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____перший (бакалаврський)_____
(рівень вищої освіти)

Апаратний потоковий обчислювач функції добування кореня

(тема)

Виконав:
здобувач ____4_____ року навчання,
групи КІУКІ-21-8

_____Красіля М.М._____
(прізвище, ініціали)

Спеціальність *123 _Комп'ютерна інженерія*

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

Освітня програма_____

_____*Комп'ютерна інженерія* _____
(повна назва освітньої прграми)

Керівник _ас. Безродний В.В._
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _Чумаченко С.В
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерної інженерії та управління _____

Кафедра_____Автоматизації проектування обчислювальної техніки_____

Рівень вищої освіти____перший (бакалаврський)_____

Спеціальність_____123_Комп'ютерна інженерія _____
(шифр і назва)

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

Освітня програма _____Комп'ютерна інженерія _____
(повна назва

ЗАТВЕРДЖУЮ:

Зав. кафедри ___  _____
(підпис)

«__06__»___05_____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту_____Красілі Максиму Миколайовчу_____
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Апаратний потоковий обчислювач функції добування кореня_____
_____
затверджена наказом по університету від від " 21 "_____05_____2025 р. № 403 Ст_____

2. Термін подання студентом роботи до екзаменаційної комісії _____19.06.2025_____

3. Вихідні дані до роботи (проекту)_____

ПЛІС Xilinx Spartan-3E

САПР Active HDL, XILINX ISE

Мова опису апаратури VHDL

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

Апаратні потокові обчислювачі для систем управління

Аналіз математичної моделі пристрою, розробка його архітектурної моделі

Розробка автоматої моделі пристрою, створення HDL-моделі

Апаратна реалізація спроектованого обислювача, імплементація в ПЛІС

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) __14__ слайдів_____

_____

_____

_____

_____

6. Консультанти розділів роботи (проекту)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження теми	06.05.2024 -07.05.2024	
2	Аналіз проблемної галузі, постановка задачі	07.05.2024 -10.05.2024	
3	Аналіз математичної моделі та розробка архітектурної моделі обчислювача	10.05.2024 -17.05.2024	
4	Розробка автоматної моделі пристрою на основі кінцевого автомату, вібір платформи ПЛІС	18.05.2024 -25.05.2024	
5	Розробка HDL-проекту з використанням HDL-шаблонів	25.05.2024 -30.05.2024	
6	Розробка апаратної реалізації обчислювача.	31.05.2024 -05.06.204	
7	Оформлення пояснювальної записки	05.06.2024 -10.06.2024	
8	Перевірка виконаного проекту керівником,	10.06.2024 -15.06.2024	
9	Захист проекту	15.06.2024 -24.06.2024	

Дата видачі завдання 06.05.2025_____

Студент _____
(підпис)

Керівник роботи (проекту) _____ _____ас. Безродний В.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 51 сторінок, 16 рисунків, 10 джерел за переліком посилань.

БІТОВИЙ ПОТІК ДАНИХ, ФУНКЦІОНАЛЬНЕ ПЕРЕТВОРЕННЯ, АРХІТЕКТУРА, КІНЦЕВИЙ АВТОМАТ, ГРАФ ПЕРЕХОДІВ, ГРАФ-СХЕМА АЛГОРИТМУ, HDL-МОДЕЛЬ

Метою кваліфікаційної роботи є розроблення апаратного потокового обчислювача функції добування кореня на платформі ПЛІС з використанням мов опису апаратури.

У ході виконання роботи розглянуто особливості апаратних потокових обчислювачів в системах управління реального часу, етапи проектування потокових функціональних обчислювачів, проаналізовано метод ступінчастой апроксимації на основі обернених функцій та математичну модель пристрою, алгоритм конвеєрних обчислень. Розроблено архітектурну модель обчислювача на основі базової архітектури біт-потокового обчислювача. Для апаратної реалізації розроблено автоматну модель пристрою на основі кінцевого автомату: граф-схему алгоритму обчислення функції та граф переходів керуючого автомату обчислювача і на їх підставі розроблено HDL-описи пристрою. Здійснено верифікацію проекту та імплементацію пристрою в ПЛІС Xilinx Spartan.

ABSTRACT

Explanatory note: 51 pages, 16 figures, 10 sources according to the list of links.

FUNCTIONAL CONVERSION, BIT-STREAM DATA, BIT-STREAM COMPUTING, APROXIMATION, ARCHITECTURE, FINITE STATE MACHINE, GRAPH DIAGRAM, VHDL-MODEL, FPGA

The purpose of the qualification work is to develop a hardware streaming computer for the root extraction function on the FPGA platform.

During the work, the features of hardware streaming computers in control systems, the stages of designing streaming functional computers, the mathematical model of the device and the algorithm of pipeline calculations were analyzed. An architectural model of the computer was developed based on the basic architecture of the bit-stream root extraction device. For hardware implementation, an automaton model of the device was developed based on a finite state machine: a flowchart of the function calculation algorithm and a transition graph of the computer's control automaton, and an HDL description of the device was developed based on them. The project was verified and the device was implemented in the Xilinx Spartan FPGA.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВС – вбудована система;

ГСА – граф-схема алгоритму;

ЕОМ – електронно-обчислювальна машина;

МОА – мова опису апаратури;

НВІС – надвелика інтегральна схема;

ОУ – об’єкт управління;

ПЛІС – програмована логічна інтегральна схема;

САПР – система автоматизації проектування;

СУ – система управління;

ЦОС – цифрова обробка сигналів;

HDL – Hardware Description Language (мова опису апаратури інтегральних схем);

RTL – Register Transfer Level (рівень регістрових передач);

FPGA – field programmable gate arrays – програмована користувачем вентильна матриця (різновид ПЛІС);

VHDL – very high speed integrated circuits HDL (мова опису апаратури);

SoC – System on Chip (система на кристалі).

ВСТУП

У розподілених системах управління, контролю та сенсорних системах реального часу необхідно забезпечувати безперервний прийом та оброблення потоків інформаційних даних у міру їх надходження. Одним із напрямків розроблення систем управління є створення та удосконалення обчислювальних засобів, які знаходять широке використання при розв'язання задач, що пов'язані із перетворенням інформаційних сигналів, що приймають від значного числа різнорідних сенсорів. Структура розподілених систем містить множину апаратних обчислювачів функціонального перетворення для виконання лінеаризації сигналів за необхідною функцією, які можуть бути реалізовані за допомогою програмних засобів або за спеціалізованх апаратних функціональних засобів. Тому тема кваліфікаційної роботи є актуальною.

Метою кваліфікаційної роботи є розробка апаратного обчислювача функції добування кореня, реалізований на платформі ПЛІС, на вхід якого подається бітовий потік даних, що представляє собою потік імпульсів. Робота спрямована на розробку математичної, архітектурної та автоматної моделей проектування обчислювача функції вилучення кореня. Застосування математичної моделі потокового обчислювача дозволяє мінімізувати абсолютну похибку обчислень, підвищити ефективність проектування пристрою за рахунок використання розроблених HDL-шаблонів опису моделі обчислювача.

Спроектований апаратний обчислювач може знайти застосування в якості функціонального обчислювального пристрою real-time перетворення бітових потоків в системах управління, у вимірювальних системах при вимірюванні витрати рідини на основі перепаду тиску, так як дані величини пов'язані квадратичною залежністю.

1 АПАРАТНІ ПОТОКОВІ ОБЧИСЛЮВАЧІ ДЛЯ СИСТЕМ УПРАВЛІННЯ

У розділі розглянуто призначення та застосування апаратних потокових обчислювачів математичних функцій в системах управління дано поняття бітового потоку даних, сформульовано мету та задачі проектування.

1.1 Призначення та застосування апаратних потокових обчислювачів математичних функцій в системах управління

В розподілених системах управління (СУ) та інтелектуальних вимірювальних системах (ІВС) потокові обчислювачі та перетворювачі є засобом функціональної, часто нелінійної обробки цифрової інформації, а також можуть бути застосовані як периферійні процесори функціонального розширення обчислювальних систем.

В СУ реального часу інформаційні сигнали отримують від різних сенсорів, що сприймають фізичні величини. Сенсори видають сигнали в частотній або імпульсній формі, якщо перетворення аналогових сигналів сенсорним компонентом здійснюється у частоту. Частотні або імпульсні вихідні сенсорні сигнали є результатами вимірювань, вони призначені для обробки за певними функціями, в результаті обробки яких реалізацуються задачи управління. При цьому необхідним є виконання різних нелінійних перетворень частотних або імпульсних сигналів, що мають назву бітових потоків. Над бітовими потоками здійснюють перетворення за допомогою функціональних перетворювачів та апаратних обчислювачів. Значно важливою вимогою до подібних пристроїв є потреба у виконані оброблення сигналів в реальному часі, тому час виконання функціональних задач є дуже важливим.

Оскільки процес перетворення сигналів необхідно здійснювати в реальному часі, то основною метою удосконалення структур апаратних біт-

потокових функціональних обчислювачів, що виконують потокову обробку, є спрощення взаємодії блоків та компонентів структури, а отже спрощення аррхітектур подібних обчислювачів. Досягнути спрощення архітектур обчислювачів можна за рахунок процесу потокової обробки сигналів, що являють собою біт-потокову форму, Крім того, спростити архітектуру можна за рахунок удосконалення математичних моделей пристроїв, можливості реконфігурації архітектур та їх універсалізма з точки зору використання тих же архітектур обчислення інщих елементарних математичних функцій.

Розподілені комп'ютерні СУ мають у своєму складі інформаційно-вимірювальну підсистему обробки потокової інформації, яка містить сенсори та функціональні перетворювачі або обчислювачі, що працюють з потоковими формами сигналів. СУ складаються з підсистеми обробки інформаційних сигналів, об'єкту управління (ОУ), сенсорів та перетворювачів, також виконавчих пристроїв, що керують ОУ (рис.1.1).

