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)

Функціональний пристрій перетворення імпульсних потоків
(тема)

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

_____Олефір А.В._____
(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 _доц. Ларченко Л.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)

«__05__»___05_____2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту_____Олефіру Антону Вікторовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Функціональний пристрій перетворення імпульсних потоків_____
_____

затверджена наказом по університету 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):

Функціональне перетворення сигналів сенсорних компонентів в СУ

Математична модель пристрою перетворення імпульсних потоків

Архітектура та автоматна модель пристрою

Апаратна реалізація функціонального пристрою перетворення імпульсних потоків на ПЛІС

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
14 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження	05.05.2025 -07.05.2025	
2	Аналіз проблемної галузі, постановка задачі	08.05.2025 -12.05.2025	
3	Аналіз декомпозиції математичної моделі функціонального пристрою	12.05.2025 -15.05.2025	
5	Розроблення архітектури пристрою заданої функції з використанням конвеєрних архітектур	16.05.2025 -.19.05.2025	
5	Розробка автоматної моделі пристрою: графових моделей кінцевгого автомату.	20.05.2025 -24.05.2025	
6	Розробка HDL-моделі пристрою, верифікація та імплементація проекту в ПЛІС	25.05.2025 -02.06.2025	
7	Оформлення пояснювальної записки	03.06.2025 -08.06.2025	
8	Перевірка виконаного проекту керівником,	09.06.2025 -14.06.2025	
9	Захист проекту	15.06.2025 -23.06.2025	

Дата видачі завдання 05.05.2025

Здобувач _____
(підпис)

Керівник роботи (проекту) _____ доц. Ларченко Л.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 51 сторінок, 12 рисунків, 10 джерел за переліком посилань.

ФУНКЦІОНАЛЬНЕ ПЕРЕТВОРЕННЯ, ІМПУЛЬСНИЙ ПОТІК, АПРОКСИМАЦІЯ, АРХІТЕКТУРА, АВТОМАТНА МОДЕЛЬ, КІНЦЕВИЙ АВТОМАТ, ГРАФ-СХЕМА АЛГОРИТМА, HDL-МОДЕЛЬ, ПЛІС

Метою кваліфікаційної роботи є розробка пристрою функціональної обробки імпульсних потоків на платформі ПЛІС.

У роботі розглянуто математичну модель пристрою, що отримана з використанням методу ступінчастої апроксимації на основі обернених функцій. Розроблено архітектуру пристрою, здійснено вибір платформи ПЛІС для імплементації проекту. Апаратну реалізацію пристрою здійснено на основі цифрового автомата Мура, розроблено HDL-модель пристрою з використанням HDL-шаблонів та здійснено імплементацію обчислювального модуля в платформу ПЛІС Xilinx Spartan.

Спроектований функціональний пристрій перетворення імпульсних потоків може знайти застосування в системах управління та вимірювань в якості функціональних online-перетворювачів імпульсних потоків, що отримують від сенсорів фізичних величин з частотним або широтно-модульованим вихідним сигналом.

ABSTRACT

Explanatory note: 51 pages, 12 figures, 10 sources according to the list of links.

FUNCTIONAL CONVERSION, PULSE STREAM, APROXIMATION, ARCHITECTURE, AUTOMATA MODEL, FINITE STATE MACHINE, GRAPH DIAGRAM, HDL-MODEL, FPGA

The purpose of the qualification work is the development of a device for functional processing of pulse streams on the FPGA platform.

The paper considers the mathematical model of the device, obtained using the stepwise approximation method based on inverse functions. The device architecture was developed, the FPGA platform was selected for project implementation. The hardware implementation of the device is based on Moore's digital automaton, an HDL model of the device was developed using HDL templates, and the computing module was implemented in the Xilinx Spartan FPGA platform.

The designed functional device for converting pulse flows can be used in control and measurement systems as functional online converters of pulse flows received from sensors of physical quantities with a frequency signal or width-modulated signal.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8
ВСТУП 9
1 ФУНКЦІОНАЛЬНЕ ПЕРЕТВОРЕННЯ СИГНАЛІВ СЕНСОРНИХ КОМПОНЕНТІВ В СИСТЕМАХ УПРАВЛІННЯ РЕАЛЬНОГО ЧАСУ 10
1.1 Особливості функціонального перетворення імпульсних сигналів в системах управління 10
1.2 Метод ступінчастої апроксимації на основі обернених функцій 16
1.3 Мета та задачі проектування 17
2 МАТЕМАТИЧНА МОДЕЛЬ ПРИСТРОЮ ФУНКЦІОНАЛЬНОГО ПЕРЕТВОРЕННЯ ІМПУЛЬСНИХ ПОТОКІВ 19
2.1 Математичне обгрунтування функціональної обробки імпульсних потоків 19
2.2 Математична модель функціонального пристрою 21
3 АРХІТЕКТУРА ТА АВТОМАТНА МОДЕЛЬ ПРИСТРОЮ 25
3.1 Архітектура функціонального пристрою перетворення імпульсних потоків 25
3.2 Автоматна модель пристрою 30
3.3 Вібір технологічної платформи для імплементації проекта 33
4 АПАРАТНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛЬНОГО ПРИСТРОЮ 37
4.1 Результати теоретичних обчислень для експерименту 37
4.2 Блок-схема функціонального пристрою 40
4.3 Опис HDL-проекту 41
4.4 Верифікація та імплементація функціонального пристрою 46
ВИСНОВКИ 49
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 50
ДОДАТОК А Графічна частина 52

ДОДАТОК Б Розрахунки значень заданої функції та обчислювального процесу у компонентах архітектури.....59
ДОДАТОК В Лістинг HDL-моделі функціонального пристрою.....62

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;

ЦФП – цифровий функціональний перетворювач;

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 (одна з мов опису апаратури);

ВСТУП

Сьогодні спостерігається подальший стрімкий розвиток систем реального часу, які включають розподілені системи управління, контролю та вимірювань, що містять підсистеми функціональної обробки сигналів сенсорних компонентів фізичних величин. В розподілених системах управління сенсорні компоненти розташовані на великій відстані від центальної обчислювальної системи. При цьому в системах управління є потреба в розробці потокових процесорів, перетворювачів та обчислювачів які здійснюють потокову функціональну обробку сигналів поблизу сенсорів, тому тема кваліфікаційної роботи є актуальною. Серед сенсорів, що застосовуються в системах управління иа вимірювальних системах знаходять застосування сенсори з частотним або час-імпульсним вихідним сигналом, які представляють собою імпульсний потік.

Метою кваліфікаційної роботи є розробка функціонального пристрою, що обробляє інформаційні сигнали у вигляді серій імпульсних потоків. Пристрій створено на основі степеневого біт-потокового обчислювача, який здійснює паралельно-послідовне перетворення імпульсних потоків. В роботі приведено математичну модель пристрою, що базується на методі ступінчастої апроксимації на основі оберненої функції, розроблено архітектурну та автоматну моделі потокового функціонального пристрою. Спроектований пристрій може бути застосований в якості функціонального real-time перетворювача в системах управління, контролю та вимірювань в якості цифрових функціональних перетворювачів, а також в системах вимірювання вібрації обертового обладнання при вимірюванні середньоквадратичних значень віброшвидкості. Пристрій імплементовано в платформу ПЛІС та досліджено його частотні характеристики.

1 ФУНКЦІОНАЛЬНЕ ПЕРЕТВОРЕННЯ СИГНАЛІВ СЕНСОРНИХ КОМПОНЕНТІВ У СИСТЕМАХ УПРАВЛІННЯ РЕАЛЬНОГО ЧАСУ

В розділі розглянуто особливості потокового функціонального перетворення сигналів сенсорних компонентів, застосування спеціалізованих функціональних пристроїв потокової обробки імпульсних сигналів в системах управління, сформульовано мету та задачі проектування.

1.1 Особливості функціонального перетворення імпульсних сигналів в системах управління

В даний час у розподілених системах управління (СУ) реального часу необхідно забезпечувати безперервний прийом та обробку потоків інформаційних даних у міру їх надходження.

Актуальним завданням для розвитку СУ реального часу є створення потокових пристроїв, що виконують функціональне перетворення сигналів сенсорних компонентів, представлених у формі імпульсного потоку. При цьому необхідно здійснювати збір та лінеаризацію сигналів, що передбачає перетворення сигналів за певними функціями.

На даний час отримала розвиток концепція присенсорних обчислень, що передбачає переміщення функціональної обробки даних ближче до сенсорних компонентів, щоб знизити обсяг переданих даних та навантаження на канали зв’язку. При цьому широке застосування знаходять сенсори, що генерують вихідний сигнал у вигляді частотного, час-імпульсного сигналу або сигналу широтно-імпульсної модуляції.

Для розподілених СУ є характерним та доцільним перетворення аналогових сигналів сенсорних компонентів у частотний сигнал, а не аналого-цифрове перетворення, так як частотні сигнали мають підвищену завадостійкість у порівнянні з двійковими кодами [1].

Частотний сигнал - потік імпульсів, що має одиничну амплітуду, в якому інформативним параметром є фіксоване значення імпульсів продовж певного часового інтервалу.

Представлення інформаційних сигналів у формі імпульсних потоків дозволяє виконувати обчислювальні операції простішими способами, що впливає на побудову архітектур, які у порівнянні з архітектурами код-потокових пристроїв, де вхідний сигнал представлений двійковим кодом, мають меншу складність.

Функціональне перетворення сигналу «біля сенсору» потребує здійснення перетворення у такому самому форматі даних, що поступають на вхід перетворювача з виходу сенсорного компонента. При використанні сенсорів з частотним або імпульсним виходом аналогові сигнали перетворюються у формат імпульсного сигналу, який можна використовувати у потокових пристроях функціонального перетворення. При цьому існує потреба у створенні спеціалізованих нескладних, швидкодіючих, надійних пристроях, які можуть виконувати лінеаризацію сигналів за необхідною функцією у реальному часі. Прикладом таких функціональних пристроїв є елементи процесора на основі імпульсної обробки [2].

