

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## АТЕСТАЦІЙНА РОБОТА

### Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Моделі та методи проектування біт-потокowego  
обчислювача степеневих функцій на базі FPGA

(тема)

Виконав:

студент II курсу, групи СКСм-19-1  
Шапа Л.С.  
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва спеціальності)

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

Освітня програма Спеціалізовані  
комп'ютерні системи  
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри АПОТ Чумаченко С.В.  
(підпис) (прізвище, ініціали)

2020 р.

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва)

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

Освітня програма Спеціалізовані комп'ютерні системи  
(повна назва)

Харківський національний університет радіоелектроніки

ЗАТВЕРДЖУЮ:

Зав.  
кафедри \_\_\_\_\_  
(підпис)

2  
0 р

“ ” - .

## ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Шапа Людмила  
(прізвище)

1. Тема роботи Моделі та методи проектування біт-потокowego Тема роботи малими літерами  
обчислювача степеневих функцій на базі FPGA

затверджена наказом“ \_\_\_\_\_ 30 \_\_\_\_\_ ”

2. Термін подання \_\_\_\_\_ 15

3. Вихідні дані до \_\_\_\_\_

FPGA кристал сімейства Xilinx Spartan-3E серії XC3S500E

САПР Active-HDL

Мова опису апаратури VHDL

4. Перелік питань, що \_\_\_\_\_

1 Аналіз предметної області та постановка завдання дослідження

2 Математична модель біт-потокowego обчислювача степеневих функцій

- 3 Структурний синтез спроектованого обчислювача
- 4 Апаратна реалізація спроектованої моделі обчислювача
- 5 Верифікація, тестування та імплементація обчислювача в платформу ПЛІС
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 19 слайдів

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2020 - 02.09.2020	
2	Аналіз предметної області	03.09.2020 - 15.09.2020	
3	Аналіз джерел з проблемної галузі	16.09.2020 - 01.10.2020	
4	Розробка математичної моделі обчислювача	02.10.2020 - 15.10.2020	
5	Структурний синтез проект. Обчислювача	16.10.2020 - 30.10.2020	
6	Розробка апаратної реалізації обчислювача	01.11.2020 - 14.11.2020	
7	Перевірка правильності роботи обчислювача	15.11.2020 - 18.11.2020	
8	Оформлення пояснювальної записки	19.11.2020 - 29.11.2020	
9	Оформлення графічного матеріалу	30.11.2020 - 07.12.2020	
10	Перевірка виконаного проекту керівником	08.12.2020 - 15.12.2020	

Дата видачі завдання 01 вересня 2020 р.

Студент



(підпис)

Керівник роботи



(підпис)

доц. Ларченко Л.В.

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 77 сторінок, 17 рисунків, 5 таблиць, 2 додатки, 11 джерел за переліком джерел.

БІТОВИЙ ПОТІК ДАНИХ, БІТ-ПОТОКОВИЙ ОБЧИСЛЮВАЧ  
СТЕПЕНЕВИХ ФУНКЦІЙ, СУМАТОР, АПРОКСИМАЦІЯ, АЛГОРИТМ,  
ГСА, VHDL-МОДЕЛЬ, ВЕРИФІКАЦІЯ, ПЛІС

В атестаційній роботі досліджено та розроблено обчислювач степеневих функцій з біт-потоквою формою подання аргументу з урахуванням абсолютної похибки обчислень, в якому операції піднесення до степеню і обчислення кореню суміщені в одному пристрої.

З використанням єдиної методики ступінчастої апроксимації неперервних функції, розроблено математичну модель обчислювача степеневих функцій. Розроблено архітектуру обчислювача з використанням відомих способів побудови конвеєрних структур цифрових обчислювачів для відтворення поліноміальних функцій.

Здійснено апаратну реалізацію обчислювача степеневих функцій на базі цифрового автомату. Розроблено граф-схему алгоритму роботи біт-потоквого обчислювача та закодовано для синтезу автомата Мура. Відповідно до отриманого завдання розроблено апаратну модель мовою VHDL з використанням автоматної моделі опису. Виконана схемна реалізація та верифікація отриманого рішення з використанням автоматизованого тестового середовища test bench. Результати моделювання співпадають з теоретичними розрахунками. Модель реалізована в програмовану логічну інтегральну схему Xilinx Spartan 3E.

## THE ABSTRACT

Master's thesis contains 77 pages, 17 figures, 5 tables, 2 additions, 11 sources according to the list of links.

BIT-STREAM, BIT-STREAM POWER FUNCTION COMPUTER, SUMATOR, APPROXIMATION, ALGORITHM, VHDL-MODEL, VERIFICATION, FPGA

In the attestation work the power function computer with bit-stream form of argument presentation taking into account the error is investigated and developed, allows combining the exponentiation and root extraction operations in one device

Using a single technique of stepwise approximation of continuous functions, a mathematical model of the power function computer is developed. The structure of the computer of the set function is developed, with use of known methods of construction of conveyor structures of digital computers for polynomial functions reproduction.