Апаратні потокові обчислювачі математичних функцій, що застосовуються у системах управління, контролю та вимірювань призначені для функціонального перетворення інформаційних сигналів сенсорних компонентів, а також для виконання інтерполяції сигналів при вирішенні траєкторних завдань.

Рисунок 1.1 – СУ реального часу з компонентами та зв'язками

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

1) витратомірні сенсори: сенсор витрати рідини або датчик турбінного типу, що використовується у приладах для відстеження функцій дихання у медицині;

2) сенсори іонізуючих випромінювань;

3) оптоелектронні сенсори кутового або лінійного переміщення.

Сигнал, що впливає на процес або об'єкт може бути представлений імпульсною послідовністю. Прикладами можуть бути електростимуляція нервово-м'язової системи опорно-рухового апарату, електродіагностика у галузі медицини, сигнали стимуляції та збудження у відновленні тканинами функцій. Зміна стану процесу або об'єкту може здійснюватися регулюванням потужності електроенергії, що є сигналом управління, який перетворюється за лопомогою пристроїв перетворення, також може бути завданням формування імпульсних послідовностей або однієї послідовності із заданими параметрами:

– частотний сигнал використовується в потокових перетворювачах постійного струму; та у перетворювачах частоти та змінної напруги;

– кількість імпульсів (бітів) для систем управління кроковими двигунами.

В частотних або імпульсних сигналах представлення інформації здійснюється безперервно у часі та перетворюється у дискретні форми сигналу, зокрема, у двійкові коди, або в одиничні коди.

За осанні роки впроваджується парадигма обчислень «біля сенсору». При цьому здійснюється переміщення функціональної обробки даних ближче до сенсорних компонентів за допомогою функціональних перетворювачів, встановлених в інтерфейси зв’язку або вбудованих у сенсори для того, щоб зменшити об’єм переданих даних та навантаження на інтерфейси зв’язку [1]. Схему сенсорної системи показано на рис.1.1, яка включає у себе функціональний перетворювач.

У розподілених системах широке застосування знаходять сенсори, що генерують вихідний сигнал у вигляді частотного, імпульсного сигналу або сигналу широтно-імпульсної модуляції.

Загальна схема сенсорної системи показано на рис.1.2.

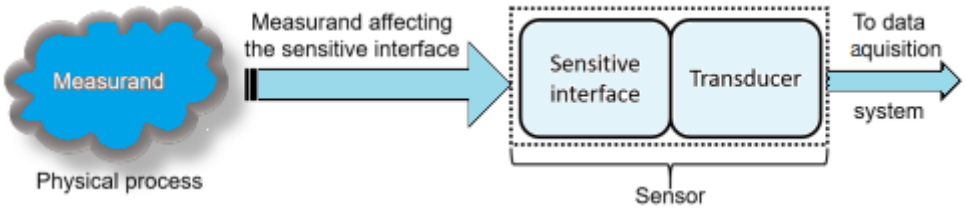

Рисунок 1.2 – Схема сенсорної системи

Перенесення обчислень ближче до сенсорів призводить до певних завдань, що необхідно вирішувати.

1. По-перше, необхідно виконувати обчислення в тому ж форматі даних, які генеруються на виході сенсора. Часто використовуваним варіантом є присенсорне перетворення в аналоговій формі. Іншим варіантом є перетворення форми імпульсу, оскільки перетворення аналогових сигналів у параметри імпульсів за певний час є нескладним, при цьому перетворення формату даних можна поєднати з необхідними обчисленнями.

2. По-друге, існує потреба в розробці спеціальних обчислювальних пристроїв, які мають бути швидкодіючими, енергоефективними, простими та надійними, що здатні виконувати свої функції у реальному часі. Прикладами таких спеціалізованих обчислювачів є стохастичні обчислення поблизу датчиків, обчислення на основі пам’яті з використанням пам’яті для окремих завдань та елементи процесора на основі імпульсної обробки [3].

Основною метою розвитку архітектур апаратних потокових обчислювачів та перетворювачів є спрощення структур та блоків за рахунок представлення інформаційних сигналів від сенсорів у біт-потоковій (імпульсній) формі. Біт-потокові обчислювачі реального часу здійснюють

потоковий спосіб обчислень (перетворення) з паралельно-послідовним виконанням перетворень при подачі вхід одиничних бітів потоку у відповідності до функції лінеаризації сигналу.

Бітовий потік - це частотний сигнал, у якому імпульси (біти) одиничної амплітуди (рис.1.3). Інформаційні дані у бітових потоках представлені фіксованим значенням бітів (імпульсів) Xmax за часовий інтервал T у потоках частотних або імпульсних сигналів, а також у потоках сигналу широтно-імпульсної модуляції.

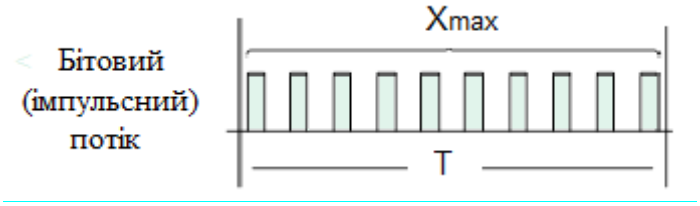

Рисунок 1.3 – Бітовий потік даних

При цифровій обробці інформаційних сигналів біт-потокова форма використовується в сенсорних інтерфейсах та при формуванні керуючих сигналів в СУ [4].

Апаратні обчислювачі функції добування кореня застосовуються:

– у розподілених СУ та ІВС в якості функціональних online-обчислювачів, що перетворюють сигнали від сенсорних компонентів з частотним та імпульсним виходом;

– у вимірювальних системах та приладах при математичній обробці інформаційних сигналів сенсорів при вимірюванні витрати рідини;

– в якості обчислювального блоку у вимірювальних системах при здійсненні непрямих вимірювань;

– в архітектурах потокових процесорів при потоковій обробці сигналів.

Розробка спеціалізованих апаратних online-обчислювачів, що виконують функціональне перетворення сигналів за необхідною функцією на основі

потокового способу обчислень є актуальним завданням. Потоковий спосіб обчислень передбачає одночасне паралельно-послідовне виконання перетворень над окремими бітами потоку, які виконують за рахунок використання методів формування приростів ступінчастої функції, зокрема на основі оберненої функції або алгоритмічним шляхом. Потоковий спосіб обчислень передбачає цифрову функціональну розгортку в реальному часі є основною перевагою таких обчислювальних операцій і дозволяє спростити архітектури пристроїв функціонального перетворення.

1.2 Етапи проектування апаратних потокових обчислювачів елементерних математичних функцій

Розробка потокових обчислювачів елементерних математичних функцій передбачає загальний порядок етапів проектування.

1. Розробка математичних моделей потокових обчислювачів з використанням методу формування приростів ступінчастих функцій з мінізацією похибки обчислень.

На даному етапі здійснюється вибір апроксимуючої функції з урахуванням заданої похибки обчислень. На підставі методу ступінчастої апроксимації на основі оберненої функції розроблюється математична модель пристрою.

2. Розробка архітектурної моделі обчислювача.

На підставі математичної моделі пристрою, алгоритму конвеєрних обчислень та загальної архітектури обчислювача створюється архітектура розроблюваного пристрою.

3. Розробка автоматної моделі обчислювача. Пристрій реалізується на основі кінцевого автомату.

На підставі математичної моделі та архітектури обчислювача здійснюється розроблення граф-схеми алгоритму (ГСА) операційного

автомата реалізації функції. та графа переходів керуючого автомату обчислювача;

4. Апаратна реалізація пристрою. Даний етап передбачає:

а) складання специфікації пристрою та виконання теоретичних розрахунків: обчислення значень апроксимуючої функції при заданій довжині вхідного бітового потоку x = xmax;

б) на основі зворотної функції визначення вибіркових значень xy вхідного бітового потоку x1, x2, …, xmax, що відповідають рівням вузлів апроксимації відтворюваної ступінчастої функції

в) розрахунок арифметичного ряду k-го порядку та арифметичних рядів різниць відповіднх порядків з використанням алгоритму конвеєрних обчислень, що покадений в основу конвеєрних структур архітектури пристрою;

г) визначення початкових значень ініціалізації компонентів пристрою; складання таблиці обчислювального процесу у компонентах розроблюваного обчислювача для заданих значень бітового потоку xmax з урахуванням значень ініціалізації компонентів;

ґ) розробка HDL-моделі обчислювача на підставі ГСА та графа переходів пристрою з використанням HDL – шаблонів на базі автоматного опису пристрою;

д) проведення верифікації поведінкової моделі та імплементації HDL-моделі обчислювача у ПЛІС.

В якості платформи для імплементації функціональних обчислювачів можут бути використані програмовані логічні інтегральні схеми (ПЛІС). Імплементація пристроїв у ПЛІС дозволяє розширити функціональні можливості пристроїв, підвишити швидкодію за рахунок конвеєрних структур та забезпечити надійність та енергефективність.

За рахунок високої функціональності платформ ПЛІС можливе об'єднання на одному кристалі процесорних пристроїв та інтерфейсної логіки. ПЛІС дозволяє зменшити час на проектування пристроїв, дає можливості

створення реконфігурованих архітектур цифрових пристроїв (ЦП), за рахунок спрощення архітектур, зменшення апаратних витрат та перепрограмування. Розроблення ЦП, зокрема, потокового online-обчислювача на платформі ПЛІС здійснюється разом із використанням систем автоматизованого проектування (САПР). При проектуванні з використанням ПЛІС треба здійснити етапи, що включають вибір платформи ПЛІС, типу FPGA; створення проекту з використанням системи проектування; верифікацію поведінкової моделі пристрою; синтез розробки пристрою; етап трасування. Виконується моделювання у часі, далі - програмування та імплементація апаратної моделі обчислювача в кристал.