У зв’язку з необхідністю потокової функціональної обробки сигналів у реальному часі актуальним завданням є створення потокових пристроїв - перетворювачів та обчислювачів, що виконують функціональне перетворення сигналів, представлених у формі імпульсного потоку. Основною метою побудови архітектур пристроїв, які працюють з імпульсною формою сигналів є спрощення їх структур та компонентів за рахунок використання сигналів у вигляді потоку імпульсів, тому що виконання обчислювальних операцій з імпульсною формою вхідних сигналів здійснюється простішими способами.

Для імплементації спеціалізованих функціональних модулів, що здійснюють присенсорні обчислення застосовуються різні технологічні платформи, зокрема, FPGA, CPLD, спеціалізовані інтегральні мікросхеми ASIC, що забезпечують надійність, гнучкість та енергоефективність [3].

Функціональне перетворення сигналів, які отримують від сенсорів СУ здійснюється за схемою рис. 1.1.

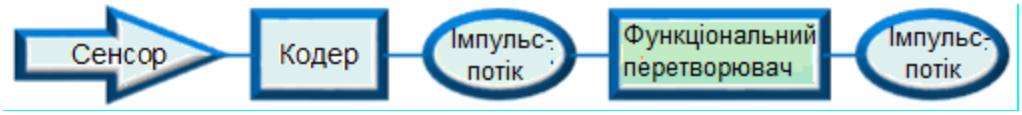

Рисунок 1.1 – Схема функціонального перетворення сигналів

Сенсор сприймає зовнішні фізичні сигнали аналогові, кодер перетворює аналоговий сигнал в імпульсних потік, який подається на вхід функціональний перетворювач, що здійснює лінеаризацію сигналу за заданою функцією. Виконавши лінеаризацію імпульсного сигналу, на виході перетворювача з'являється імпульсний потік, перетворений за необхідною функцією. Надалі, імпульсний потік з вихода перетворювача поступає на декодер і далі декодер видає сигнал керування на об'єкт управління (ОУ).

Потокові функціональні пристрої побудовано на основі потокового способу обчислень, що здійснює паралельно-послідовне виконання перетворень над імпульсами потоку, при цьому архітектура пристроїв містить базові швидкодіючі конвеєрні структури.

При потоковому способі обчислень в пристрою потокової обробки обчислення значень функції отримують для сусідніх значень аргументу. Потокові способи функціональної обробки імпульсних сигналів реалізують з використанням методів формування приростів, зокрема, принципово-точних на основі обернених функцій. Принципи організації алгоритму потокової обробки функціональних пристроїв розглядаються як безперервний процес формування результату.

Основою сучасних розподілених СУ є обчислювальна система, яка реалізує основні функції СУ, що здатні здійснювати прийом даних від сенсорів та аналізувати дані про ОУ; здійснювати обробку даних, їх порівняння із задачами управління ОУ та генерувати відповідні сигнали керування на ОУ.

Компонентами СУ є ОУ, датчики, сенсори, перетворювачі, виконавчі пристрої – актуатори, та підсистема управління реального часу (рис.1.2).

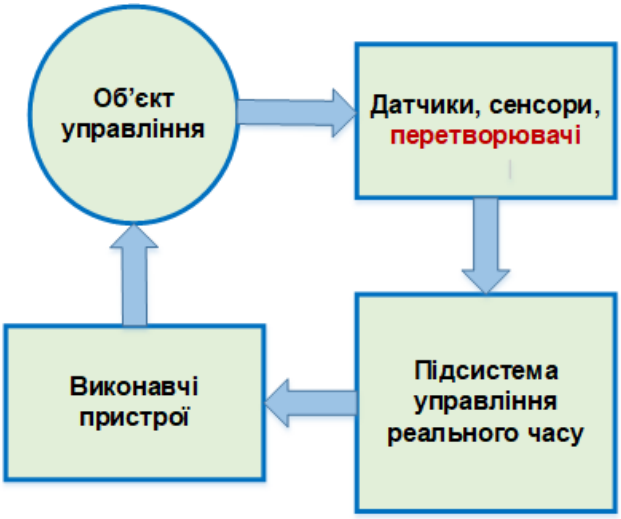

Рисунок 1.2 – СУ та взаємодія її компонентів

Складовими інтерфейсу зв'язку є ОУ, сенсори, а також перетворювачі, які є спеціалізованими пристроями оброблення сигналів. Функціональні перетворювачі здійснюють лінеаризацію сигналів за певними елементарними математичними функціями. Функціональні пристрої перетворення сигналів працюють з різними формами даних, у тому числі і з потоковими, що являють собою бітові – імпульсні потоки.

У складі СУ є підсистеми первинного і вторинного оброблення інформаційних сигналів. Призначення підсистем первинного оброблення є перед процесорна обробка сигналів: декодування, демодуляція, лінеаризація сигналів та інші способи оброблення сигналів.

При первинній функціональній обробці сигналів необхідно забезпечувати безперервний процес перетворення - надходження на вхід пристроїв перетворення сигналів - даних, їх обробка та формування результату. Всі названі операції маєть здійснюватись у реальному масштабі часу[4].

При виконанні обчислювальних операцій у перетворювачах на виході пристрою формується новий імпульсний (бітовий) потік, який пов'язаної

певною функціональною залежністю з імпульсним вхідним потоком. Імпульсні потоки можуть бути у вигляді одного потоку, або серій імпульсних потоків.

Підсистеми первинної обробки інформаційних сигналів з певними обчислювальними ресурсами повинні мати досить високу продуктивність, що забезпечується за рахунок паралелізму алгоритмів реазізації завдань. Сучасні розподілені інформаційно – керуючі системи працюють у реальному часі, забезпечують технічну досконалість архітектури та можливість нарощування продуктивності засобів обробки даних, вбудованість апаратних засобів безпосередньо в ОУ та підвищення стійкості СУ.

При здійсненні вимірювання таких параметрів процесів, як рівень фізичного середовища, рівня рідіни, величини щільності, тиску, температури знаходять широкк використання сенсори з частотним та імпульсним вихідним сигналом. Важливим напрямом підвищення точності та надійності вимірювання параметрів є розвиток функціональних можливостей перетворювачів, що здійснюють первинне перетворення та вторинну математичну обробку імпульсних та частотних сигналів при вирішенні завдань лінеаризації [5].

Функціональні біт-потокові обчислювачі, що розглядаються, побудовані на основі цифрового функціонального перетворення, що передбачає послідовне обчислення значень функції, що виконуються для сусідніх значень аргументу. Таке перетворення має назву розгортуючого. У розгортках є можливість враховувати попередню історію обчислень: кожне чергове значення відтворюваної функції обчислюють на основі попереднього обчислення. В таких пристроях перше обчислення функції здійснюється з врахуванням початкових значень [6].

Серед датчиків сучасних СУ, систем контролю та вимірювань значну частину становлять частотні датчики, в яких вихідним інформативним параметром є частота імпульсної послідовності. Тому актуальними залишаються проблеми вимірювання частотно-часових параметрів сигналів. Використання частотних датчиків забезпечує більш просте перетворення

частоти в цифровий код. Такі завдання ставляться при вимірюванні постійного і диференціального тиску у трубопроводах, в авіаційній техніці при вимірюванні вібрацій та в системах, де необхідно вимірювати рівень рідини.

Широке застосування функціональні перетворювачі імпульсних потоків знаходять використання при проектуванні вимірювальних засобів та пристроїв, в яких використовуються функціональні перетворювачі в якості обчислювальних вузлів та блоків.

Функціональний пристрій перетворення імпульсних потоків, може знайти застосування:

а) в системах управління та контролю в якості функціональних потокових процесорів при обробці частотних, час-імпульсних та широтно-модульованих сигналів;

б) в СУ реального часу у підсистемах функціонального оброблення імпульсних сигналів від сенсорних компонентів при вимірюванні середньоквадратичних значень при оцінці параметрів вібрації обертового обладнання;

в) у вимірювальних підсистемах СУ при вимірювання вібрації обертового обладнання при вимірюванні середньоквадратичних значень віброшвидкості та віброприскорення агрегатів з обертовими частинами .

Для імплементації спеціалізованих функціональних модулів, що здійснюють присенсорні обчислення застосовуються технологічні платформи FPGA, CPLD, спеціалізовані інтегральні мікросхеми ASIC, що забезпечують надійність, гнучкість та малу енергоефективність [7]. Також пристрої функціональної потокової обробки можуть підійти для розгортання на системах на кристалі (SoC), таких як Xilinx ZYNQ, що містять частину системи програмування (ARM) та компонент програмованої логіки (PL) з оптимальним розподілом обчислень між ними, використання часткової реконфігурації для розгортання різних функціональних перетворювачів імпульсного потоку, з можливостями управління ресурсами на стороні ARM.

1.2 Метод ступінчастої апроксимації на основі обернених функцій

Одним з методів, що використовується при побудові потокових пристроїв функціонального перетворення є принципово-точний метод, оснований на визначенні обернених функцій. При цьому застосовуються різні апроксимуючі функції, що апроксимують безперервні. Визначальним при відтворенні математичних функцій пристроями функціональної потокової обробки є абсолютна похибка апроксимації, яка впливає на результат відтворення функцій. Принципово-точні методи не мають методичної похибки апроксимації, тому дають кращий результат за точністю обчислення (відтворення) функції.

У роботі [8] розглянуто метод ступінчастої апроксимації безперервних функцій на основі обернених функцій, що є принципово-точним. Абсолютна похибка вітворення функції забезпечується мінімальною величиною, яка дорівнює 0,5 одиниці молодшого біту аргументу.

Безперервна функція $y^* = f(x^*)$, яка може бути відтворена на виході потокового функціонального пристрою, є висхідною, $x^*, y^* \geq 0$; $y^* \leq x^*$ та має обернену функцію $x^* = \psi(y^*)$ може бути відтворена на виході пристрою за допомогою апроксимуючою функцією:

$$y = [f(x) + |\delta_{max}|], \tag{1.1}$$

де x, y – вхідний та вихідний імпульсні потоки

У функціональному потоковоому пристрою та вихідний y сигнали x та y - імпульсні потоки.