The graph-scheme of the algorithm of operation of the bit-flow calculator is developed, and coded for synthesis of the Moore machine. According to the received task the hardware model in VHDL language with use of the automatic model of the description is developed.

The hardware implementation of the device for calculation of the set function on the basis of the digital automatic machine is carried out, the circuit implementation and verification of the received decision with use of the automated test environment test bench is executed. The simulation results coincide with the theoretical calculations. The model is implemented in a programmable logic integrated circuit Xilinx Spartan 3E.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	12
1.1 Функціональне перетворення бітових потоків даних.....	12
1.2 Застосування пристроїв обчислення степеневих функцій з біт-потоким кодуванням.....	14
1.3 Особливості проектування з використанням ПЛІС.....	17
1.4 Постановка завдання дослідження.....	20
2 МАТЕМАТИЧНА МОДЕЛЬ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА СТЕПЕНЕВИХ ФУНКЦІЙ.....	21
2.1 Метод ступінчастої апроксимації неперервних висхідних функцій.....	21
2.2 Математичне обґрунтування обчислення степеневих функцій.....	26
2.2.1 Алгоритм обчислення поліноміальних функцій.....	27
2.2.2 Математична модель апаратного біт-потокимого обчислювача степеневих функцій.....	29
3 СТРУКТУРНИЙ СИНТЕЗ ОБЧИСЛЮВАЧА СТЕПЕНЕВИХ ФУНКЦІЙ. 33	
3.1 Принцип побудови біт-потокимих апаратних обчислювачів на основі конвеєрних архітектур.....	33
3.2 Структурно-функціональна модель обчислювача поліноміальних функцій.....	35
3.3 Узагальнена структурно-функціональна модель обчислювача степеневих функцій. 36	
3.4 Структурно-функціональна модель біт-потокимого обчислювача степеневих функцій .....	38
3.5 Вибір елементної бази.....	44
4 АПАРАТНА РЕАЛІЗАЦІЯ ДОСЛІДЖУВАНОЇ МОДЕЛІ ОБЧИСЛЮВАЧА .....	49
4.1 Специфікація досліджуваного обчислювача.....	49
4.2 Результати обчислення заданої функції та обчислювальний процес в компонентах пристрою .....	50
4.3 Граф-схема алгоритму роботи біт-потокимого пристрою.....	52
4.4 Граф переходів управляючого автомату пристрою .....	54
4.5 Структурно-блокова схема досліджуваного обчислювача .....	54
4.6 Опис блоків проекту МОА.....	58
4.7 Опис проекту МОА.....	59
4.8 Верифікація та тестування роботи пристрою .....	63
4.9 Імплементация моделі пристрою .....	65
Висновки.....	70

Перелік джерел посилання.....	72
ДОДАТОК А.....	74
Графічний матеріал атестаційної роботи.....	74
ДОДАТОК Б.....	85
Лістинг коду програм.....	85

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

- CLB (Configurable Logic Block) – програмований логічний блок;
- IOB (Input/output blocks) – блок введення – виведення;
- FPGA (Field Programmable Gate Arrays) – програмована користувачем вентильна матриця;
- SDH (Synchronous Digital Hierarchy) – принцип побудови цифрових систем передачі;
- TCP (Transmission Control Protocol) – протокол керування передачею;
- ВІС – велика інтегральна схема;
- ГСА – граф-схема алгоритму;
- ЕОМ – електронно-обчислювальна машина;
- ІКС – інформаційно-керуюча система;
- КМОН – комплементарна структура метал-оксид-напівпровідник;
- КСАУ – комп’ютерні системи автоматичного управління;
- МОА – мова опису апаратури;
- НВІС – надвелика інтегральна схема;
- ОЗП – оперативний запам’ятовуючий пристрій;
- ПЛІС – програмована логічна інтегральна схема;
- САПР – система автоматизації проектування;
- ЦФПР – цифрові функціональні перетворювачі розгортуючого типу.

## ВСТУП

Одним з прогресивних напрямків розробки сучасних обчислювальних систем є створення нових і вдосконалення існуючих засобів обчислювальної техніки, широко використовуваних при вирішенні найрізноманітніших завдань. Структура таких обчислювальних систем містить множину функціональних перетворювачів і біт-потоківих обчислювачів для реалізації відповідної множини функціональних завдань. Реалізувати окремі функціональні завдання можна або за допомогою спеціалізованих програмних засобів, або допомогою спеціального алгоритмічного забезпечення, відображеного на спеціалізовану апаратуру. В атестаційній роботі прийнято програмно-апаратний напрямок проектування біт-потоківих обчислювачів.

У більшості систем управління та інформаційно-вимірювальних систем біт-потоківих обчислювачі є основним засобом нелінійної обробки аналогової та цифрової інформації, а в ряді випадків, виступають в ролі периферійних процесорів функціональних розширювачів високоефективних обчислювальних систем. Важливою вимогою до таких пристроїв є необхідність виконання обробки в реальному масштабі часу, що накладає жорсткі обмеження на час вирішення завдань.