1.3 Мета та задачі проектування

Метою кваліфікаційної роботи є розроблення апаратного потокового обчислювача функції вилучення кореня на платформі ПЛІС з використанням мов опису апаратури.

Об’єктом розробки є біт-потокові обчислювачі математичних функцій. Предметом розробки є математичні, архітектурні та автоматні моделі біт-потокових обчислювачів функції вилучення кореня.

Задачами кваліфікаційної роботи є:

– аналіз способу функціонального перетворення бітових потоків в потокових обчислювачах математичних функцій;

– аналіз математичної моделі потокового обчислювача функції вилучення кореня;

– розробка архітектурної моделі обчислювача;

– розробка автоматної моделі пристрою на основі кінцевого автомату;

– вибір платформи для апаратної реалізації пристрою;

– виконання теоретичних розрахунків екпериментальної частини проекту;

– розробка HDL-моделі пристрою з використанням HDL-шаблонів;

– верифікація поведінкової моделі обчислювача та його імплементація використанням САПР цифрових пристроїв у платформу ПЛІС Xilinx.

2 МАТЕМАТИЧНА, АРХІТЕКУРНА ТА АВТОМАТНА МОДЕЛІ АПАРАТНОГО ОБЧИСЛЮВАЧА ФУНКЦІЇ ВИЛУЧЕННЯ КОРЕНЯ

2.1 Математична модель обчислювача

При розробленні математичної моделі обчислювача функції вилучення кореня з біт-потоковою формою аргументу може бути використано метод відтворення функцій за допомогою ступінчастої апроксимації на основі оберненої функції. Важливим і визначальним при відтворенні функцій ступінчастим способом є похибка апроксимації. Принципово-точні методи апроксимації функцій, до яких відноситься метод на освнові оберненої функції дають кращий результат за точністю обчислення функції тому, що методична похибка апроксимації при даному методі відсутня. У методі абсолютну похибку вітворення функції враховано у аналітічному виразі апроксимуючої функції, похибка є мінімальною та складає ±0,5 одиниці молодшого біту аргументу.

За методом формування приростів ступінчастих функцій на основі обернених функцій [6] вибіркові значення x_y, що обираються з вхідного потоку x та відповідають рівням вузлів апроксимації відтворюваної ступінчастої функції визначаються за формулою:

$$\Psi(y-|\delta_{max}|) \le x_y < \Psi(y-|\delta_{max}|)+1 \qquad (2.1)$$

де x, y – вхідний, вихідний бітові потоки; $\Psi(y-|\delta_{max}|)$ – обернена функція.

Значення xy, що обираються з вхідного бітового потоку визначаються при підстановці y від 1 до y_k е ліву частину нерівності (2.1). При цьому обчислюють ліву частину нерівності та округлюють дискретні значення у більшу сторону

до цілого числа. Метод забезпечує потоковий спосіб відтворення функції на виході обчислювача в реальному масштабі часу.

З нерівності (2.1) може бути записана рівність, що дозволяє визначити значення вибірок x_y з мінімальною похибкою обчислень $|\delta_{max}| = 0{,}5$

$$x_y = [\Psi(y - 0{,}5)] + 1 \tag{2.2}$$

У роботі було проаналізовано математичну модель потокового обчислювача функції вилучення кореня довільного степеня з абсолютною похибкою обчислень. Метод ступінчастої апроксимації на основі оберненої функції забезпечує потоковий спосіб відтворення функції на виході обчислювача в реальному масштабі часу.

Розглянемо відому математичну модель апаратного біт-потокового обчислювача вилучення кореня квадратного.

Апроксимуюча функція, яку відтворює на виході апаратний обчислювач, має вигляд:

$$y = [\sqrt{x} + 0{,}5] \tag{2.3}$$

Мінімальна абсолютна похибка обчислення функції (2.3) врахована в аналітичному запису функції, дорівнює $|\delta_{max}| = 0{,}5$.

На основі оберненої функції, може бути записано нерівність, що реалізується в обчислювачі:

$$2^2 x_y \geq (2y_k - 1)^2 \quad . \tag{2.4}$$

Значення x_y, що обираються з вхідного бітового потоку та подаються на вихід пристрою, визначаються за формулою:

$$x_y = [(y - 0,5)^2] + 1 \qquad (2.5)$$

Математична модель апаратного потокового обчислювача добування квадратного кореня має вигляд системи нерівностей:

$$\begin{gathered} 2^2 x_1 \geq (2y_1 - 1)^2 \\ 2^2 (x_2 - x_1) + \Delta_1 \geq (2y_2 - 1)^2 - (2y_1 - 1)^2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ 2^2 (x_y - x_{y-1}) + \Delta_{y-1} \geq (2y_k - 1)^2 - (2y_{k-1} - 1)^2 . \end{gathered} \qquad (2.6)$$

У (2.6) різниця Δ_{y-1}, що утворюється при порівнянні лівої та правої частин кожної нерівності у математичній моделі визначається як:

$$\Delta_{y-1} = 2^2 (x_y - x_{y-1}) + \Delta_{y-2} - (2y_k - 1)^2 + (2y_{k-1} - 1)^2 .$$

При виконанні першої нерівності математичної моделі обчислювача випливає, що першому вихідному біту y=1 відповідає біт (імпульс) с номером x1 вхідного потоку x, який буде обраний з потоку бітів x. Аналогічним чином, наступні вихідні біти y_k будуть формуватися, коли будуть виконуватися відповідні нерівності математичної моделі.

Для апроксимуючої функції вилучення квадратного кореня (2.4) запропоновано інший підхід до розроблення обчислювача, який спрощує архітектуру пристрою та зменшує апаратурні витрати.

В роботі було запропоновано математичну модель спроектованого обчислювача та розглянуто алгоритм конвеєрних обчислень, що є основою побудови біт-потокових обчислювачів з конвеєрною структурою. Розглянемо більш детально дані питання.

1. Математична модель спроектованого обчислювача.

Для функції квадратичного радикалу з урахуванням похибки обчислень 0,5 формула вибіркових значень x_y, яка може бути записана на основі (2.1) та (2.2):

$$(y-0{,}5)^2 < x_y < (y-0{,}5)^2 + 1 \tag{2.7}$$

$$x_y = [(y-0,5)^2] + 1. \tag{2.8}$$

На підставі (2.8) формула для визначення вібіркових бітів x_y з вхідної послідовності x з округленням до цілих чисел набуває вигляду:

$$x_y = y^2 - y + 1. \tag{2.9}$$

Тоді основна нерівність, що реалізується в обчислювачі має вигляд:

$$x_y \geq (2y_k - 1)^2 . \tag{2.10}$$

Математична модель обчислювача по аналогії з (2.4) має аналогічний аналітичний запис, крім множника 2^2, що є у лівій частині нерівностей. Це спрощує архітектуру спроектованого обчислювача.

Математична модель спроектованого обчислювача є системою різницевих нерівностей:

$$\begin{gathered} x_1 \geq (2y_1 - 1)^2 \\ (x_2 - x_1) + \Delta_1 \geq (2y_2 - 1)^2 - (2y_1 - 1)^2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ (x_y - x_{y-1}) + \Delta_{y-1} \geq (2y_k - 1)^2 - (2y_{k-1} - 1)^2 . \end{gathered} \tag{2.11}$$

Приведена математична модель реалізується у спроектованому обчислювачі. Віхідні біти y_k будуть згенеровані на виході пристрою при виконанні кожної нерівності (2.11).

2. Алгоритм конвеєрних обчислень.

В основу принципа функціонування потокового обчислювача функції вилучення кореня покладено алгоритм конвеєрних обчислень за допомогою якого здійснюється конвеєрна обробка біт-потокових даних.

Алгоритм конвеєрних обчислень передбачає обчислення полінома n-го порядку, аргументом є бітовий потік x максимальною довжиною x_{max}.

Поліном n-го порядку має вигляд:

$$y = \sum_{i=0}^{n} a_i x^i \quad . \tag{2.12}$$

де n, a_i – цілі числа.

Якщо $a_n = 1,\ a_{n-1} = a_{n-2} = ... = a_0 = 0$, тоді поліном (2.12) приймає вигляд:

$$y = x^n \tag{2.13}$$

Задача побудови біт-потокових обчислювачів з конвеєрною структурою реалізується здійсненням зменшенння порядку різницевих рівнянь, а отже, визначенням арифметичного ряду n-го порядку та арифметичних різниць 1-го, 2-го , ..., n-го порядків відповідно.

Значення функції y_i для полінома визначаються як:

$$y_i = f(i+1) - f(i) \tag{2.14}$$

де i – цілі числа від 1 до y_k. $f(i+1)$ - значення функції при x_{i+1}.

Арифметичні ряди різниць відповідних порядків визначаються на основі системи рівнянь:

$$\begin{aligned} \Delta_i &= y_{i+1} - y_i \\ \Delta_i^2 &= \Delta_{i+1} - \Delta_i \\ \Delta_i^3 &= \Delta_{i+1}^2 - \Delta_i^2 \\ &\cdots\cdots\cdots \\ \Delta_i^n &= \Delta^{n-1}{}_{i+1} - \Delta_i^{n-1} \end{aligned} \qquad (2.15)$$

Система рівнянь (2.15) - математична модель конвеєрного біт-потокового обчислювача поліномів, включеного у зворотний зв'язок архітектури обчислювача вилучення кореня довільного степеня.

Розглянутий алгоритм конвеєрних обчислень використано при проектуванні архітектури апаратного потокового обчислювача функції вилучення квадратного кореня, що розроблений у даній роботі.