Формула для визначення вибіркових значень x_y, що обираються з вхідного імпульсного потоку x та відповідають рівням вузлів апроксимації відтворюваної ступінчастої функції має вигляд:

$$\Psi(y-|\delta_{max}|) \le x_y < \Psi(y-|\delta_{max}|)+1 \quad (1.2)$$

де x, y – вхідний та вихідний імпульсні потоки, y, x = 1,…,i;

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

Якщо похибка відтворення функції є мінімальною, то $|\delta_{max}| = 0{,}5$, формула для знаходження значень x_y мають вигляд:

$$\Psi(y-0{,}5) \le x_y < \Psi(y-0{,}5)+1 \quad (1.3)$$

Формула (1.3) є основою для розроблення математичних моделей функціональних потокових перетворювачів імпульсних та частотних сигналів.

1.3 Мета та задачі проектування

Метою кваліфікаційної роботи є розроблення спеціалізованого функціонального пристрою перетворення імпульсних потоків на технологічній платформі ПЛІС з використанням мов опису апаратури.

Об’єктом розробки є функціональні пристрої перетворення з імпульсними потоками даних. Предметлм розробки є математичні та автоматні моделі функціональних пристроїв потокової обробки, а також конвеєрні архітектури потокових пристроїв.

Для досягнення мети було визначено нступні задачі:

– аналіз функціонального перетворення імпульсних потоків сенсорних компонентів в системах управління;

– аналіз математичної моделі пристрою функціонального перетворення імпульсних потоків;

– розробка архітектури потокового функціонального пристрою;

– розробка автоматної моделі пристрою;

– вибір платформи для реалізації проекту;

– виконання теоретичних розрахунків для проведення екперименту;

– розробка HDL-моделі пристрою на основі HDL-шаблонів;

– верифікація поведінкової моделі та імплементація пристрою з використанням САПР Xilinx ISE в платформу ПЛІС.

2 МАТЕМАТИЧНА МОДЕЛЬ ПРИСТРОЮ ФУНКЦІОНАЛЬНОГО ПЕРЕТВОРЕННЯ ІМПУЛЬСНИХ ПОТОКІВ

2.1 Математичне обґрунтування функціонального перетворення імпульсних потоків

При функціональному перетворенні імпульсних потоків, при визначенні середньоквадратичних значень віброшвидкості при вимірюванні вібрації, періодичних несинусоїдних струмів і напруг, пристрій перетворення відтворює дробово-ірраціональну функцію. Крім того, в спеціалізованих функціональних перетворювачах дробово-ірраціональна функція реалізується в аналізаторах спектра частотно-модульованих сигналів.

Реалізацію апроксимуючих дробово-ірраціональних функцій, аргумент якіих проедставлений імпульсним потоком може бути здійснено на основі двох функціональних пристроїв:

– біт-потокового дробово-раціонального обчислювача. Недоліком пристрою є обмеження частоти вхідного імпульсного потоку внаслідок утворення на виході дробово-раціонального пристрою серій імпульсних потоків, який є внутрішнім проміжним сигналом пристрою;

– біт-потокового степеневого обчислювача. Перевагою пристрою на основі степеневого обчислювача є відсутність генерації на його виході серій імпульсних потоків, так як пристрій працює в режимі вибірки імпульсів з вхідного потоку, що дозволяє розширити частотний діапазон вхідного сигналу функціонального пристрою.

У зв’язку з названою перевагою в роботі було обрано функціональний пристрій перетворення імпульсних потоків на основі біт-потокового обчислювача степеневих функцій.

Дробово-ірраціональна апроксимуюча функція, що має бути відтворена на виході функціонального пристрою має вигляд:

$$y = [\frac{1}{p}[\sqrt{\sum_{i=1}^{n} x_i^2} + |\delta_{2max}|] + |\delta_{1max}|], \quad (2.1)$$

де x_i – серії імпульсних потоків;

n – кількість серій імпульсних потоків вхідного сигналу;

p – знаменник кутового коефіцієнта, який дорівнює $n = p^2$, ціле число, може бути винесений за знак квадратичного радикалу;

$|\delta_{1max}|$ – абсолютна похибка при діленні $\sqrt{\sum_{i=1}^{n} x_i^2}$ на число p;

$|\delta_{2max}|$ – абсолютна похибка, шо виникає при вилученні кореня.

Для того, щоб забезпечити мінімальну похибку апроксимації при обчисленні вилучення кореня та ділення числа на константу p, похибки обчислення дорівнюють $|\delta_{1max}| = |\delta_{2max}| = 0{,}5$:

$$y = [\frac{1}{p}[\sqrt{\sum_{i=1}^{n} x_i^2} + 0{,}5] + 0{,}5] \quad (2.2)$$

Вхідним інформаційним сигналом обчислювача є бітовий потік x - серії бітів, на виході пристрою формується бітовий потік y, що відтворює задану ірраціональну функцію.

Відтворення дробово-ірраціональної функції у пристрою здійснюється за допомогою двох етапного обчислення при перетворенні вхідного імпульсного потоку.

1. На першому етапі здійснюється піднесення аргументу, що являє собою серії імпульсних потоків x_i до другого степеня і вилучення кореня, тобто

обчислюється степенева функція з дробовим показником. Ці дві обчислювальні операції виконуються за допомогою біт-потокового степеневого обчислювача.

2. На другому етапі виконується поділ проміжного отриманого результату обчислення степеневої функції на ціле число p. Дана обчислювальна операція може бути здійснена в дільнику чисел, що реалізований на біт-потоковому обчислювачі лінійної функції.

2.2 Математична модель функціонального пристрою

Враховуючи двох етапне обчислення заданої функції у функціональному пристрою перетворення імпульсних птоків і те, що її реалізація може бути здійснена у пристроях обчислення степеневої та лінійної функцій, математична модель потокового функціонального пристрою, що реалізує апроксимуючу дробово-ірраціональну функцію представлена декомпозицією, що включає математичні моделі названих пристроїв, які базуються на методі ступінчастої апроксимації на основі обернених функцій.

1. На першому етапі функціональним пристроєм реалізується проміжна апроксимуюча степенева функція другого порядку, аргумент якої представлений серіями імпульсних потоків. Функція має вигляд:

$$y = [\sqrt{\sum_{i=1}^{n} x_i^2} + 0,5]. \qquad (2.3)$$

Математичну модель степеневого обчислювача розглянуто вхідний сигнал у вигляді однієї серії імпульсного потоку [9].

Степенева апроксимуюча функція:

$$y = [\sqrt{x^2} + 0,5] \qquad (2.4)$$

Визначивши обернену функцію для (2.4) та скорставшись формулою (1.3) нерівність, що реалізується у пристрою має вигляд:

$$2^2 x_y^2 \geq (2y_k - 1)^2. \tag{2.5}$$

Переходячи до приростів функцій лівої та правої частин (2.5), математична модель степеневого обчислювача являє собою систему нерівностей, що виражені через різниці:

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

де Δ_{y-1} – різниця, що визначається на попередньому кроці в результаті порівняння приростів поточних значень функцій $2^2 x^2$ и $(2y-1)^2$, які визначаються між двома сусідніми вузлами апроксимації;

Δ_1- різниця, яка може бути отримана при порівнянні першої нерівності (2.6).

При цьому $1 \leq y_k \leq k$; x_0, $\Delta_0 = 0$.

В (2.6) значення Δ_1 и Δ_{y-1} визначаються за формулами:

$$\begin{gathered} \Delta_1 = 2^2 x_1^2 - (2y_1 - 1)^2, \\ \Delta_{y-1} = 2^2 (x_y^2 - x_{y-1}^2) + \Delta_{y-2} - (2y_k - 1)^2 + (2y_{k-1} - 1)^2. \end{gathered}$$

Значення x_y можуть бути визначені при обчисленні приростів ґратчастої функції $2^2 x_y^2$ на кожному з інтервалів сходинок апроксимуючої функції та їх

порівнянні з приростами ґратчастої функції $(2y_k - 1)^2$ з урахуванням різниці Δ_{y-1} обчислення, яка отримана на попередньому кроці обчислень.

Другий етап обчислень (перетворень) у спроектованому пристрою передбачає здійснення ділення проміжного результату обчислення степеневої функції на константу p. При цьому враховується похибка апроксимації $|\delta_{max}| = 0{,}5$. Обчислення реалізовано з використанням обчислювача лінійних функцій, що підключається до виходу степеневого обчислювача, який буде відтворювати функцію:

$$y = [\frac{1}{p} x + 0{,}5], \tag{2.7}$$

де x – імпульсний (бітовий) потік.

Математична модель обчислювача лінійних функцій отримано на підставі методу ступінчастої апроксимації на основі обернених функцій.

де x – імпульсний (бітовий) потік.

Отже, визначивши обернену функцію для (2.7), з урахуванням похибки $|\delta_{max}| = \frac{1}{2}$, нерівність для виначення вибіркової значень x_y має вигляд:

$$\frac{p(2y-1)}{2} \le x_y < \frac{p(2y-1)}{2} + 1, \tag{2.8}$$

де y = 1, 2, 3, ... – вихідний потік імпульсів; x_y - номер обраного біту (імпульсу), який з’явиться на виході пристрою при виконанні нерівності (2.9). Математичну модель обчислювача лінійних функцій отримано на основі оберненої функції.

Після перетворень, нерівність, що має бути реалізована в пристрою на другому етапі обчислень має вигляд:

$$2x_y \geq p(2y_k - 1) \quad . \tag{2.9}$$

Якщо у правій частині нерівності (2.9) y_k приймає значення $1, 2, 3, \ldots,$ то можна записати арифметичний ряд: $p, 3p, 5p, 7p, 9p$ і т.д. При обчисленні ряду першої різниці, отримано константу $2p$.