Розвиток таких галузей, як робототехніка, сенсорика, інтернет речей та інших систем, де існує необхідність отримання і обробки даних від великої кількості різноорієнтованих сенсорів, та розвиток архітектур потоківих процесорів, що включають у себе декілька зовнішніх модулів обробки потоків даних та типове процесорне ядро, зумовлюють збільшення складності завдань по організації обчислень та взаємодії між компонентами пристроїв. Використання традиційних методів обчислень в таких системах є менш ефективним за продуктивністю та надійністю, що зумовлює необхідність використання нових обчислювальних пристроїв.

Одним із напрямків розвитку обчислювальних систем є використання бітових потоків даних для їх послідовної передачі в системах. Пристрої на основі біт-потоків обчислень можуть бути використані в різноманітних системах, як складові зовнішніх апаратних модулів розподілених систем, що працюють з поточними формами інформації.

Доцільним є використання біт-потоків сигналу в системах автоматичного управління і регулювання, які працюють в комплексі з ЕОМ, оскільки перетворення бітового потоку здійснюється простими засобами з малими втратами точності. Також подібні обчислювачі можуть одночасно виконувати прості математичні операції з потоками даних, тим самим спрощуючи програмування роботи при одночасному підвищенні швидкодії всієї системи в цілому.

Метою атестаційної роботи є дослідження та розробка моделей і методів автоматизованого проектування пристрою обчислення степеневих функцій з біт-поточною формою подання аргументу з урахуванням абсолютної похибки відтворення функції, що дозволяє об'єднати операції піднесення до степеню і взяття кореня в одному пристрої, на технологічній платформі ПЛІС.

Пристрій може бути застосований для реалізації різних функціональних залежностей в автоматичних системах моделювання, контролю, діагностики та проектування складних динамічних об'єктів, при побудові аналого-цифрових і цифро-аналогових каналів сучасних обчислювальних, управляючих і інформаційно-вимірювальних систем, вимірювально-обчислювальних комплексів і автоматизації наукових досліджень, в системах штучного інтелекту.

Застосування пристрою також доцільне в системах, де має місце цифрова функціональна розгортка, що передбачає реалізацію поточного методу обчислень в часі, з послідовним обчисленням значень функції для сусідніх значень аргументу, та з паралельним виконанням перетворень над бітовим потоком даних, відповідно до заданої функції.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

В розділі розглянуто функціональне перетворення бітових потоків даних, галузі застосування пристроїв обчислення степеневих функцій з біт-потоківим кодуванням, особливості проектування на технологічній платформі ПЛІС, поставлено мету дослідження, визначені об'єкт, предмет дослідження, а також сформульовані завдання дослідження.

### 1.1 Функціональне перетворення бітових потоків даних

Бітовий потік даних – це суцільна послідовність бітів, що представляє собою потік даних, які передаються безперервно по комунікаційному шляху послідовно по одному біту. Біт-потоківий спосіб передачі інформації можна розглядати, як спосіб передачі послідовності окремих бітів каналом зв'язку від однієї системи до іншої. Такий канал часто є двонаправленим, та майже у всіх випадках канал володіє властивостями надійності, тобто послідовність відправлених бітів повністю співпадає з послідовністю отриманих бітів.

Бітові потоки даних широко використовуються в телекомунікаціях та обчислювальних технологіях: наприклад, технологія зв'язку SDH передає синхронні потоки бітів, а протокол зв'язку TCP транспортує байтовий потік без синхронізації. Термін бітовий потік часто використовується для опису даних конфігурації, що завантажуються в FPGA.

Більшість цифрових систем працюють з позиційним поданням даних, такими як двійкове кодування. Подання інформації у вигляді бітових потоків є гарною альтернативою позиційному представленню інформації, оскільки біт-потоківі сигнали є квазіцифровими. Це сигнали, для яких характерним є предствлення інформації неперервно в часі, але за необхідності вони легко можуть бути перетворені в дискретні форми подання інформації, такі як двійкове кодування. Ця характеристика дозволяє застосовувати біт-потоківі

форми представлення аргументів для побудови біт-потоків пристроїв одночасно з використанням простої логіки обчислень, реалізуючи їх у базисі традиційних цифрових елементів.

Будь - яка фізична величина загалом може бути перетворена в біт-потоківу послідовність. Такі види подання інформації, як частота і біт-потоківий код, дають змогу здійснити цифрове функціональне перетворення розгортуючого типу, в основі якого лежить цифрова функціональна розгортка, тобто послідовне обчислення значень функції, що виконуються для сусідніх значень аргументу. При цьому є можливість враховувати попередню історію виконання обчислень: кожне наступне значення функції обчислюється з урахуванням результату, отриманого з попереднього обчислення. При цьому перше обчислення здійснюється з урахуванням додаткової інформації, такої як налаштування або заданні початкові умови.

В біт-потоківу послідовність можуть бути перетворені дані, що отримуються від різнорідних сенсорів. В такому випадку бітові потоки будуть представляти собою імпульси тої ж сутності, що й сигнали давача, з одиничним значенням амплітуди, параметри яких будуть пов'язані з часом. Такими даними модуть бути отримані з різноманітних сенсорів біологічні, оптичні, пневматичні, електричні, та інші сигнали.