При підстановці y=1, 2, 3, … у (2.9) буде отримано числову послідовність значень x_y, які обираються з вхідного бітового потоку x і являють собою вихідний бітовий потік у пристрою. При цьому x_y : x_1, x_2, x_3, x_4, x_5, … є арифметичним рядом 2-го порядку.

Далі визначають арифметичні ряди різниць першого Δ і другого Δ^2 порядків відповідно за формулами:

$$\begin{aligned} \Delta_i &= y_{i+1} - y_i, \\ \Delta_i^2 &= \Delta_{i+1} - \Delta_i \end{aligned} \qquad (2.16)$$

Отримані арифметичний ряд 2-го порядку x_y і ряди різниць першого Δ і другого Δ^2 порядків використовують при побудові розробленої у данній роботі архітектури апаратного обчислювача вилучення квадратного кореня.

Для функції $y=[\sqrt{x}+0,5]$ арифметичний ряд 2-го порядку та арифметичні ряди різниць першого Δ і другого Δ^2 порядків відповідно мають вигляд:

$$\begin{aligned} &x_y: 1, 3, 7, 13, 21, 31, \dots \\ &\Delta: \quad 2, 4, 6, 8, 10, \dots \\ &\Delta^2: \quad 2, 2, 2, 2, \dots \end{aligned} \tag{2.17}$$

В арифметичному ряду різниць другого порядку - константа, що дорівнює 2. Розрахунки арифметичних рядів та їх різниці необхідні для розрахунків експериментальної частини проекту.

2.2 Архітектурна модель обчислювача

В роботі було розроблено архітектуру апаратного потокового обчислювача функції квадратичного радикалу, який побудовано на основі аналізу математичної моделі пристрою та алгоритму конвеєрних обчислень, а також архітектури біт-потокового обчислювача вилучення квадратного кореня.

Відома загальна архітектура біт-потокового обчислювача, що приведено на рис. 2.1. Архітектура має 3 компонента:

– суматор результату SM_RES, який є основним обчислювальним компонентом, де здійснюється порівняння двійкових кодів лівої та правої частин нерівностей математичної моделі обчислювачів;

– блок прямого зв’язку Block 1, з якого в SM_RES подаються прямі двійкові коди чисел функції аргументу (ліва частина нерівностей математичної моделі);

– блок зворотного зв’язку Block 2, зі сторони якого в SM_RES подаються додаткові двійкові коди чисел функції рівнів вуздів апроксимації відтворюваної функції.

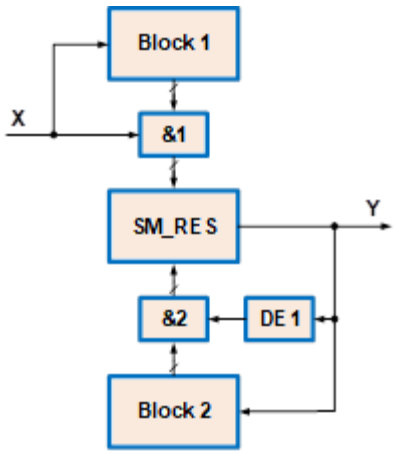

Рисунок 2.1 – Загальна архітетура біт-потокового обчислювача

Архітектуру біт-потокового обчислювача квадратичного радикалу показано на рис. 3.2. Дана архітектура реалізує математичну модель пристрою загального підходу (2.6).

Обчислювач містить компоненти: регістри RG1, RG2; паралельні суматори SM_RES, SM; елементи затримки DE1, DE2; групи логічних елементів &1, &2, &3.

В архітектурі групи елементів &1, &2, &3 поразрядно з'єднують регістр RG1, і регістр суматора SUM_RES, регістри суматорів SUM_RES та SUM1, та RG2 і SUM1 відповідно. Час затримки елементу DE1 менше, ніж час затримки елементу DE2.

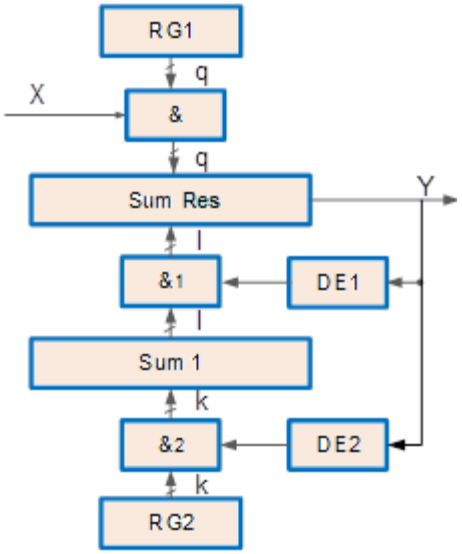

Рисунок 2.2 – Архітектура біт-потокового обчислювача вилучення квадратного кореня

Апаратний біт-потоковий обчислювач вилучення квадратного кореня працює у режимі вибірки, коли з вхідного бітового потоку обираються певні біти і подаються на його вихід. Пристрій у режимі вибірки є дільником чисел, у якому змінний коефіцієнт ділення.

Компоненти, що містить архітектура, перед подачею на вхіт бітового потоку x ініціалізуються наступними значеннями:

RG1 - прямим двійковим кодом числа 2^2.

Sum_Res ініціалізується додатковим двійковим кодом числа 1, тобто 2^n-1, де n – розрядність суматора (перше значення арифметичного ряду 2-го порядку, що дорівнює 1);

Sum1 ініціалізується прямим двійковим кодом першого значення арифметичного ряду різниць 1-го порядку;

RG2 – прямим двійковим кодом константи, що утворена в арифметичному ряду різниць 2-го порядку.

Функціонування пристрою здійснюється наступним чином. При подачі на його вхід бітового потоку x. Коли приходить перший біт потоку x на вхід пристрою, він відкриває групу елементів &. При цьому прямий двійковий код

числа 2^2 з регістра RG1, переноситься у SM_RES та підсумовується із його вмістом. У результаті вміст регістра суматора SM_RES стане рівним $2^n–1+2^2$. При цьому на виході пристрою з'явиться біт переповнення SM_RES. При цьому реалізується перша нерівність системи нерівностей математичної моделі пристрою (3.10).

Біт переповнення SM_RES є вихідним бітом пристрою Y. По зворотному зв'язку, проходить DE1 та відкриває групу елементів - ключів &1, переносить в SM_RES із суматора SUM1 додатковий двійковий код числа, що міститься в суматорі. Пройшовши через елемент затримки DE2, біт переповнення подається на групу &2 і паралельний двійковий код числа з RG2 переноситься в SUM1 та підсумовується з вмістом. Таким чином, у зворотному зв'язку архітектури здійснюються конвеєрні обчислення.

Надалі при надходженні чергових бітів потоку x процеси обчислення у пристрої будуть повторюватися циклічно.

Другий біт переповнення на виході SM_RES з'явиться при надходженні на вхід пристрою вибіркового біта x_2, при якому буде виконана друга нерівність (2.10). В результаті роботи пристрій відтворює ступінчасту функцію (2.7).

В роботі було запропоновано реконфігуровану архітектурну модель апаратного потокового обчислювача функції вилучення квадратного креня на основі аналізу математичної моделі пристрою (2.16) та отриманих формул (2.14) і (2.21).

З архітектури (рис.2.2) було виключено елементи: регістр RG1, група &, RG2, група &2, елемент затримки DE2. Суматор Sum 1 замінено на лічильник Count_RES.

Архітектурну модель спроектованого апаратного потокового обчислювача з вхідним бітовим потоком даних x, що реалізує розглянутий вище принцип вибірки бітів x_y. приведено на рис. 2.3.

В реконфігурованій архітектурі компонентами є: паралельний суматор SUM, підсумовуючий лічильник результату Count_RES, група елементів &1, елемент затримки DE1.

Розроблена архітектура у порівнянні з архітектурою, зображеною на рис. 3.2 є містить менше компонентів та є простішою, генерує результат обчислення функції вилучення квадратного кореня у біт-потоковій формі і у двійковому коді, що накопичується в лічільнику результату Count_RES - це є перевагами даного пристрою.

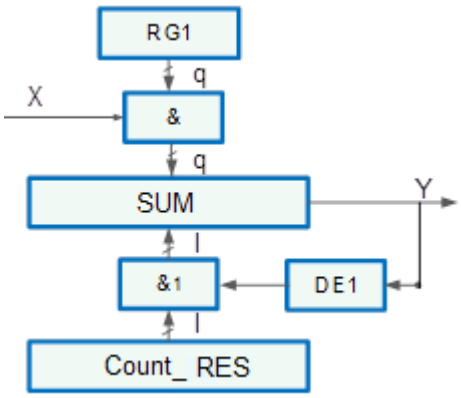

Рисунок 2.3 – Архітектурна модель спроектованого апаратного обчислювача добування квадратного кореня

Ініціалізація компонентів обчислювача:

– суматор SUM ініціалізується числом $2^i - 1$, числом 1 у додатковому коді, так як перше значення x_y дорівнює 1: для функції $y = [\sqrt{x} + 0,5]$.при x=1, y=1 – перший біт, що подається на вхід пристрою має бути обраним і з'явитися на виході пристрою;

– лічильник Count_RES ініціалізується числом 0;

– регістр RG1 ініціалізується числом 1.

Розглянемо роботу пристрою. На вхід обчислювача подається бітовий потік x, выдкривъ групу елементів &. Двійковий код числа ініціалізації регістра RG переноситься в суматор і підсумовується з його вмістом. В

обчислювачі за допомогою групи елементів &1 двійковий код числа з Count_ RES, переноситься в суматор SUM у додатковому коді зі зсувом на один розряд вправо, тобто $2^i - 2N$. де N – двійковий код числа, яким ініціалізовано лічильник результату Count_ RES.