Математична модель дільника чисел, що представляє собою потоковий обчислювач лінійних функцій, представляється у вигляді різницевих нерівностей та має вигляд:

$$\begin{gathered} 2x_1 \geq p, \\ 2(x_2 - x_1) + \Delta_1 \geq 2p, \\ 2(x_3 - x_2) + \Delta_2 \geq 2p, \\ \ldots\ldots\ldots\ldots\ldots, \\ 2(x_y - x_{y-1}) + \Delta_{y-1} \geq 2p \end{gathered} \quad , \tag{2.10}$$

де

$$\Delta_{y-1} = 2(x_y - x_{y-1}) + \Delta_{y-2} - 2p \quad . \tag{2.11}$$

При виконанні нерівностей математичної моделі пристрою обчислення лінійної функції на його виході з'являються вихідні біти *y*.

У розділі розглянуто математичні моделі пристроїв обчислення степеневої та лінійної функцій, що лежать в основі побудови архітектури функціонального пристрою, що реалізує дробово-ірраціональну функцію.

3 АРХІТЕКТУРА ТА АВТОМАТНА МОДЕЛЬ ПРИСТРОЮ

У даному розділі розглнуто узагальнену архітектуру біт-потокових обчислювачів дробово-ірраціональних функцій, що реалізує двох етапне обчислення дробово-ірраціональної функції. На її основі синтезовано архітектуру спроектованого пристрою, складовими якої є обчислювач степеневої функції та дільник числел, побудований на пристрої обчислення лінійної функції. Приведено опис спроектованого пристрою та обрано платформу для його апаратної реалізації.

3.1 Архітектура функціонального пристрою перетворення імпульсних потоків

При розробленні функціонального пристрою потокової обробки, що відтворює дробово-ірраціональну функцію і враховуючи декомпозицію математичної моделі було побудовано архітектуру пристрою на базі загальної архітектури біт-потокового пристрою обчислення дробово-ірраціональної функції, яку наведено на рис. 3.1.

Загальна архітектура містить 2 модулі:

а) перший модуль являє собою архітектуру біт-потокового обчислювача степеневих функцій. Модуль містить Block1, Block2 та суматор SM_int. На виході SM_int формується бітовий потік, що є результатом обчислення проміжної степеневої функції $y_{int} := f(x)$;

б) другий модуль представляє собою архітектуру дільника чисел, побудовану на основі біт-потокового обчислювача лінійної функції та реалізує другий етап обчислень – ділення значення проміжної функції на константу. Дільник містить блок Block3 і суматор результату SM_RES. На виході SM_RES формується бітовий потік, що є результатом обчислення дробово-ірраціональної функції.

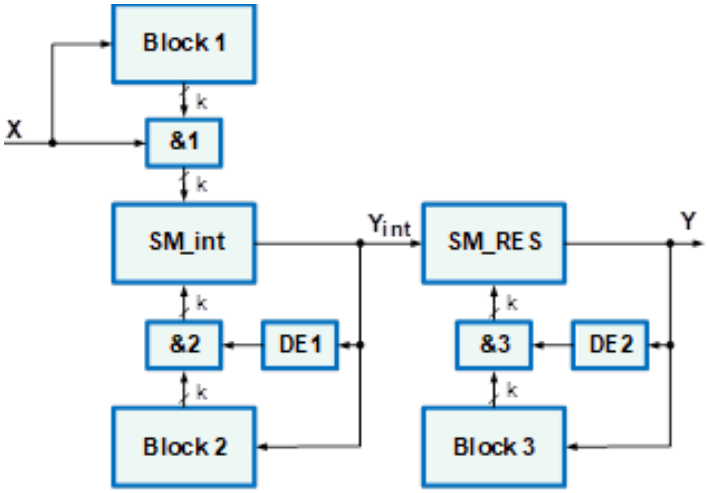

Рисунок 3.1 – Загальна архітектура біт-потокового пристрою

Block1 і Block2 - конвеєрні архітектури, які включені у прямий та зворотний зв’язок архітектури степеневого обчислювача, що є складовою частиною загальної архітектури.

На вхід пристрою подаються серії імпульсних (бітових) потоків x_i. На виході SM_int формується результат обчислення проміжної степеневої функції – вихідний бітовий потік y_{int}, що подається на вхід SM_RES, який є компонентом дільника чисел. На виході SM_RES формується вихідний бітовий потік, що є результатом обчислення дробово-ірраціональної функції *y*. Біти переповнення SM_RES відповідають формуванню сходинок апроксимуючої функції спроектованого пристрою.

Основними обчислювальними компонентами архітектури є суматори SM_int і SM_RES зі зворотним зв'язком, що використовуються в якості елемента, що порівнянює паралельні коди чисел лівої та правої частин нерівностей математичних моделей степеневого обчислювача та дільника чисел на кожному кроці порівняння, коли реалізується чергова нерівність відповідної математичної моделі. У суматор SM_int з Block1 здійснюється перенос прямих паралельних двійкових кодів чисел, а с Block2 – додаткових кодів чисел [10].

Далі розглянуто архітектуру розробленого пристрою перетворення імпульсних потоків, що створена на підставі загальної архітектури біт-

потокового пристрою (рис. 3.2).

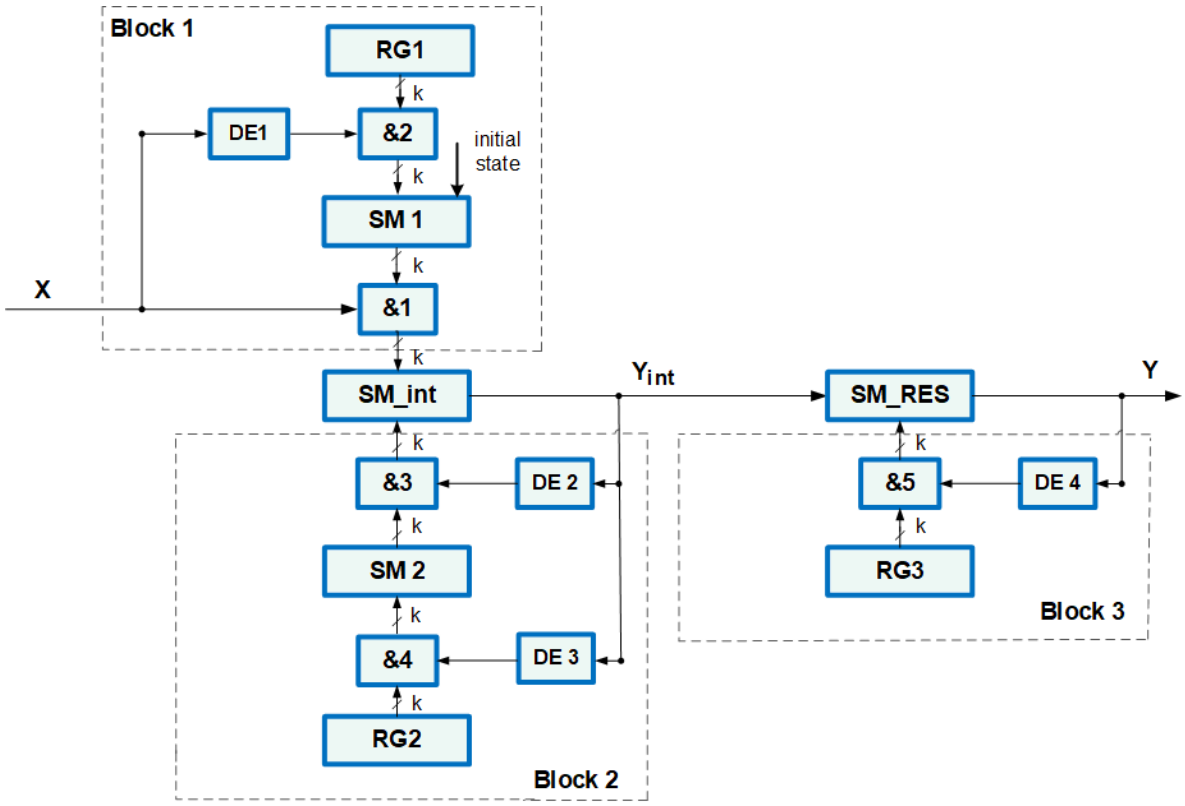

Рисунок 3.2 - Архітектура функціонального пристрою перетворення імпульсних потоків

Архітектура функціонального пристрою містить дві складові, що розлянуті нижче.

I. Степеневий біт-потоковий обчислювач має у складі у Block1 пристрій квадратор, що здійснює піднесення до другого степеня серій імпульсних потоків. Він включає суматори SM1, SM_int, регістр RG1, групи елементів &1, &2. У Block2 – пристрій вилучення квадратного кореня, побудований на компонентах SM_int, SM1, регістр RG2, групи елементів &3, &4, елементи затримки DE1, DE2, DE3.

SM_int – спільний компонент архітектури, на виході якого формується бітовий потік обчислення степеневої функції Y_{int}.

Степеневий обчислювач обчислює функцію (2.3). Основна нерівність, яка

реалізується в обчислювачі (2.5).

II. Дільник чисел має у складі суматор SM_RES та Block3, що включає регістр RG3, елемент затримки DE4 та групу логічних елементів &5.

Дільник чисел обчислює функцію (2.7). Основна нерівність, яка реалізується в обчислювачі (2.9).

Вхідним та вихідним інформаційними сигналами є імпульсні потоки x та y відповідно. Вхідний імпульсний потік подається на вхід пристрою у вигляді серій імпульсного потоку.

У всіх компонентах архітектури мають бути встановлені початкові значення – числа ініціалізації компонентів, перед подачею на вхід імформаційного сигналу X.

Ініціалізація компонентів степеневого обчислювача здійснена певними числами у двійковому коді:

– суматор SM_int: додатковий двійковий код $2^i - 1$;

– суматор SM1: двійковий код 1, тобто перше значення арифметичного ряду першої різниці зі сторони Block1, квадратора;

– суматор SM2: двійковий код числа, що є першим значенням ряду першої різниці зі сторони Block2, пристрою вилучення квадратного кореня;

– регістр RG1: двійковий код 2!, що утворюється в арифметичному ряду другої різниці функції x^2;

– регістр RG2: двійковий код константи в ряду різниць 2-го порядку функції $(2y-1)^2$ зі сторони пристрою вилучення кореня.