Використання бітових потоків для послідовної передачі даних в пристроях, які складають елементну базу зовнішніх апаратних модулів, що працюють з потоківими формами даних в децентралізованих системах, дозволяє неабияк спростити взаємодію між компонентами пристрою. Враховуючи цю особливість, використання бітових потоків актуальне при розробці систем, де ставиться завдання узгодження сенсорів з цифровими підсистемами збору і обробки інформації. Такими як: сенсорні інтерфейси, інтернет речей, а також системи призначені для реалізації взаємодії між комп'ютером та людиною.

Одним з широко поширених завдань в системах, що потребують здійснення первинної математичної обробки даних, отриманих з давачів, де

окрім виконання алгебраїчних, арифметичних, та інших математичних операцій також необхідні певні функціональні перетворення з потоками даних, є функціональне перетворення бітових потоків. Також функціональне перетворення бітових потоків може використовуватися в інформаційно-вимірювальних системах, системах моделювання, контролю, управління.

Представлення вхідних і вихідних даних апаратних обчислювачів, що входять до складу різноманітних систем, у вигляді бітових потоків, також дає велику перевагу. Реалізація більшості операцій з даними, представленими у вигляді бітових потоків є більш простою, в порівнянні з іншими видами кодувань, що є важливим для ряду систем. Таких як, системи, що відносяться до областей управління, контролю та вимірювань. Це зумовлено тим, що названі області висувають жорсткіші вимоги до надійності роботи систем [1,2].

## 1.2 Застосування пристроїв обчислення степеневих функцій з біт-потоківим кодуванням

Ефективне застосування електронно-обчислювальних машин загального призначення в сучасних керуючих та вимірювальних системах передбачає використання спеціальної апаратури їх спряження з давачами і виконавчими органами об'єкту керування. У багатьох випадках на цю апаратуру покладаються обчислювальні операції, які зручно вкладаються в процес перетворення та обробки інформації.

Основною перевагою таких поширених обчислювальних операцій є здатність здійснювати функціональне перетворення в реальному масштабі часу, по мірі надходження бітового потоку даних на вхід апаратного обчислювача. Цифрове розгортання у часі неперервних часових залежностей здійснюється за допомогою спеціалізованих обчислювачів – цифрових функціональних перетворювачів розгортуючого типу (ЦФПР) [2].

Одними з найпоширеніших датчиків, в яких вихідним інформативним параметром є частота гармонійного сигналу або біт-поточної послідовності, серед датчиків сучасних систем управління та контролю, є частотні датчики. Тому проблеми вимірювання частотно-часових параметрів сигналів, залишаються актуальними, особливо враховуючи постійне зростання тактових частот, вимог до швидкодії і точності систем. З використанням частотних датчиків досягається більш просте перетворення частоти в цифровий код. Подібні задачі виникають при вимірюванні статичного і диференціального тиску в трубопроводах, швидкостей, переміщень, прискорень і тисків в авіаційній техніці.

Пристрої для обчислення степеневих, лінійних, дробово-раціональних функцій знайшли широке використання в реалізації різних функціональних залежностей в автоматичних системах моделювання, контролю, діагностики та проектування складних динамічних об'єктів, при побудові аналого-цифрових і цифро-аналогових каналів сучасних обчислювальних, управляючих і інформаційно-вимірювальних систем, вимірювально-обчислювальних комплексах і автоматизації наукових досліджень, в системах штучного інтелекту, також в системах, де при проведенні математичної обробки первинної вимірювальної інформації, що отримують від вимірювальних сенсорів потрібно виконання різних нелінійних перетворень бітових потоків даних. За допомогою обчислювачів можна будувати розподілені підсистеми ІКС, наближені безпосередньо до джерел даних і об'єктів управління. Отже, є необхідність в розвитку і дослідженні методів і засобів проектування таких пристроїв.

Особливо широке застосування функціональні перетворювачі з біт-поточним кодуванням знайшли в проектуванні цифрової контрольно-вимірювальної апаратури в якості обчислювальних вузлів цифрових вимірювальних приладів різного призначення (частотомірів, амперметрів, вольтметрів, аналізаторів спектрів та ін.), що реалізують функціональні залежності між їх вхідними та вихідними величинами, при синтезі

перетворювачів форм подання інформації. Виконання обчислювальних операцій в таких перетворювачах зводиться зазвичай до вирішення завдання отримання нової послідовності бітів частоти, пов'язаної певної функціональною залежністю з частотами вихідних біт-потоків послідовностей, які отримують за допомогою елементів ланцюгів генераторів частот, що змінюють свої параметри під впливом зовнішніх дій.

У більшості інформаційно-вимірювальних систем та систем управління обчислювачі є основним засобом нелінійної обробки аналогової і цифрової інформації, а в ряді випадків, виступають в ролі периферійних процесорів функціональних розширювачів високоефективних обчислювальних систем.