В процесі конвеєрних обчислень, кожним бітом переповнення SUM, в суматор SUM переноситься додатковий код подвійного числа, що утворюється в Count_ RES в процесі накопичення вихідних бітів пристрою в лічильнику результату та відповідає обчисленням значень в ряду різниць першого прорядку в (2.22).

Результат обчислення функції вилучення квадратного кореня в процесі роботи обчислювача формується у вигляді бітового потоку на виході SUM та у вигляді паралельного двійкового коду в лічільнику результата Count_RES.

З приходом першого біту послідовності x на вхід SUM, на його виході з'являється перший біт переповнення, а його вміст стає рівним нулю.

Біт переповнення з виходу SUM:

– надходить на вихід пристрою y;

– потрапляє на вхід Count_ RES, в результаті вміст Count_ RES буде дорівнювати 1 у двійковому коді;

– пройшовши елемент затримки DE1, за допомогою групи елементів &1 вносить у SUM число, $2^i - 2$, тобто додатковий двійковий код числа 1 зі зсувом на один розряд вправо. Число 2 є першим значенням в ряду різниць 1-го порядку в (2.22).

При надходженні на вхід обчислювача чергових двох бітів послідовності x SUM знову переповнюється і його вміст стає рівним 0.

Далі, аналогічно до попереднього біту переповнення, другий біт переповнення SUM:

– надходить на вихід пристрою, а також на вхід лічильника Count_ RES. В результаті вміст лічильника буде мати двійковий код числа 2;

– пройшовши DE1, за допомогою групи елементів &1 з Count_ RES в SUM вносить додатковий код числа 2 зі зсувом вправо на один розряд, тобто $2^i - 4$. Число 4 є другим значенням в ряду різниць 1-го порядку в (2.22).

Надалі при надходженні чергових вхідних бітів x описані процеси в обчислювачі будуть циклічно повторюватися. У SUM будуть послідовно вводитися додаткові коди чисел ряду різниць першого порядку Δ (2.22), а на вихід пристрою будуть надходити біти з номерами xy бітової послідовності x, що відповідають рівням вузлів апроксимації ступінчастої функції. При надходженні в пристрій останнього вхідного біту послідовності x в лічильнику результату Count_ RES буде зафіксований результат обчислення функції вилучення квадратного кореня (2.7).

Позитивною особливістю даної моделі апаратного обчислювача є те, що він може видобувати квадратний корінь з чисел, представлених паралельними двійковими кодами

3.3 Автоматна модель обчислювача

В кваліфікаційній роботі запропоновано автоматну модель апаратного потокового обчислювача добування кореня на підставі кінцевого автомату, який містить певні стани, що пов’язані з обчислювальними операціями, та переходи між станами.

Автоматна модель обчислювача є сукупністю керуючого і операційного автоматів. Керуючий автомат керує обчислювальними операціями пристрою, перехід з одного стану в інший відбувається по умові переповнення суматору, який служить основним обчислювальним компонентом архітектури. Автоматну модель було розроблено на основі автомата Мура, який є більш придатним для реалізації обчислювачів елементарних математичних функцій з біт-потоковою формою вхідного сигналу, зручним є те, що обчислювальні операції здійснюються безпосередньо у станах керуючгого автомату.

Граф переходів керуючого автомату пристрою приведений на рис. 2.4, є нескладним. Він містить три стани: a0, a1, a2 та працює за певним алгоритмом.

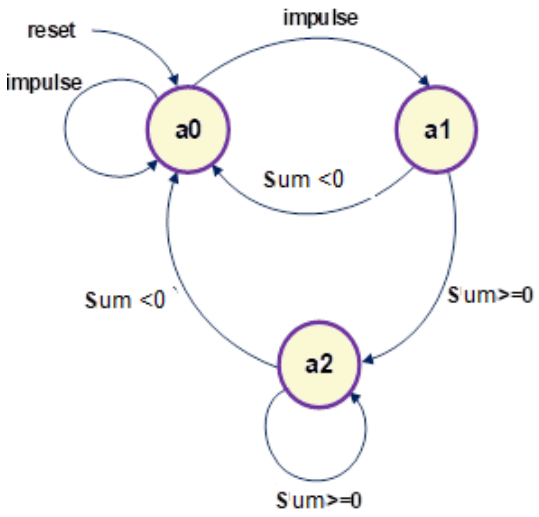

Рисунок 2.4 – Граф переходів керуючого автомата пристрою

Розглянемо алгоритм роботи керуючого автомату.

а) сигнал скиду «reset» переводить автомат у стан a0, що є початковим, та знаходиться у стані до того, як буде поданий сигнал «impulse» вхідного потоку бітів x. З приходом «impulse» автомат переходить в a1;

б) в a1 здійснюються обчислення у SUM обчислювача: додається до SUM біт, який поступає на вхід пристрою, Якщо вміст суматора SUM>=0, то на виході SUM біт переповнення, що є вихідним бітом пристрою, керуючий автомат видає сигнал для формування вихідного біта y. При цьому автомат переходить у a2. Якщо вміст SUM<0, автомат переходить у a0;

в) в a2 виконуються обчислення у зворотному зв'язку архітектури. У цому стані здійснюються конвеєрні обчислення: до вміста лічильника додається біт і злічильник здійснюється перенос числа з лічильника в суматор. Якщо значення суми у суматорі SUM>=0 , автомат знову переходить у стані a2, якщо умова не виконується, тоді SUM<0 і автомат переходить у a0.

Рис. 2.5 демонструє граф-схему алгоритму (ГСА) операціного автомату реалізації функції вилучення квадратного кореня, яка була розроблена на

підставі математичної, архітектурної моделей обчислювача та розмічена за принципом розмітки автомата Мура.

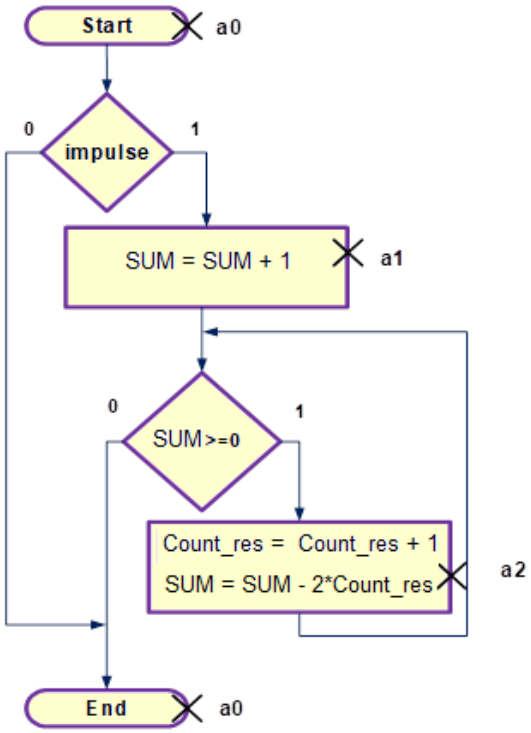

Рисунок 2.5 – ГСА операційного автомата обчислювача

ГСА містить початкову та кінцеву вершини, що розмічені a0, операторні вершини a1 та a2 та умовні вершини. Вершина a1 містить обчислення у суматорі SUM. Вершина a2 містить дві обчислювальних операції, що відбуваються в компонентах Count_res та SUM зворотного зв'язку архітектури.

Розглянемо ГСА операційного автомата, що реалізує алгоритм обчислень у пристрою:

а) при скиданні сигналом «reset» компоненти пристрою переходять у початковий стан – стан ініціалізації компонентів, регістри компонентів ініціалізуються певними двійковими кодами чисел, що заздалегідь визначені: SUM = -1 , Count_RES=0;

б) в ГСА є умовна вершина «impulse», коли приходить на вхід пристрою біта вхідного потоку x, далі переходимо в операторну вершину a1, в якій значення регістра SUM збільшується на значення 1 вхідного біта і вміст суматора стає рівним SUM>=0;

в) коли значення регістра суматора SUM>=0, то на виході пристрою генерується біт y. При цьому у операторній вершині a2 здійснюються конвеєрні обчислення: значення лічільника Count_рез збільшується на 1 та від значення регістра суматора SUM віднімається подвоєне значення лічильника Count_рез, а отже, SUM – 2* Count_RES. Якщо переповнення SUM не відбувається, тобто SUM<0, то втомат переходить у стан a0.

3.4 Вібір платформи ПЛІС для апаратної реалізації обчислювача

В процесі проектування обчислювача було здійснено синтез та імплементацію розробленої моделі пристрою у платформу ПЛІС.

Спеціалізовані обчислювальні платформи FPGA (Field-Programmable Gate Array) відрізняються своєю апаратною гнучкістю. Компонентами FPGA, що складають основу платформи є програмована логіка, логічні елементи, блоки пам'яті та порти входу/виходу.

Для виконання обчислювальних завдань є можливості перепрограмування FPGA для створення спеціалізованих логічних схем, що реалізують певну функціональність, за рахунок логіки та з'єднань між елементами. Використання FPGA здійснюється для реалізації цифрової обробки сигналів, прискорення обчислень та інших завдань.

На платформі FPGA можна реалізовувати складні логічні функції та обчислювальні операції за допомогою програмування логічних елементів та їх з'єднань, а також створювати спеціалізовані обчислювальні архітектури, що здійснюють оброблення сигналів, криптографічні операції, машинне навчання та багато інших завдань.

Важливим є здатність до перепрограмування FPGA, це дає можливість використовувати одну апаратну платформу для виконання різних завдань без необхідності зміни апаратних компонентів, що говорить про гнучкість і потужність FPGA, яка є важливою технологією для великої кількості застосувань у сучасних обчислювальних системах.