Ініціалізація компонентів дільника чисел здійснюється двійковими кодами чисел:

– суматор SM_RES: додатковий код числа $2^i - p$;

– регістр RG3: додатковий код числа $2^i - 2p$.

Вихід елемента затримки DE1 з'єднаний з групою елементів &2, за допомогою якої з регістра RG1 в SM1 кожним вхідним бітом послідовності x переноситься паралельний двійковий код числа 2 та підсумовується з

двійковим кодом числа, що міститься у SM1.

При надходженні на вхід степеневого обчислювача серії імпульсного потоку x в SM1 будуть утворюватися наступні значення ряду першої різниці функції x^2.

Перенос прямого двійкового коду числа в SM_int, який утворюється в суматорі SM1 необхідно виконувати зі зсувом на два розряди вправо, що відповідає множенню на 2^2 в лівій частині нерівностей математичної моделі степеневого обчислювача.

На виході SM_int генеруються вихідні імпульси (біти) степеневої функції y_{int} і запускають обчислювальні процеси в конвеєрній архітектурі зворотного зв'язку. Вихід SM_int поєднаний з елемент затримки DE3 і через нього з групою &4 таким чином, що кожен вихідний біт SM_int ініціює дії:

а) відкриває групу &3 і здійснюється перенос паралельного додаткового двійкового кода числа, що утворюється у SM2, в суматор SM_int, тобто числа ряду першої різниці функції рівнів вузлів апроксимації $(2y-1)^2$;

б) здійснює перенос прямого двійкового кода константи, що зберігається з RG2 у SM2. В результаті у SM2 будут утворюватися числа ряду першої різниці функції $(2y-1)^2$.

Якщо на вхід пристрою подаються серії імпульсного потоку і треба здійснювати обробку кожної серії і обримувати результат функції, то після обчилювальної обробки кожної серії, останім імпульсом серії - імпульсом скидання «initial state» необхідно встановлювати SM1 у число ініціалізації, тобто 1^2. В результаті на виході SM_int буде формуватися результат обчислення степеневої функції з дробовим показником степеня, а отже, вилучення квадратного кореня із суми квадратів серій імпульсів.

На лругому етапі на вхід дільника з виходу SM_int надходять біти потоку функції x: $y_{int} := f(x)$. З приходом першого біту на вхід SM_RES, його вміст збільшується на число 2 у прямому двійковому коді, тобто рівний $2^i - p + 2$. Кожним бітом вхідної послідовності x, в SM_RES потрібно вносити число 2, у відповідності до математичної моделі обчислювача лінійних функцій, в якій в

лівої частині нерівностей прирости функції аргументу помножуються на число 2.

Імпульси переповнення суматора SM_RES дільника є імпульсами (бітами) вихідного потоку *y* функціонального пристрою перетворення та з'являються при виконанні нерівностей математичної моделі лінійного обчислювача (2.10).

Архітектура спроектованого пристрою на основі степеневого обчислювача дозволила розширити частотний діапазон у порівнянні з прототипом пристрою на основі дробово-раціонального обчислювача, в якому на виході SM_int генеруються серії вихідного бітового потоку, що є обмеженням по частоті вхідного імпульсного потоку пристрою уцілому.

У розробленому пристрою після кожного переповнення суматора SM_int у ньому залишається число у прямому двійковому коді, яке менше ніж число, яке внесено у SM_int по зворотному зв'язку зі сторони SM2 у додатковому коді, тому при переповненні суматора SM_int не будуть генеруватися серії імпульсів, а генеруються одиничні імпульси. В результаті з надходженням першої серії імпульсів на вхід пристрою генеруються відповідні імпульси.

Після введення першої серії імпульсів і наступних серій суматор SM1 встановлюється у число ініціалізації імпульсом «initial state», а у SM_int залишається в додатковому коді число і при надходженні кожної наступної серії імпульсів на вхід пристрою частота імпульсів (бітів), що з'являються на виході SM_int буде зменшуватися. Це означає, що степеневий обчислювач працює в режимі вибірки імпульсів з вхідного імпульсного потоку.

3.2 Автоматна модель пристрою

Функціональний потоковий пристрій перетворення представлено мікропрограмним кінцевим автоматом моделі Мура, що містить складові - керуючий і операційний автомати. Операційний автомат має обчислювальні стани, в яких виконуються мікрооперації - перетворення даних (конвеєрні

обчислення), що ініціюються сигналами керування автомату. При цьому керуючий автомат визначає послідовність керуючих сигналів та виконання мікрооперацій у певній послідовності на основі граф-схеми алгоритму (ГСА). В роботі створено ГСА операційного автомату, що наведено на рис.3.3.

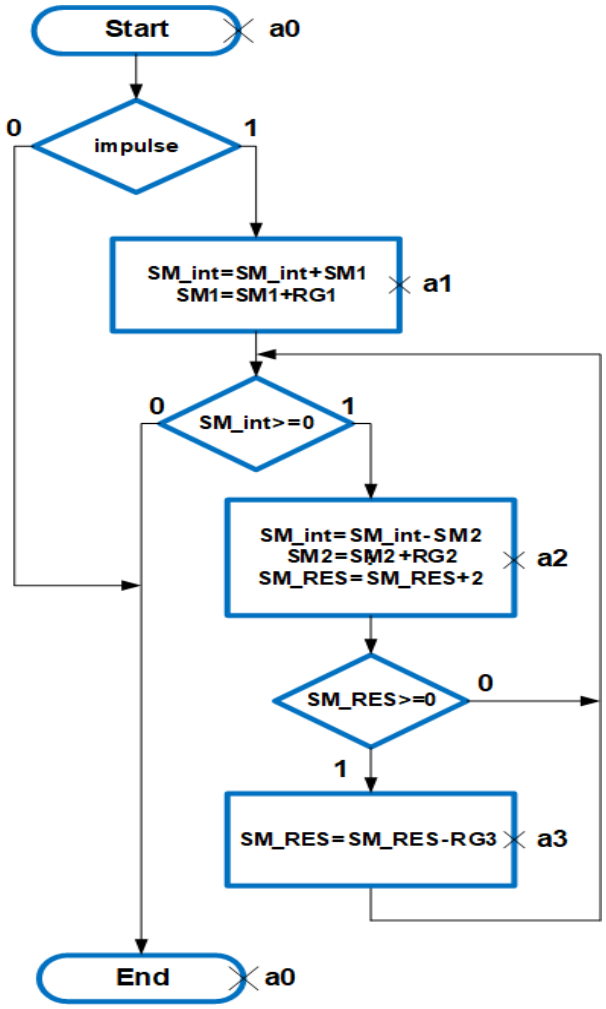

Рисунок 3.3 – ГСА операційного автомату функціонального пристрою

Автоматна модель пристрою на основі автомату Мура є зручною і найбільш придатною для опису, так як обчислювальні операції здійснюються у станах керуючого автомату, а не на переходах.

Відповідно до ГСА біт-потоковий обчислювач працює за наступним алгоритмом.

При скиданні пристрою за сигналом «reset» регістри встановлюються у

значення ініціалізації відповідних компонентів. ГСА містить умовну вершину «impulse». При надходженні на вхід пристрою чергового «impulse» в арифметичному блоку обчислювача виконуються конвеєрні обчислення: значення регістра суматора SM_int збільшується на значення регістра суматора SM1, значення регістра суматора SM1 збільшується на значення регістра RG1.

Якщо значення регістра $SM_int \geq 0$, то на його виході генерується вихідний біт, що є результатом обчислення проміжної степеневої функції $y_{int} := f(x)$, від значення регістра $SM_int \geq 0$ віднімається значення регістра суматора SM2, значення регістра суматора SM2 збільшується на значення регістра RG2. Також, до значення регістру суматора SM_RES додається константа 2.

Якщо вміст регістра суматора SM_RES - невід'ємне число, то на його виході буде сгенеровано біт *y*, що є вихідним бітом дробово-ірраціонального обчислювача, за допомогою якого з регістра суматора SM_RES віднімається значення регістра RG3.

В результаті розмітки ГСА розроблено граф переходів керуючого автомату пристрою, що зображено на рис.3.4.

Граф забезпечує чіткість та наочність керування обчислювальними станами пристрою.та має чотири стани: a0, a1, a2 – стани керування арифметичним блоком степеневого обчислювача. Стани a2 і a3 описують керування обчислювальними станами дільника чисел, в якому здійснюється ділення результату проміжної степеневої функції на константу p.

У стані a1 виконуються конвеєрні обчислення в компонентах прямого зв'язку архітектури степеневого обчислювача. Якщо значення регістра суматора $SM_int \geq 0$ - число додатне, то на виході суматора з'являється вихідний біт степеневої функції і автомат переходить у стан a2. Якщо значення регістра суматора $SM_int < 0$ - від'ємне, автомат переходить у стан a0. У стані a2 виконуються конвеєрні обчислення у компонентах зворотного зв'язку

архітектури степеневого обчислювача. Також, у стані a2 в SM_RES вноситься константа та підсумовується з вмістом SM_RES. Автомат переходить у стан a3, якщо значення регістра суматора $SM_RES \geq 0$. У стані a3 від значення регістра суматора SM_RES віднімається значення регістра RG3, а також на виході суматора SM_RES буде сформовано вихідний біт пристрою і автомат переходить у стан a0.