Обчислювачі застосовуються також при розробці систем, де має місце необхідність узгодження сенсорів з цифровими системами збору і обробки інформації. Таких як системи орієнтовані на наскрізні технології інтелектуальних датчиків, інтернет речей, а також на удосконалення інтерфейсів взаємодії між людиною і комп'ютером [3].

Досліджуваний біт-потоків обчислювач степеневих функцій може бути застосований:

- в системах, орієнтованих на технології інтелектуальних сенсорів в якості функціональних перетворювачів бітових потоків даних, отриманих з вимірювальних сенсорів фізичних величин;

- в сучасних системах управління та контролю в якості спеціальної апаратури їх спряження з датчиками і виконавчими органами об'єкту керування;

- при відтворенні траєкторій рухомих об'єктів в двомірному і тривимірному просторі;

- при проведенні математичної обробки первинної вимірювальної інформації в інформаційно-вимірювальних системах;

- в якості обчислювальних вузлів в системах і приборах при здійсненні непрямих вимірювань;

– при виконанні різних функціональних перетворень частоти біт-потоків послідовностей.

### 1.3 Особливості проектування з використанням ПЛІС

ПЛІС типу FPGA призначені для реалізації складних проектів, і їх ємність досягає десятків мільйонів «еквівалентних вентилів». Така висока функціональність дозволяє об'єднувати на одному кристалі декілька процесорних пристроїв та інтерфейсну логіку.

Перевагами сучасних ПЛІС є:

- малий час і простота проектування;
- низька вартість розробки;
- більш низька вартість в розрахунку на одну мікросхему в порівнянні з рекомендованими ІС;
- більш тривалий час перебування продукту на ринку завдяки можливості перепрограмування;
- можливість створення динамічно реконфігурованих пристроїв.

До недоліків можна віднести більш низьку швидкість роботи ПЛІС в порівнянні з повністю замовними ІС, а також нерентабельність використання в великосерійному виробництві.

Створення цифрових пристроїв на базі FPGA відбувається із застосуванням спеціалізованих систем автоматизованого проектування і розрахунку. Проектування за допомогою таких систем являє собою послідовне використання наданих програмних засобів. В термінології САПР такий процес називається маршрутом проектування. Проектування розділяють на наступні етапи:

- вибір сімейства та типу FPGA;
- створення проекту за допомогою відповідної системи проектування;
- функціональне моделювання;
- синтез спроектованого пристрою;

- етап трасування проекту;
- етап часового моделювання;
- етап програмування FPGA;
- імплементація проекту в кристал.

Перед створенням нового проекту слід визначитися з вибором методу опису біт-потокowego пристрою. Під час вибору сімейства і типу ПЛІС для реалізації проекту необхідно не тільки оцінювати його складність із урахуванням вимог до швидкодії, споживаної потужності, умов експлуатації, але і врахувати додаткові фактори, такі як: вартість, можливість перепрограмування в системі. Обране сімейство або тип кристала при необхідності досить легко можна змінити в процесі проектування. На цьому ж етапі можна встановити тимчасові і топологічні обмеження, які повинні враховуватися при синтезі, розміщенні і трасуванні проекту в кристалі.

Після створення схем у графічному редакторі або методом текстового опису, відбувається етап функціонального моделювання.

Етап функціонального моделювання дозволяє виконати попередню перевірку проекту. Його мета – виявлення і усунення можливих логічних помилок, шляхом перевірки відповідності вихідних сигналів алгоритмам роботи проектуваним пристроям. На цьому етапі фактично не використовуються тимчасові характеристики та особливості архітектури кристала, на базі якого передбачається реалізація проектуючої системи.

Синтез - це процес трансформації вихідного HDL-опису спроектованого пристрою в список ланцюжків, що виконаний на низькому логічному рівні. Елементи низькорівневого опису, сформованого в процесі синтезу, мають відповідати архітектурі сімейства ПЛІС, що обраний для реалізації проекту. Синтезований список ланцюжків повинен бути максимально адаптованим до ресурсу, що використовується кристалом, яке забезпечує його найбільш ефективне відображення засобів розміщення та трасування на фізичному рівні.

Етап трасування і розміщення проекту складається з двох частин – трансляція і розподіл ресурсів кристалів для реалізації біт-потокowego пристрою. Під час виконання першої частини об'єднується уся необхідна інформація про списки з'єднань і обмеження, що міститься у UCF файлах, за рахунок чого формується логічний опис проекту низького рівня в термінах Xilinx. Під час виконання другої частини, опис проекту розбивається на блоки згідно з ресурсами обраного типу ПЛІС. Також на цьому етапі відбувається мінімізація використовуваних ресурсів кристалу. В процесі виконання цього етапу проектування також визначаються реальні значення затримок поширення сигналів, які необхідні для повного часового моделювання пристрою.

Результатом етапу трасування і розміщення проекту є створення файлу, що містить у собі опис використання фізичних ресурсів кристалів для реалізації функцій біт-потокowego пристрою.

Результатом етапу часового проектування є отримання часових діаграм роботи проектованого пристрою.