FPGA використовують таблиці LUT (Look-Up Table). Вони зберігають комбінації вхідних сигналів і визначають вихідний сигнал відповідно до програми. Також, FPGA мають блоки пам'яті, які дозволяють зберігати дані та конфігураційну інформацію для програмування. Ці блоки пам'яті дозволяють FPGA зберігати стан і дані між операціями. Обчислення відбуваються паралельно, що дозволяє в режимі реального часу здійснювати складні обчислювальні операції, у тому числі і при здійсненні конвеєрних обчислень [10].

При створенні прототипів платформи ПЛІС є зручними при макетуванні пристроїв у вигляді великих інтегральних схем (ВІС). Використання ПЛІС предоставляє можливість ефективно та швидко із невеликими затратами створювати складні пристрої, крім того змінювати багато разів конфігурації пристроїв та удосконалювати їх певні функції, здійснювати налагодження шляхом перепрограмування обчислювальних та інших функцій, а також, зв'язків елементів.

Основою ПЛІС FPGA є матриця логічних комірок (logic cells). Логічні комірки оточені блоками введення-виведення та забезпечують підключення до зовнішніх виводів ПЛІС.

Розроблення апаратного потокового обчислювача функції добування кореня здійснювалась на платформі ПЛІС Xilinx Spartan-3E серії XC3s500e (рис.2.6).

Ресурси ПЛІС Xilinx Spartan є зручними для здійснення процесу проектування цифрових пристроїв різного призначення. Програмні засоби WebPACK ISE - засоби, що реалізують процес наскрізного проектування. Він

передбачає повний цикл проектування та розроблення спеціалізованих цифрових пристроїв на ПЛІС.

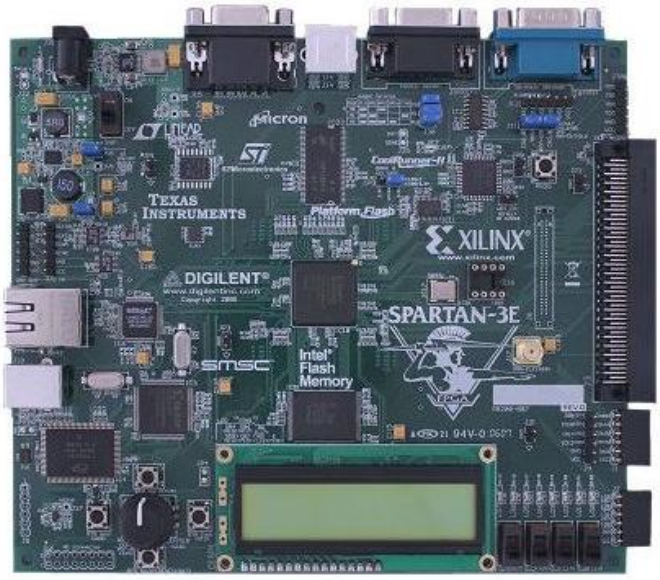

Рисунок 2.6 - ПЛІС Xilinx SPARTAN 3E серії XC3s500e

На рис. 2.7 приведено технічні характеристики кристалу, що обраний для імплементації потокового обчислювача.

Серія	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

Рисунок 2.7 – Технічні характеристики кристалу FPGA Spartan 3E XC3S500E

Відносно до проектування подібних потокових обчислювачів цикл проектування описано у підрозділі 1.2 Система наскрізного проектування включає у себе системотезнічну та схемотехнічну складові, включаючи різні етапи розробки описів проекту, а також здійснює синтез та моделювання. Після даних етапів виконується розміщення, програмування кристала ПЛІС.

Застосування прграмного пакету WebPACK ISE дає можливості проектувальникам значно скоротити час на здійснення розробки цифрового пристрою, а також підвищити ефективність результатів. У склад пакету входить модуль iMPACT, що може бути застосований при визначеній конфігурації пристрою уцілому для всіх кристалів Xilinx.

3 АПАРАТНА РЕАЛІЗАЦІЯ СПРОЕКТОВАНОГО ОБЧИСЛЮВАЧА

У розділі приведено результати експериментальної апаратної реалізації потокового обчислювача функції вилучення кореня, приведено результати теоретичних обчислень заданої функції та обчислювального процесу у компонентах розробленої архітектури, здійснено верифікацію поведінкової моделі та імплементацію пристрою в ПЛІС Xilinx Spartan.

3.1 Розрахункова частина для експериментальної апаратної реалізації

В роботі було проведено експериментальну апаратну реалізацію з метою перевірки коректності роботи обчислювача.

Апроксимуюча функція, яку має відтворити розроблений обчислювач:

$$y = [\sqrt{x} + 0,5] \tag{3.1}$$

де x – бітовий вхідний потік (імпульсний потік);

0,5 – абсолютна похибка обчислення функції добування квадратного креня. Квадратні дужки означають, що дробова частина у результатах обчислення функції має бути не врахована. Пристрій працює з числами з фіксованою точкою.

Вихідні дані для проведення експерименту приведено на рис.3.1.

Показник степеня радикалу	Абсолютна похибка $\lvert\delta_{max}\rvert$	Кількість бітів (імпульсів) x_{max}
2	0,5	13

Рисунок 3.1 – Вихідні дані для екперименту

x_{max} – максимальна довжина вхідного бітового потоку x, кількість бітів у потоці, що подається на вхід пристрою.

На вхід потокового обчислювача поступають біти потоку x, обчислювач функціонує у режимі вибірки бітів x_y з вхідного бітового потоку.

На вхід обчислювача подається вхідний бітовий потік, довжина якого складає x_{max} = 13 бітів. На виході обчислювача буде сформований вихідний бітовий потік y - це результат обчислення функції (3.1) вилучення квадратного кореня із заданою похибкою. Результати обчислення функції (3.1) значень бітового потоку для потоку бітів потоку приведено на рис. 3.2. Результат обчислення функції Y отримано у цілих числах.

X – вхідні біти	Значення функції квадратичного радикалу Y
X=1	$y=[\sqrt{1}+0{,}5]=[1{,}5]=1$
X=2	$y=[\sqrt{2}+0{,}5]=[1{,}91]=1$
X=3	$y=[\sqrt{3}+0{,}5]=[2{,}23]=2$
X=4	$y=[\sqrt{4}+0{,}5]=[2{,}5]=2$
X=5	$y=[\sqrt{5}+0{,}5]=[2{,}74]=2$
X=6	$y=[\sqrt{6}+0{,}5]=[2{,}95]=2$
X=7	$y=[\sqrt{7}+0{,}5]=[3{,}15]=3$
X=8	$y=[\sqrt{8}+0{,}5]=[3{,}33]=3$
X=9	$y=[\sqrt{9}+0{,}5]=[3{,}5]=3$
X=10	$y=[\sqrt{10}+0{,}5]=[3{,}66]=3$
X=11	$y=[\sqrt{11}+0{,}5]=[3{,}82]=3$
X=12	$y=[\sqrt{12}+0{,}5]=[3{,}96]=3$
X=13	$y=[\sqrt{13}+0{,}5]=[4{,}1]=4$

Рисунок 3.2 – Результати обчислення функції вилучення квадратного кореня

Формула для визначення вібіркових бітів x_y з вхідної послідовності x з округленням до цілих чисел має вигляд:

$$x_y=[(y-0,5)^2]+1 \qquad (3.2)$$

При y = 1, 2, 3, 4 у (3.2) пораховано значення бітів послідовності x_y = 1, 3, 7, 13 відповідно. В результаті з вхідного потоку довжиною 13 бітів, буде обрано та подано на вихід обчислювача 1, 3, 7 та 13 біти потоку y = 1, 2, 3, 4 відповідно.

Скориставшись формулами (2.19), (3.21) алгоритму конвеєрних обчислень пораховано арифметичний ряд другого порядку та арифметичні ряди різниць 1-го та 2-го порядків, отримано результати:

x_y: 1, 3, 7, 13, 21, 31, . . .

Δ: 2, 4, 6, 8, 10, ...

Δ^2 2, 2, 2, 2, ...

Для архітектури (рис. 3.3) необхідно записати числа ініціалізації компонентів: SUM $=2^i$ -1, Count_RES=0; RG1=1. На рис. 3.3 показано обчислювальні операції у компонентах обчислювача.

Вхідний бітовий потік X	SUM	Імпульс переповнення SUM	Count_RES	Номери бітів потоку Y
1	-1 + 1 = 0 0 – 2 = -2	1	0+1=1	1
2	-2 + 1 = -1			
3	-1 + 1 = 0 0 – 4 = -4	1	1+1=2	2
4	-4 + 1 = -3			
5	-3 + 1 = -2			
6	-2 + 1 = -1			
7	-1 + 1 = 0 0 – 6 = -6	1	2+1=3	3
8	-6 +1 = -5			
9	-5 +1 = -4			
10	-5 +1 = -4			
11	-3 +1 = -2			
12	-2 +1 =–1			
13	-1+1 = 0 0 -8 = -8	1	3 + 1 = 4	4

Рисунок 3.3 – Обчислення у компонентах спроектованого обчислювача

Розрахунки обчислень у компонентах пристрою (рис. 3.3) та поява вихідних бітів Y відповідають результатам розрахунку функції вилучення кореня (рис.3.2) .

3.2 Структурно-блокова схема апаратного потокового обчислювача

При проведенні експериментальної апаратної реалізації пристрою було розроблено структурно-блокову схему апаратного обчислювача, яку зображено на рис. 3.4.