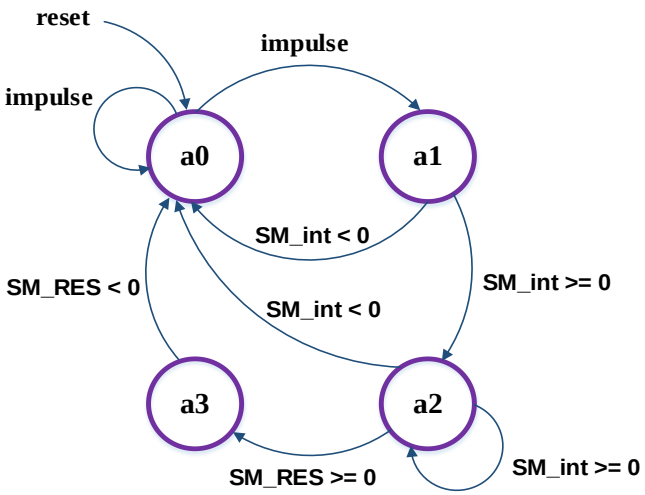

Рисунок 3.4 – Граф переходів керуючого автомату пристрою

3.3 Вібір технологічної платформи для імплементації проекту

В роботі було здійснено вибір платформи ПЛІС для імплементації розробленого функціонального пристрою. Компанія Xilinx – це одна з основних світових компанії, що здійснює виробництво ПЛІС, які мають досить широкий вибір кристалів.

ПЛІС - цифрові надвеликі інтегральні схеми (НВІС) з певною програмованою внутрішньою структурою і які мають змогу паралельно здійснювати значну кількість обчислювальних операцій. ПЛІС FPGA рекомендовані к використанню для імплементації архітектур, що здійснюють

конвеєрні обчислення. Крім того, вони мають можливість здійснювати реконфігурацію обчислювальних архітектур.

ПЛІС виконані за різною архітектурою та мають певні переваги, зокрема, високу ступінь інтеграції, що дозволяє розмішувати пристрій на одному кристалі; високу швидкодію пристроїв та енергоефективність. Крім того, дають можливість скоротити час проектування пристроїв, мають зручні інструменти систем автоматизації проектування (САПР).

В процесі проектування було здійснено синтез та імплементацію моделі пристрою у ПЛІС FPGA. Розроблення апаратної реалізації моделі функціонального пристрою виконувалось на основі платформи ПЛІС Xilinx Spartan-3E серії XC3S500E, що приведено на рис.3.5.

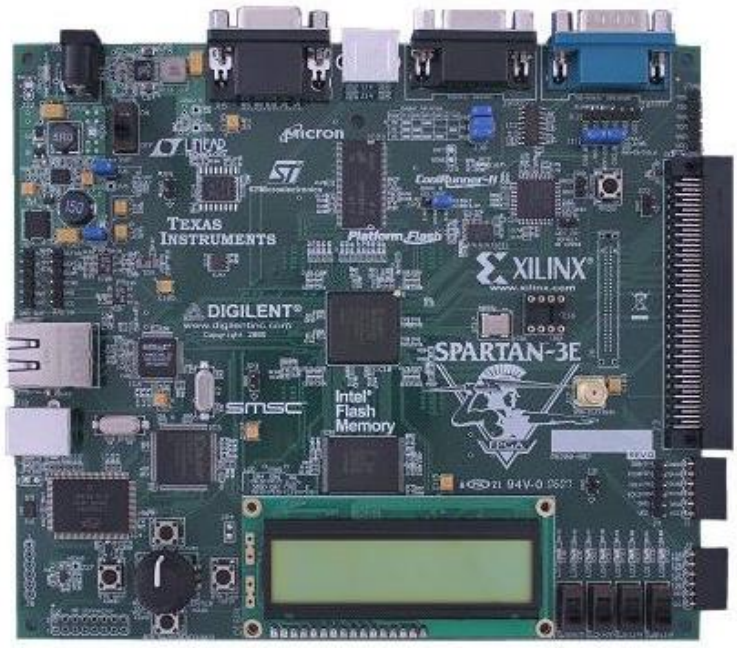

Рисунок 3.5 – Платформа ПЛІС FPGA Xilinx Spartan 3E серії XC3S500E

Програмними засобами, що представляють систему наскрізного проектування, є засоби WebPACK ISE. Система наскрізного проектування інтегрує різні етапи розроблення, зокрема, аналіз, проектування пристрою, розроблення та тестування моделей, забезпечуючи безперервний та повний

процес проектування цифрових пристроїв на платформі ПЛІС.

При повному циклі проектування здійснюються певні етапи: складається специфікація пристрою - вихідні дані проекту з описом певних функцій, сигналів, а далі розроблюється модель пристрою на мові опису апаратури, виконується синтез цифрового модуля, розміщення, трасування, програмування кристала ПЛІС і наступним етапом є моделювання, оцінка апаратних затрат та частотних характеристик пристрою.

Значним фактором є значне скорочення часу на розроблення цифрового пристрою, або системи, а також забезпечення підвищення рівня ефективності результатів проектування, завдяки використанню ПЗ ISE при створенні цифрового пристрою.

ПЛІС являє собою програмований вентильний масив та дозволяє проектувальнику створювати спеціалізовані пристрої певного завдання і призначення. Програмування ПЛІС дозволяє реалізовувати різні функції за допомогою логічних елементів та зв'язків.

Рис. 3.4 містить технічні характеристики платформи ПЛІС FPGA Spartan 3E серії XC3S500E, що була використана для імплементації розробленого функціонального пристрою.

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 3.4 – Характеристики кристала ПЛІС FPGA Spartan 3E серії XC3S500E

Матриця логічних комірок (logic cells) складає основу ПЛІС FPGA. Базовим логічним елементом FPGA служить логічний блок (CLB). Даний блок може бути реконфігурований і створений зі slice-секцій. Секції являють собою

поєднання логічних комірок. CLB-блок містить у собі чотири логічних комірки. Логічні комірки оточені блоками введення/виведення (input/output blocks). Ці блоки підключаються до певних зовнішніх виводів корпусу ПЛІС.

Логічні комірка включає чотирьох входовий функціональний генератор, логічні елементи прискореного перенесення і запам'ятовуючий елемент. Вихід кожного функціонального генератора у кожній логічній комірці приєднанно до вихідної лінії CLB-блоку і до D-входу тригера. Кожен CLB-блок в серії Spartan-3E містить чотири логічних комірки, організовані у вигляді двох однакових секторів (Slice). Функціональні генератори реалізовані як 4-х входові функціональні таблиці (LUT). Кожний LUT-елемент може бути використаний як синхронна пам'ять типу RAM розмірністю 16x1 біт.

ПЛІС широко використовуються у багатьох вбудованих системах (ВС) реального часу, що дає розробнику можливість створювати досить складні пристрої, здійснювати їх реконфігурацію, а також дає можливість здійснювати удосконалення моделей проектування шляхом їх перепрограмування.

4 АПАРАТНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛЬНОГО ПРИСТРОЮ

У розділі приведено результати апаратної реалізації функціонального пристрою перетворення імпульсних потоків, приведено результати теоретичних обчислено функції та обчислювального процесу у компонентах розробленої архітектури пристрою. Розроблено автоматну HDL-модель пристрою, здійснено верифікацію моделі пристрою та її імплементацію у ПЛІС Xilinx Spartan.

4.1 Теоретичні розрахунки для експерименту

В роботі було розроблено апаратну реалізацію пристрою, було виконано математичні розрахунки для порівняння теоретичних результатів з практичною реалізацією.

Функціональний пристрій перетворення імпульсних потоків відтворює дробово-ірраціональну функцію:

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

де x_i – серії бітового (імпульсного) вхідного потоку;
p – ціле додатнє число.

На вхід обчислювача подається бітова послідовність x, що представляє собою серії бітів (імпульсів). На виході пристрою формується вихідний бітовий потік *y,* який відтворює функцію (4.1.) з похибкою 0,5 одиниці молодшого біту аргументу x.

У таблиці 4.1 наведено вихідні дані для експерименту: серії x_i, що задані кількістю бітів у пачці x x_1, x_2, x_3, x_4, коефіцієнта p, кількість серій

імпульсів n. Вихідні дані для проведення експерименту та теоретичних розрахунків приведено на рис. 4.1.

x_1	x_2	x_3	x_4	n	p
7	4	5	6	4	2

Рисунок 4.1 – Вихідні дані для проведення експериментальної апаратної реалізаії

З урахуванням даних функція, що реалізується в пристрої має вигляд:

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

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

1. При виконанні підготовчих тероретичних розрахунків необхідно порахувати значення обчислення дробово-ірраціональної функції для 4-х серій імпульсного потоку, що наведено на рис. 4.1. Результати обчислення заданої функції приведено у додатку Б, табл. Б1.

2. На наступному етапі необхідно знайти значення арифметичних рядів та їх різниць для функцій лівої та правої частин основної нерівності, що реалізується у степеневому обчислювачі на першому етапі обчислень, так як в архітектурі пристрою використано конвеєрну архітектуру степеневого обчислювача, що базується на алгоритмі конвеєрних обчислень.

Нерівність, яка реалізується в обчислювачі степеневої функції, на першому етапі обчислень має вигляд: $2^2 x_y^2 \geq (2y_k - 1)^2$. На основі алгоритму

конвеєрних обчислень обчислено арифметичні ряди та їх різниці функцій лівої та правої частин нерівності. Максимальна довжина серії для розрахунків складає 7 імпульсів, тому для розрахунку значень арифметичних рядів імпульсні потоки дорівнюють: x = y = 7 імпульсів.

Для функції x_y^2 арифметичний ряд другого порядку та ряди першої та другої різниць мають вигляд:

f_x: 1, 4, 9, 16, 25, 36, 49.

: 3, 5, 7, 9, 11, 13

: 2, 2, 2, 2, 2.

Для функції $(2y_k - 1)^2$ арифметичний ряд другого порядку та ряди першої та другої різниць пораховані як:

f_y: 1, 9, 25, 49, 81, 121, 169.

: 8, 16, 24, 32, 40, 48,

: 8, 8, 8, 8, 8.

Наступним кроком необхідно визначити значення ініціалізації компонентів архітектури функціонального пристрою. У архітектурі компоненти ініціалізуються першими значеннями арифметичних рядів та їх різниць (двійкові коди чиесл).