Завершенням процесу розробки цифрового пристрою є завантаження конфігураційних даних в кристал за допомогою відповідних програм і завантажувального кабелю.

Не всі з етапів проектування ПЛІС є обов'язковими. У процесі розробки можуть не проводитися моделювання апаратної частини системи. Тому можна виключити етапи підготовки специфікацій моделювання, генерації моделей, функціонального та тимчасового моделювання. При цьому слід враховувати, що моделювання апаратної частини системи дозволяє підвищити ефективність процесу проектування за рахунок більш раннього виявлення можливих помилок [4].

#### 1.4 Постановка завдання дослідження

Метою атестаційної роботи є дослідження та розробка моделей і методів автоматизованого проектування біт-потокowego обчислювача степеневих функцій на основі ПЛІС.

Об'єктом дослідження є спеціалізовані обчислювачі степеневих функцій перетворення бітових потоків даних з зовнішніх сенсорів фізичних величин.

Предмет дослідження – математична та структурна моделі біт-потокowego обчислювача степеневих функцій, структурні моделі обчислювачів на основі конвеєрних архітектур, апаратні моделі обчислювачів на основі цифрових автоматів.

Поставлена мета визначила наступні завдання дослідження:

- аналіз особливостей функціонального перетворення бітових потоків в апаратних обчислювачах математичних функцій;
- аналіз способу обчислення степеневих функцій, аргумент яких представлений бітовим потоком на основі методу ступінчастої апроксимації відтворення неперервних висхідних функцій;
- розробка математичної моделі біт-потокowego обчислювача степеневих функцій;
- аналіз способу побудови конвеєрних архітектур апаратних обчислювачів поліноміальних функцій з біт-потоковой формою даних;
- розробка структурно-функціональної моделі обчислювача заданої функції;
- розробка апаратної реалізації пристрою на основі кінцевого автомата Мура;
- розробка HDL-моделі обчислювача;
- верифікація, тестування та імплементація отриманої апаратної моделі обчислювача в платформу ПЛІС.

## 2 МАТЕМАТИЧНА МОДЕЛЬ БІТ-ПОТОКОВОГО ОБЧИСЛЮВАЧА СТЕПЕНЕВИХ ФУНКЦІЙ

Розділ присвячено розробці математичної моделі обчислювача степеневих функцій. Для отримання математичної моделі використано метод ступінчастої апроксимації неперервних висхідних функцій обраного класу. В розділі отримано математичну модель біт-потокowego обчислювача степеневих функцій з дробовим показником степеню.

### 2.1 Метод ступінчастої апроксимації неперервних висхідних функцій

Сьогодні досить широке застосування знаходять спеціалізовані апаратні обчислювачі відтворення неперервних функцій виду що мають наступні обмеження

$$\begin{aligned} & , \\ & , \end{aligned} \tag{2.1}$$

де функція має зворотну .

При таких обмеженнях функції є монотонно зростаючими, що належать першому координатному куту на координатній площині. Їх можна умовно розділити на два підкласи: першому з них належать функції, які розташовуються нижче бісектриси першого координатного кута, а другому - вище бісектриси.

В даній атестаційній роботі для вирішення завдання дослідження спеціалізованого обчислювача степеневих функцій використовується метод ступінчастої апроксимації неперервних висхідних функцій.

Відомо, що спрощення схеми спеціалізованих пристроїв для вимірювання, керування і управління є більш важливим фактором ніж збільшення їх швидкодії, так як спеціалізовані пристрої часто працюють з

запасом за швидкістю і зниження швидкодії є для них допустимим без збитку для якості їх роботи. У зв'язку з цим технічна реалізація більшості операцій з кодами в спеціалізованих обчислювачах даного класу може бути спрощена, якщо представити вхідні і вихідні величини в біт-потоківому кодуванні.

Таким чином вхідний  $x$  і вихідний  $y$  інформаційні сигнали обчислювача, що розглядається в роботі, являють собою дві біт-потоківі послідовності відповідно  $x$  та  $y$ . При цьому періодичність подання бітів послідовності  $x$  визначається способом квантування функції, що відтворює пристрій, а періодичність подання бітів послідовності  $y$  - алгоритмом функціонування пристрою.

У випадку, коли неперервна функція  $y^*$  змінюється монотонно, на вхід обчислювача подають періодичну послідовність бітів прямокутної форми, для забезпечення рівномірного квантування аргументу цілими значеннями  $x = 1, 2, 3, \dots$

Розглянемо оптимальні режими функціонування обчислювача з точки зору часу та точності обчислення.

З точки зору часу обчислення функції  $y$ , оптимальний режим роботи обчислювача спостерігається у випадку, коли за час введення до пристрою кінцевого числа бітів  $x$ , на вході формується  $y$  вихідних бітів.

З іншого боку для забезпечення точності роботи пристрою, оптимальним режимом є той, що забезпечить для всіх цілих  $x$  граничне значення абсолютної похибки при обчисленні  $y$ , що не буде перевищувати половину одиниці молодшого біту аргументу.

