: учебное пособие /А. Ю Попов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. – 80 с.
8. Строгонов А.В. Системное проектирование программируемых логических интегральных схем: Учебное пособие / А.В. Строгонов. –

Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет». – 2012. – 322 с.

9. 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.

10. Ларченко Б.Д. Декомпозиція математичної моделі біт-потокowego обчислювача ірраціональних функцій / О.С. Шкіль, Б.Д. Ларченко, Л.В. Ларченко // Радіоелектроніка та інформатика. №4. – 2019. – С 34-39.

11. Ларченко Б.Д. Апаратний online обчислювач дробово – раціональних функцій з бітовим потоком даних. / Ларченко Б.Д., Ющенко С.В. // Актуальні проблеми сучасної науки, LIX Міжнародна науково-практична інтернет-конференція. – м. Дніпро, 18 січня, 2021. – С. 156 – 160.

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

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

14. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов / Е.П. Угрюмов. – Изд. 3-е, перераб. и доп. – СПб.: БХВ-Петербург. – 2010. – 816 с.