Числа ініціалізації компонентів архітектури наступні:

степеневий обчислювач: SM_int: $2^i - 1$; SM1: 1; SM2: 8; RG1: 2; RG2: 8;

дільник чисел: SM_RES: $2^i - 2$; RG3: $2^i - 4$.

Для проведення експерименту необхідно записати обчислення у компонентах архітектури спроектованого пристрою. Обчислення у компонентах розробленої архітектури функціонального пристрою приведено у додатку Б, таблиці Б.2.. При виконанні обчислень у компонентах враховано перенос чисел із SUM1 в SUM_int зі зсувом на 2 розряди вправо у бік старших розрядів, що означає множення аргументу x на число 2^2 у нерівностях математичної моделі пристрою у лівій частині нерівностей, що відповідає функції аргументу.

При порівнянні розрахунків значень обчислення дробово-ірраціональної функції (таб.Б.1) і обчислень у компонентах пристрою (табл.Б.2) значення дробово-ірраціональної функції для 4-х серій імпульсного потоку *x* та поява імпульсів вихідного потоку *y* на виході SM_RES, який є результатом відтворення апрксимуючої дробово-ірраціональної функції, відповідають обчисленням.

4.5 Блок-схема функціонального пристрою

При розробленні апаратної реалізації функціонального пристрою створено блок-схему функціонального пристрою (рис.4.1).

У складі пристрою є два блоки.

Перший блок - детектор імпульсу. Даний блок сприймає імпульсний потік x, що подається на вхід пристрою і формує на виході імпульси «impulse», амплітуда яких дорівнює логічній «1». Цей імпульс буде поданий в арифметичний блок функціонального пристрою перетворення імпульсного потоку.

Другий блок - блок функціонального пристрою перетворення імпульсних потоків. У даному блоку є арифметичний блок «Speccalc», він виконує двох етапне обчислення: піднесення аргументу x (серій імпульсного потоку) у дробовий степінь – вилучення квадратного корення з суми квадратів серій імпульсного потоку аргументу *x*, а на другому етапі ділення отриманого результату на константу p. При цьому буде отримано результат обчислення дробово-ірраціональної функції із заданою похибкою обчислення 0,5 одиниці молодшого біта аргументу. Після обчислень арифметичний блок сгенерує сигнал «ready», що буде означати готовність арифметичного блоку прийняти наступний біт (імпульс) на обробку з виходу детектора імпульсів.

Рисунок 4.1 – Блок-схема функціонального пристрою

4.4 Опис HDL-проекту

При створенні HDL-проекту пристрою в якості середовища моделювання та верифікації проекту на платформі ПЛІС FPGA було використано пакет Active-HDL, за допомогою якого може бути автоматизовано процес введення чтвореного проекту у САПР та здійснено аналіз проектного рішення. Автоматну модель пристрою написано мовою опису апаратури (МОА) VHDL. При описі моделі використано автоматну моделі опису, автоматне програмування. Мову VHDL обрано завдяки високій адаптації засобів мови для здійснення проектування цифрових пристроїв. Іншою перевагою є наявність ефективного середовища моделювання синтезу. HDL-модель пристрою є складним проектом, що має ієрархічну структуру з деталізацією опису пристрю на рівнях нижчої ієрархії.

HDL-модель пристрою файлової структури проекту включає файли: speccalc_top.vhd, speccalc.vhd, speccalc_oa.vhd, speccalc_ua.vhd, speccalc_imp.vhd, speccalc_tb. v, config_packages.vhd, traceability_unit.vhd.

Файли HDL-опису проекту об'єднані за допомогою top-файла «Speccalc.top.vhdl», він є верхнім рівнем ієрархії HDL-проекту. HDL-опис «Speccalc.top.vhdl» містить компоненти другого рівня ієрархії, що об'єднані за допомогою оператору port map, які є об'єднаними файлами нижнього другого рівня ієрархії (лістинг 4.1). Цей файл є реалізацією специфікації усього

пристрою. Файл «Speccalc.top.vhdl» містить опис компонентів другогого рівня «Speccalc.imp.vhdl», який описує детектор імпульсів, «Speccalc.vhdl», що є top-файлом функціонального пристрою, представленого цифровим автоматом.

Фрагмент HDL-опису з'єднання блоків верхнього рівня ієрархії проекту описує їх порти: блок-схеми пристрою – детектора імпульсів speccalc_imp та блок функціонального пристрою перетворення серій імпульсного потоку speccalc та опис traceability_unit, що описує порти сигналів формування серій імпульсного потоку.

Лістинг 4.5 – HDL-опис з'єднання блоків верхнього рівня у файлі "Speccalc.top.vhdl"

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

```
        next_bundle => next_bundle);
```

«Speccalc.top.vhdl» містить entity, в якому описано порти вхідних та вихідних сигналів функціонального пристрою, заявлені складові компоненти рівня другого нижнього рівня ієрархії, що відповідають блок-схемі пристрою, а отже, компонент детектора імпульсів «Speccalc.imp.vhdl» та компонент функціонального пристрою «Speccalc.vhdl» (лістинг 4.2). Сигнал «next bundle» - сигнал початку формування чергової серії імпульсів, та сигнал «get next bundle» - сигнал завершення формування чергової серії імпульсів.

Лістинг 4.2 – Фрагмент HDL-коду, що містить опис компонентів другогого рівня «Speccalc.imp.vhdl» і «Speccalc.vhdl»

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

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

У лістингу 4.3 наведено фрагмент HDL-опису компонента третього рівня

ієрархії проекту: керуючого автомату функціонального пристрою перетворення. Фрагмент дає опис переходів автомата у відповідні стани a1, a2, a3, a4 на основі графа переходів керуючого автомату пристрою, при цьому використовується оператор case.

Лістинг 4.3 – Фрагмент HDL-опису керуючого автомату функціонального пристрою

```
process(state, x_i, sum1_above_zero_i, sum2_above_zero_i)
     begin
          case (state) is
               when a_0 =>
               if x_i = '1' then
                    next_state <= a_1;
               else
                    next_state <= a_0;
               end if;
               when a_1 =>
               if sum1_above_zero_i = '1' then
                    next_state <= a_2;
               else
                    next_state <= a_0;
               end if;
               when a_2 =>
               if sum2_above_zero_i = '1' then
                    next_state <= a_3;
               else
                    next_state <= a_0;
               end if;
               when a_3 =>
               next_state <= a_0;
               when others =>
               next_state <= a_0;
          end case;
end process;
```

HDL-опис операційного автомату функціонального пристрою створюється для заданої дробово-ірраціональної функції. У лістингу (4.4) приведено фрагмент коду, в якому описано операційний автомат блоку «Speccalc.oa». В даному описі перераховано компоненти з їх ініціалізацією певними значаннями чисел, конвеєрні обчислення в компонентах

функціонального пристрою HDL-модель операційного автомата функціонального пристрою наведено у додатку В.

Лістинг 4.4 – Фрагмент HDL-опису операційного автомата Speccalc.oa функціонального пристрою

```
process (clock_i, reset_i)
     begin
     if (reset_i = '1') then
          counter <= CONV_STD_LOGIC_VECTOR(1, width);
          sum1 <= CONV_STD_LOGIC_VECTOR(-1, width);
          sum2 <= CONV_STD_LOGIC_VECTOR(8, width);
          sum3 <= CONV_STD_LOGIC_VECTOR(-p, width);
          get_next_bundle <= '0';
     else
          if (falling_edge(clock_i)) then
          if (x_imp_i = '1') then
          sum1 <= sum1 + CONV_INTEGER(counter(width-3 downto 0)&
    "00");
               if (next_bundle = '1') then
                    counter <= CONV_STD_LOGIC_VECTOR(1, width);
                         get_next_bundle <= '1';
                    else
                         counter <= counter + step_cnt;
                         get_next_bundle <= '0';
                    end if;
               elsif (sum1_ovf_i = '1') then
                    sum1 <= sum1 - sum2;
                    sum2 <= sum2 + step_sum;
                    sum3 <= sum3 + step_cnt;

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

У додатку В наведено HDL-код програми-обгортки Wrapper.VHDL, який необхідний для здійснення імплементації моделі пристрою та інтеграції із індикацією на платі.

Лістинг 4.5 містить HDL-опис формування серій імпульсів потоку x файлу «config_package.vhdl», в якому можна задавати кількість імпульсів у

кожній з 4-х серій імпульсів потоку x, і який відповідає за управління серіями імпульсів вхідного потоку.

Лістинг 4.5 – Фрагмент HDL-опису формування серій імпульсів потоку x файлу «config_package.vhdl»

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

4.2 Верифікація та імплементація функціонального пристрою

Верифікація поведінкової моделі функціонального пристрою, що реалізує ступінчасту дробово-ірраціональну функцію, виконувалась з використанням САПР Active-HDL. Для верифікації поведінкової моделі пристрою було розроблено модуль test bench, що містить примірник тестового модуля та здійснює подачу тестових сигналів, фрагмент HDL-опису test bench приведено в додатку В.

На часовій діаграмі (рис.4.2) проілюстровано результати моделювання поведінкової моделі функціонального пристрою перетворення імпульсних потоків, що відтворює дробово-ірраціональну функцію, аргумент якої представлений серіями імпульсного потоку x.

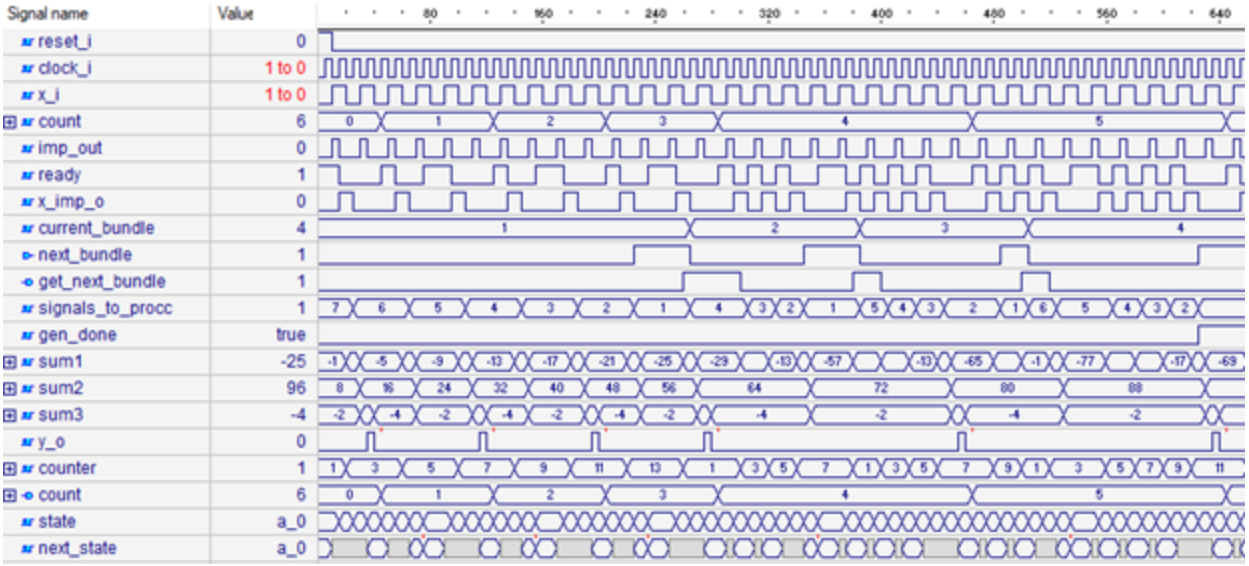

Рисунок 4.2 – Поведінкова модель функціонального пристрою перетворення імпульсних потоків

На діаграмі показано, що на вхід пристрою було подано 4 серії імпульсів по 7, 4, 5, 6 імпульсів у кожній серії відповідно. На виході пристрою було сформовано 6 імульсів потоку y. На діаграмі також показано сигнали «next bundle» та «get next bundle», за допомогою яких формуються серії імпульсного потоку. Імпульсний сигнал «next bundle» - сигнал початку формування чергової серії імпульсів, та сигнал «get next bundle» - сигнал завершення формування чергової серії імпульсів. На діаграмі показано значення регістрів компонентів пристрою, які відповідають розрахунковим значенням у компонентах архітектури пристрою, шо наведено у додатку Б табл. Б.2.

При розробленні апаратної реалізації було здійснено імплементацію моделі пристрою у платформу ПЛІС Xilinx Spartan 3E серії xc3s500e. Після імплементації було проведено моделювання, максимальна частота, на якій коректно працює пристрій складає 135 MHz. В результаті було отримано Synthesis report, в якому зазначено максимальну частоту роботи пристрою, мінімальний період сигналу синхронізації та використання ресурсів кристалу. Звіт приведено нижче.