В тому випадку, коли обчислювач буде виконувати функцію апроксиматора, для забезпечення оптимального режиму, задана абсолютна похибка повинна бути як для цілих так і для дробових значень аргументу.

У зв'язку з цим реалізація оптимального методу формування апроксимуючих ступінчастих функцій з точки зору точності та часу обчислення (відтворення) повинна включати основні етапи:

– вибір апроксимуючої ступінчастої функції  $y$  для заданої безперервної та граничного значення абсолютної похибки відтворення функції при її обчисленні в цілих точках;

– встановлення функціонального зв'язку  $y$  у вигляді аналітичної залежності між номерами вихідних бітів  $1, 2, 3, \dots$  послідовності  $y$  у пристрою та відповідними їм вхідними бітами  $x_1, x_2, x_3, \dots$  послідовності  $x$ ;

– створення технічного пристрою, що забезпечує формування вихідних бітів пристрою в моменти, коли біти послідовності  $x - x_1, x_2, x_3, \dots$  подаються на його вхід.

При використанні даного методу формування ступінчастих функцій, процес обчислення їх цілих значень в обчислювачі пов'язаний з операцією округлення дробових значень ґратчастої функції в цілих точках до цілих чисел. Такому обчисленню відповідає певний порядок проходження вихідних бітів пристрою, в залежності від граничного значення абсолютної похибки округлення.

У зв'язку з цим існує певна функціональна залежність між видом функції  $y$ , що відтворюється, похибкою  $\epsilon$ , номерами  $y = 1, 2, 3, \dots$  вихідних бітів пристрою та відповідними їм вхідними бітами  $x_y$  (вибірка) послідовності  $x$ .

Існування цього зв'язку дає можливість отримати формулу загального члена  $x_y$  числової послідовності  $x_1, x_2, x_3, \dots$ , що дозволить, проаналізувавши її, перейти до синтезу технічного пристрою, що здійснює процес вибірки визначених членів числового ряду з вхідної послідовності  $x$ .

Для опису математичних моделей спроектованих обчислювачів, може бути використано метод ступінчастої апроксимації неперервних висхідних функцій, що наведено у [5].

Задана неперервна функція може бути відтворена на виході обчислювача із заданою похибкою апроксимуючою ступінчастою функцією

де  $x$  – аргумент функції;  $f(x)$  – ступінчаста апроксимуюча функція;  
 – максимальна абсолютна похибка відтворення неперервної функції ступінчастою. Квадратні дужки у виразі (2.2) означають цілу частину числа. Неперервна та апроксимуюча степеневі функції приведені на рисунку 2.1.

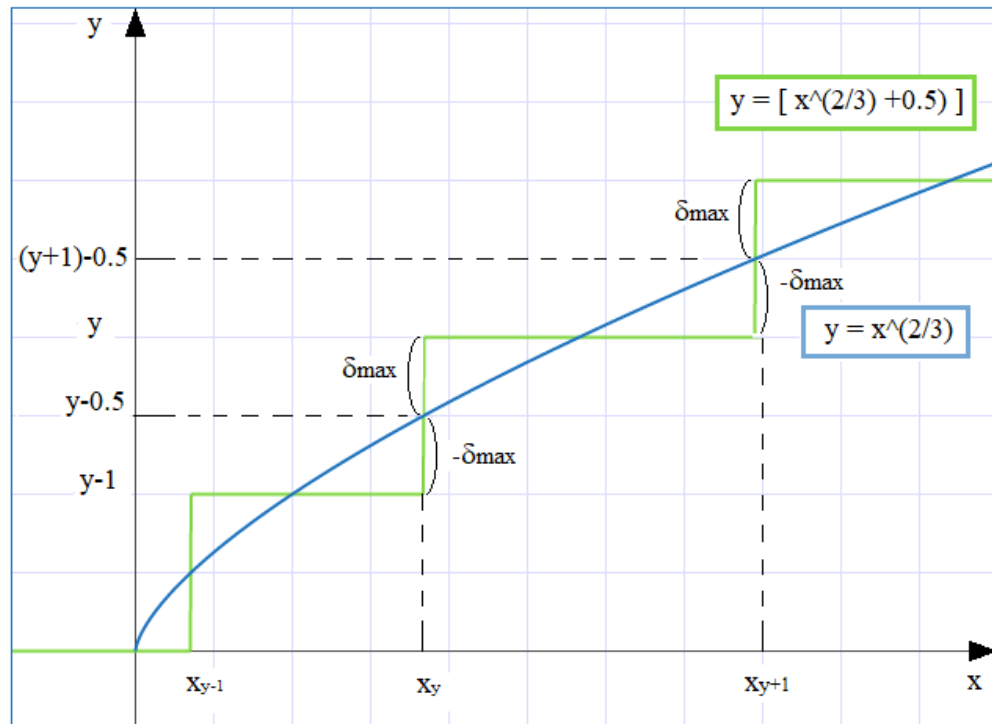


Рисунок 2.1 – Неперервна та апроксимуюча степеневі функції

Перейдемо до пошуку значень незалежної змінної  $x$ , які відповідають моментам початку формування кожної сходинки вихідної функції обчислювача. Пошук проведемо для функції , враховуючи обмеження (2.1).