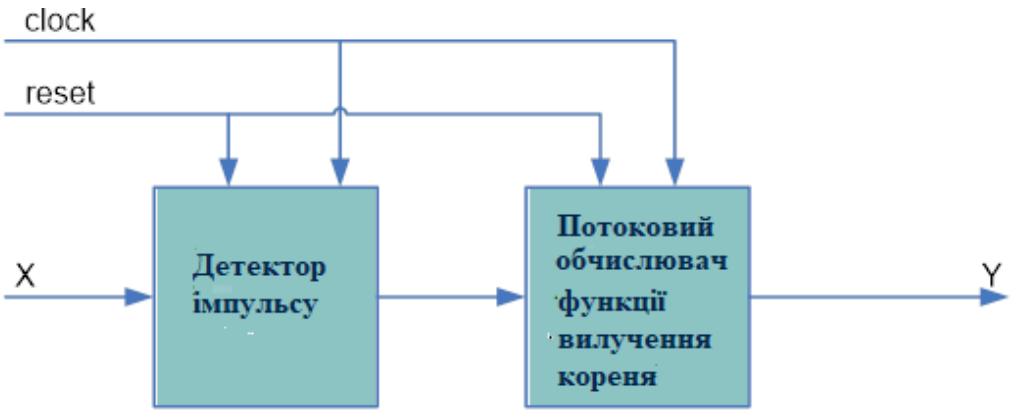

Рисунок 3.4 – Структурно-блокова схема апаратного обчислювача

У пристрою є два основних блоки.

1. Детектор вхідного імпульсу .
2. Блок потокового обчислювача функції вилучення кореня.

Блок «Детектор імпульсу» призначений для детектування бітового потоку (імпульсної послідовності) на вході пристрою x.

Імпульс детектується по двом сусіднім тактам сигналу clock. Якщо на черговому такті значення сигналу x = 0, а на наступному такті значення x = 1, то детектор детектує імпульс, і на виході встановлює відповідний сигнал «impulse», що дорівнює логічній «1». Даний сигнал буде отримано арифметичним блоком потокового обчислювача.

Блок «Потоковий обчислювач функції вилучення кореня» виконує операцію вилучення квадратного кореня аргументу x із заданою похибкою 0,5.

В потоковому обчислювачі результат вилучення кореня округлюється до найближчого цілого числа. На виході обчислювача формується вихідна біт-потокова послідовність Y. Після того, як з'явиться сигнал Ready на виході арифметичного блоку, що рівний '1', це буде означати, що даний блок видає сигнал готовності, щоб прийняти наступний біт на обчислювальну обробку. Результатом роботи є сигнал y, що являє собою результат обчислення заданої функції.

3.3 Опис HDL-моделі пристрою

При розробці апаратного потокового обчислювача функції вилучення кореня і написанні проекту на мові опису апаратури було обрано мову VHDL. Для здійснення верифікації проекту обрано пакет Active-HDL. Active-HDL дозволяє автоматизувати процес введення проекту в САПР, а також проводити аналіз результатів при проектуванні та вироблювати рішення на основі результатів.

Active-HDL є системою проектування та моделювання проектів на ПЛІС, у тому числі має змогу підтримувати проектування багаторівневих певної складності проектів на FPGA. Крім того пакет дає можливість проектувати цифрові пристрої на основі структурних схем, створювати графові моделі цифрових автоматів, а такоє конвертувати HDL-описи проектів в структурні схеми.

HDL-опис спроектованого пристрою має трирівневу ієрархічну структуру. Проект містить групу файлів, що об'єднуються за допомогою файла верхнього рівня «Powerfunc.top.vhdl». У цьому файлі задекларовані всі файли другого рівня, що мають нижчий рівень ірархії: «Powerfunc_inputbuffer.vhdl» та «Powerfunc.vhdl». Файл верхнього рівня «Powerfunc.top.vhdl» реалізує специфікацію обчислювача, що містить інтерфейс верхнього рівня: entity, в

якому записано усі вхідні і вихідні сигнали складових блоків, а також, всі порти компонентів архітектури, розрядність компонентів: 8 та 16. Лістинг 3.1 включає у себе HDL-опис компонентів даного пристрою.

Лістинг 3.1 – HDL-опис компонентів спроектованого обчислювача верхнього рівня «Powerfunc.top.vhdl»

```
architecture struct of powerfunc_toplevel is
   -- Component declaration of the "powerfunc(struct)" unit defined in
   -- file: "./src/powerfunc.vhd"
   component powerfunc
      generic(
         width1: natural := 8;
         width2: natural :=16);
      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(width2-1 downto 0));
   end component;
   -- Component declaration of the "powerfunc_inputbuffer(beh)" unit defined in
   -- file: "./src/powerfunc_inputbuffer.vhd"
   component powerfunc_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;
```

Опис поєднання блоків «Powerfunc_inputbuffer.vhdl» та «Powerfunc.vhdl» здійснено за допомогою оператору port map. Фрагмент поєднання блоків описано у лістингу 3.2. Дані блоки належать другому рівню ієрархії проекту.

Лістинг 3.2 – Фрагмент HDL-опису по'єднання блоків «Powerfunc_inputbuffer.vhdl» і «Powerfunc.vhdl»

```
   signal inputbuffer_out, powerfunc_out, ready: std_logic;
   signal count_t: std_logic_vector(width1-1 downto 0);
   signal sum_t: std_logic_vector(width2-1 downto 0);
begin
   InputBuffer_1 : powerfunc_inputbuffer
   generic map(
      width => width1
      )
   port map(
      x_i => x_i,
      freq_i => freq_i,
      reset_i => reset_i,
      ready_i => ready,
      y_o => inputbuffer_out
      );
   powerfunc_111 : powerfunc
   generic map(
      width1 => width1,
      width2 => width2)
   port map(
      x_i => inputbuffer_out,
      ready_o => ready,
      freq_i => freq_i,
      reset_i => reset_i,
      y_o => powerfunc_out,
      sum_o => sum_t
      );
```

HDL-файл «Powerfunc.vhd» об’єднує у собі компоненти 3-го рівня ієрархії – фйли керуючого автомату обчислювача «Powerfunc.ua.vhdl» та операційного автомата «Powerfunc.oa.vhdl». У лістингу 3.3 наведено фрагмент HDL-опису компонентів.

Лістинг 3.3 – HDL-опис компонентів керуючого та операційного автоматів пристрою

```
architecture struct of powerfunc is
   -- Component declaration of the "powerfunc_ua(beh)" unit defined in
   -- file: "./src/powerfunc_ua.vhd"
   component powerfunc_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);
    end component;
    -- Component declaration of the "powerfunc_oa(beh)" unit defined in
    -- file: "./src/powerfunc_oa.vhd"
    component powerfunc_oa
        generic(
            width1: natural := 16;
            width2: natural := 24);
        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(width2-1 downto 0));
    end component;
```

Фрагмент HDL-опису операційного автомату пристрою приведено у лістингу 3.4, в якому дано опис поведінкової моделі архітектури, приведено ініціалізацію компонентів SUM та Count_RES, а також обчислювальні операції у компонентах, конвеєр обчислень.

Лістинг 3.4 – HDL-опис обчислювальних операцій і операційному автоматі обчислювача

```
architecture beh of powerfunc_oa is
    signal sum: std_logic_vector(width1-1 downto 0);
    signal counter: std_logic_vector(width2-1 downto 0);
begin
    sum_o <= sum_1;
    sum_less_zero_o <= '1' when (sum_1 < 0) else '0';
process (clock_i, reset_i)
        begin
            if (reset_i = '1') then
                sum <= CONV_STD_LOGIC_VECTOR(-1, width);
                counter <= (others => '0');
            else
                if (falling_edge(clock_i)) then
                if (sum_plus_x_i = '1') then
                sum <= sum + 1;
            else
                if (sum_minus_counter_i = '1') then
                    counter <= counter + 1;
                    sum <= sum - counter - counter - 2;
                end if;
            end if;
        end if;
```

```
end if;
end process;
end beh;
```

Фрагмент HDL-моделі керуючого автомату обчислювача у формі автоматного шаблону описано у лістингу 3.5, для переходів з одного стану в інший в стан автомату здійснюється за допомогою оператору case.

Лістинг 3.5 – Фрагмент програми, що описує роботу керуючого автомата пристрою

```
process(state, x_i, sum_less_zero_i)
begin
   case (state) is
       when a0 =>
          if x_i = '1' then
             next_state <= a1;
          else
             next_state <= a0;
          end if;
       when a1 =>
          if sum_less_zero_i = '1' then
             next_state <= a0;
          else
             next_state <= a2;
          end if;
       when a2 =>
          if sum_less_zero_i = '1' then
             next_state <= a0;
          else
             next_state <= a2;
          end if;
       when others =>
             next_state <= a0;
  end case;
end process;
```

3.4 Верифікація та імплементація моделі потокового обчислювача

При розробленні апаратної реалізації потокового обчислювача добування кореня було отримано поведінкову модель пристрою.

Часові діаграми верифікації пристрою приведено на рис. 3.6 та 3.7. Діаграми демонструють результати моделювання поведінкової моделі

обчислювача та коректність його роботи. На діаграмі (рис.3.6) показано вхідний та вихідний інформаційні сигнали x та y. На вхід пристрою був поданий бітовий потік x = 13 бітів (імпульсний сигнал). З них на вихід пристрою пройшло 4 біти, а отже, 1, 3, 7, 13 біти.