```
     Target Device : xc3s500e
     Target Package : cp132
     Target Speed  : -5
     Minimum period: 7.501ns (Maximum Frequency: 135.146MHz)
Number of External IOBs                          11 out of 232     4%
Number of External Input IOBs                     3
Number of LOCed External Input IBUFs          3 out of 3     100%
Number of External Output IOBs                    8
Number of LOCed External Output IOBs          8 out of 8     100%
Number of BUFGMUXs                            1 out of 24      4%
Number of Slices                              46 out of 4656    1%
```

При синтезі функціонального пристрою у платформу Xilinx Spartan 3E серії xc3s500e було задіяно 4% логічних елементів, пристрій є економічним з точки зору використання кристалу. В результаті синтезу було отримано RTL-схему вентильного рівня, що містить детектор імпульсів та потоковий функціональний пристрій, який є композицією керуючого та операційного автоматів (рис.4.3).

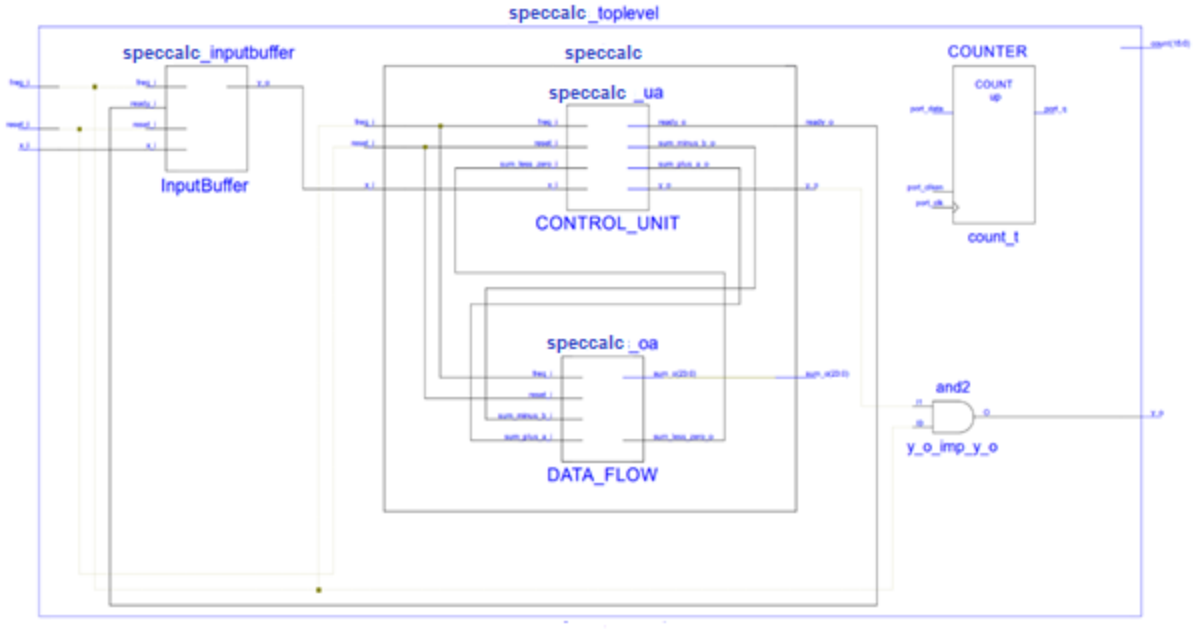

Рисунок 4.3 – RTL-схема вентильного рівня функціонального пристрою перетворення імпульсних потоків

ВИСНОВКИ

В ході виконання кваліфікаційної роботи розроблено функціональний пристрій перетворення імпульсних потоків на платформі ПЛІС з використанням САПР цифрових пристроїв.

У роботі розглянуто СУ реального часу та особливості функціонального перетворення імпульсних потоків у пристроях потокової обробки сигналів. Розглянуто метод апроксимації на основі обернених функцій та математичну модель спроектованого пристрою, що відтворює дробово-ірраціональну функцію та є декомпозицією математичних моделей степеневого обчислювача та обчислювача лінійних функцій.

Приведено загальну архітектуру обчислювача дробово-ірраціональних функцій, на підставі якої було розроблено архітектуру функціонального пристрою перетворення імпульсних потоків. Архітектура спроектованого пристрою є синтезом двох архітектур – архітектури біт-потокового конвеєрного степеневого обчислювача та дільника чисел, побудованого з використанням архітектури обчислювача лінійної функції. Основний компонент архітектури - суматор зі зворотним зв’язком.

Розроблено автоматну модель пристрою на основі кінцевого автомата Мура: створено граф переходів керуючого автомата та граф-схему алгоритма операційного автомата пристрою, що містить конвеєрні обчислення . в обчислювальних станах. Приведено результати апаратної реалізації моделі пристрою, здійснено необхідні розрахунки. Створено HDL-модель пристрою для введення в САПР з використанням HDL-шаблонів коду, що містять автоматний опис. Здійснено верифікацію проекту, отримано поведінкову модель пристрою, результати моделювання якої співпадають з теоретичними розрахунками. Модель пристрою було синтезовано засобами САПР Xilinx. Для імплементації використано платформу FPGA Spartan 3E серії XC3S500E.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Zhou F., Chai Y., “Near-sensor and in-sensor computing,” Nature Electronics, (2020), Vol. 3 (11). pp. 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, P. 1–6.

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

4. Bureneva, O; Kupriyanov, M.; Safyannikov, N. Bit Streaming Processing Algorithms for Intelligent Hardware Converters. / O, Bureneva; M, Kupriyanov; N. Safyannikov // Procedia Computer Science. 2021, Volume 186, P. 194-201. DOI: 10.1016/j.procs.2021.04.138.

5. Shkil A.S., Larchenko L.V., Larchenko B.D. Bit-Stream Power Function Online Computer. / A.S.,Shkil, L.V. Larchenko, B.D. Larchenko // 2020 East-West Design & Test Symposium (EWDTS), Sep 2020, Varna, Bulgaria, P. 1-6. DOI: 10.1109/EWDTS50664.2020. 9224764.

6. Bureneva O., Mironov S., Safyannikov N., Sukhinets Zh. "Functional Converter for Intelligent Sensor and Its Layout Design. Engineering Proceedings", 2023, 33(1):50. https://doi.org/10.3390/engproc2023033050.

7. Kumar P. A., “FPGA implementation of the trigonometric functions using the CORDIC algorithm”, Proceedings of the 5th International conference on Advanced computing & communication systems (ICACCS), 2019, pp. 894-900, DOI: 10.1109/ICACCS.2019.8728315.

8. Korniienko V., Larchenko L., Parkhomenko A. et al. Design Models of Bit-Stream Online-Computers of Elementary Mathematical Functions]. / 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), DOI: 10.1109/IDAACS58523.2023.10348885.

9. Шкіль О.С., Ларченко Б.Д., Ларченко Л.В., "Декомпозиція математичної моделі апаратного біт-потокового обчислювача ірраціональних функцій", Радіоелектроніка та інформатика, 2019, №4, С. 46-52.

10 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, vol. 1(68), 2024, P. 48-61, https://doi.org/10.15588/1607-3274-2024-1-6.