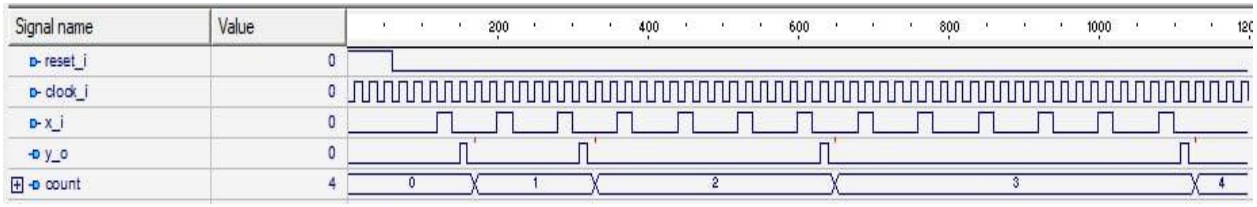

Рисунок 3.5 – Результати верифікації поведінкової моделі потокового обчислювача

На деталізований діаграмі (рис.3.7) показано що обчислювач працює в режимі вибірки бітів з вхідної послідовності. Номери бітів, що обрано, співпадають з розрахунками обчислень у компонентах пристрою. Значення регістрів компонентів обчислювача співпадають з результатами обчислень на рис. 3.3.

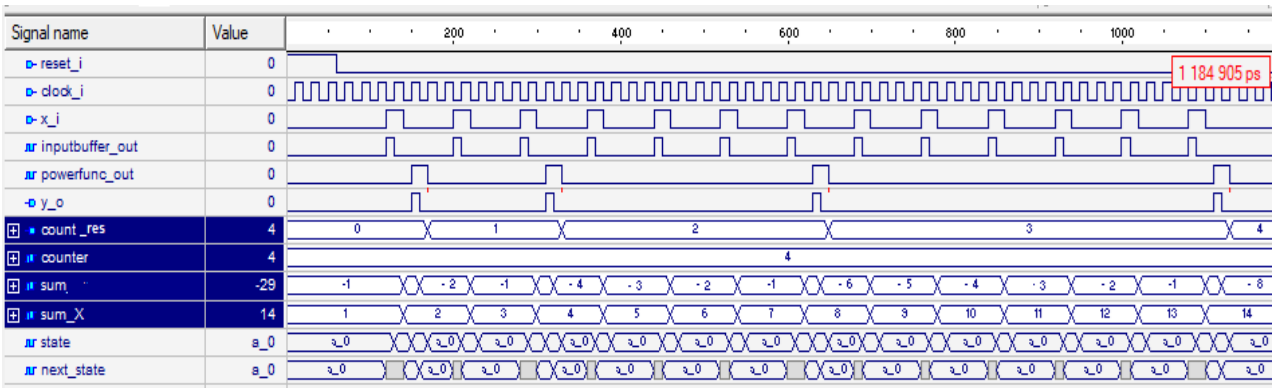

Рисунок 3.6 – Деталізована діаграма поведінкової моделі обчислювача вилучення кореня

Після верифікації проекту було виконано синтез і імплементацію розробленого обчислювача в платформу ПЛІС FPGA. Xilinx SPARTAN 3E серії xc3s500E. Максимальна частота, на якій працює обчислювач склада. 135 MHz. В проекті було використано 16 та 8 розрядні компоненти SUM, Count_ RES. При синтезі створено RTL-схеми вентильного рівня пристрою.

Характеристики використаного кристалу:

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

Нижче приведено звіт про мінімальний період сигналу синхронізації, максимальну частоту роботи пристрою.

Minimum period: 7.223ns (Maximum Frequency: 135.101MHz)

Number of External IOBs	11 out of 232	4%
Number of External Input IOBs	3	
Number of External Output IOBs	8	
Number of BUFGMUXs	1 out of 24	4%
Number of Slices	46 out of 4656	1%

На рис. 3.8 приведено RTL-схему пристрою верхнього рівня Powerfunc.top

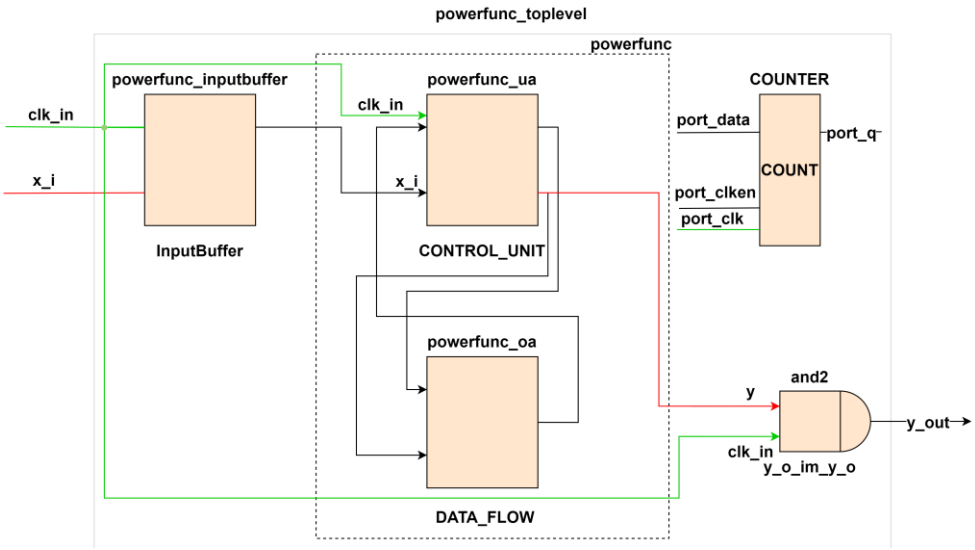

Рисунок 3.8 – RTL-схему пристрою верхнього рівня Powerfunc.top

ВИСНОВКИ

В кваліфікаційній роботі розроблено апаратний потоковий обчислювач функції добування кореня, що імплементований у платформу ПЛІС Xilinx. Spartan з використанням САПР на основі мов опису апаратури.

У роботі проведено огляд СУ реального часу, розглянуто схему сенсорної системи, дано визначення бітового потоку даних, розглянуто етапи проектування функціональних потокових обчислювачів.

В роботі проаналізовано математичну модель обчислювача функції вилучення кореня довільного степеня На підставі методу ступінчастої апроксимації на основі оберненої функції, проаналізовано математичну модель обчислювача функції квадратного кореня. Розроблено реконфігуровану архітектурну модель обчислювача на основі архітектури пристрою добування кореня. Архітектурна модель пристрою дозволяє спростити процес відтворення функції, за рахунок виключення з архітектури певних компонентів, що зменшує апаратурні витати та дає можливість отримання результату обчислення функції у вигляді бітового потоку та двійкового коду одночасно.

Розроблено автоматну модель обчислювача на основі кінцевого автомата Мура, розроблено граф переходів керуючого автомата та граф-схему алгоритма операційного автомата. Приведено результати теоретичних розрахунків для експериментальної частини апаратної реалізації проекту. Створено HDL-модель обчислювача з використанням HDL-шаблонів, що використовують автоматне програмування. Здійснено верифікацію поведінкової моделі спроектованого обчислювача, яка повністю підтвердила моделі проектування, та теоретичні розрахунки. Модель пристрою було синтезовано засобами САПР Xilinx. Для імплементації використано платформу FPGA Spartan 3E серії XC3S500E. Максимальна частота пристрою складає 135 MHz.

Апаратний потоковий обчислювач функції вилучення квадратного кореня може бути застосований в системах управління, контролю, інформаційно вимірювальних системах в якості функціональних перетворювачів інформаційних потоків у біт-потоковій формі, що отримують від сенсорів з частотним та імпульсним виходом.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Zhou F., Chai Y., “Near-sensor and in-sensor computing,” Nature Electronics, 2020, Vol. 3(11). P. 664–671. doi: https://doi.org/ 10.1038/s41928-020-00501-9.

2. Al-Makhles, D.; Patel, N.; Swain, A. Conventional and hybrid bit-stream in real-time system. / Proceedings of the 11th Workshop on Intelligent Solutions in Embedded Systems (WISES), 2015, pp. 1–6.

3. Shkil A. S., Larchenko L. V., and Larchenko B. D., Bit-stream power function online computer, Proceedings of the IEEE East-west design & test symposium (EWDTS), 2020, P. 1-6. doi: 10.1109/EWDTS50664.2020.9224764

4. Sajjad M., Yusoff M. b. Z. and Ahmed M., Design of Double-Precision Fully-Programmable Computational Unit for FPGA and ASIC, International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, P. 21-26.

5. Stanley, M.; Gervais-Ducouret, S.; Adams, J. T. Intelligent sensor hub benefits for wireless sensor networks. / Proceedings of IEEE Sensors applications symposium, 2016, pp. 643-648. doi: 10.1109/SAS.2012.6166299.

6. A.I. Gulin, N.M. Safyannikov, O.I. Bureneva, A.Yu. Kaydanovich. Assurance of Fault-Tolerance in Bit-Stream Computing Converters // Proceeding of 16th IEEE East-West Design & Test Symposium (EWDTS’2018). Kazan, Russia, September 14 – 17, 2018. – pp. 418 – 421.

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

8. Larchenko L.V., Parkhomenko A.V., Larchenko B.D., Korniienko V.R., Design Models of Bit-Stream Online-Computers for Sensor Components, Radio Electronics, Computer Science, Control, 2024, vol. 1(68), pp. 48-61, https://doi.org/10.15588/1607-3274-2024-1-6.

9. Dang Pham K., Horta E., Koch D. BITMAN: A Tool and API for FPGA Bitstream Manipulations / K. Dang Pham, E. Horta, D. Koch // 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar 2017, Lausanne, Switzerland, pp. 894-897. DOI: 10.23919/DATE.2017.7927114.

10. Korniienko V., Larchenko L., Parkhomenko A. et al. Design Models of Bit-Stream Online-Computers of Elementary Mathematical Functions [Text] / V. Korniienko, L. Larchenko, A. Parkhomenko et al. // 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Technology and Applications (IDAACS), Sep 2023, Dortmund, Germany. – Germany: P. 585-589. IDAACS), Sep 2023, Dortmund, Germany, pp. 585-589. DOI: 10.1109/IDAACS58523.2023.10348885