Для будь-якого рівня , де , можна вказати пару сусідніх цілих значень аргументу  $x$  і  $x+1$ , для яких справедлива система нерівностей

(2.3)

Виведемо з системи (2.3) значення вибірок  $x_y$  і отримаємо формулу загального члена числової послідовності  $x_1, x_2, x_3, \dots$  що відповідає вузлам апроксимації вихідної функції  $y$

$$, \quad (2.4)$$

де - функція, зворотна

Оскільки ліва і права частина нерівності (2.4) відрізняються на одиницю, значення  $x_y$  можна отримати шляхом послідовної підстановки  $y=1,2,3, \dots$  в нерівність (2.4), та обчисленні або його лівої частини з округленням до найближчого більшого цілого числа, або обчисленні правої частини з округленням результату до найближчого меншого цілого числа. Враховуючи їх різницю в одиницю, отримаємо єдине цілочисельне значення  $x_y$ .

Таким чином, нерівність (2.4) можна замінити рівністю

$$. \quad (2.5)$$

При значеннях  $y$ , яким відповідають цілі значення  $x_y$ , нерівність (2.4) трансформується в рівність

$$. \quad (2.6)$$

При  $\epsilon$  буде досягнута мінімальна абсолютна похибка при ступінчастої апроксимації неперервних функцій, при цьому група формул (2.4), (2.5), (2.6) прийме вигляд

$$, \quad (2.7)$$

$$, \quad (2.8)$$

$$. \quad (2.9)$$

Формули (2.7), (2.8), (2.9) відповідають оптимальній апроксимації з точки зору точності обчислення значень функції в цілих точках.

Даний метод забезпечує неперервний процес відтворення функції (2.2) в реальному масштабі часу при подачі на вхід обчислювача бітового потоку аргументу  $x$ .

## 2.2 Математичне обґрунтування обчислення степеневих функцій

У відповідності до мети дослідження біт-потоків обчислювач степеневих функцій має обчислювати функцію

$$, \quad (2.10)$$

де  $x$  – аргумент функції, що представляє собою бітовий потік;

– граничне значення абсолютної похибки добування кореня третього степеню. Задана абсолютна похибка обчислень .

Обчислення з похибкою в обчислювачі забезпечує округлення результатів добування кореня до найближчих цілих чисел. При цьому забезпечується мінімально можлива похибка відтворення заданої функції, що є оптимальним [6].

Вхідним інформаційним сигналом обчислювача є біт-поток послідовність  $x$ , що подається на вхід обчислювача. На виході пристрою формується біт-поток послідовність  $y$ , яка відтворює неперервну задану степеневу функцію.

Обчислення степеневі функції зводиться до виконання операцій піднесення до степеню і добування кореня значення .

В зв'язку з цим, біт-потоків обчислювач степеневих функцій має

містити 2 блоки, в яких операції піднесення до степеню і добування кореня суміщені і виконуються одночасно.

Далі розглянемо алгоритм обчислення поліноміальних функцій, що покладено в основу побудови біт-потоківих поліноміальних обчислювачів, які входять складовою частиною в обчислювач степеневих функцій, та наведемо математичну модель обчислювача заданої функції, що була отримана на основі математичної моделі біт-потоківого обчислювача степеневих функцій довільного степеню.

### 2.2.1 Алгоритм обчислення поліноміальних функцій

Розглянемо відомий алгоритм обчислення поліноміальних функцій, що лежить в основі конвеєрних біт-потоківих обчислювачів поліномів  $n$ -го степеню виду

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n, \quad (2.11)$$

де  $n$  – ціле додатне число,  $a_i$  – коефіцієнти.

Особливістю поліномів з цілочисельними коефіцієнтами  $a_i$  є те, що послідовність його цілочисельних значень функції, що відповідають значенням  $x = 1, 2, 3, \dots, i$  є арифметичним рядом  $n$ -го порядку.

Отже, задача синтезу поліноміального обчислювача, може бути вирішена шляхом зниження порядку різниць.

Визначимо значення функції  $y_i$  (2.17), за виразом

$$y_i = a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n, \quad (2.12)$$

де  $y_i$  – значення функції в точці  $x_i$ ,  $y_{i-1}$  – значення функції в точці  $x_{i-1}$ , відповідно.

Для значення  $i$  аргументу, різниці першого порядку можна визначити за формулою

$$, \quad (2.13)$$

Різниці другого порядку можна обчислити за формулою

$$, \quad (2.14)$$

Різниці n-го порядку відповідно за формулою

$$(2.15)$$

В результаті обчислень отримано представлену в різницевих рівняннях математичну модель поліноміальної функції

$$(2.16)$$

. . . . .

$$(2.17)$$

де .

Розглянемо випадок, коли  $a_n=1$ ,  $a_{n-1} = a_{n-2} = \dots = a_0 = 0$ , тоді поліном (2.11) приймає вигляд  $y = x^n$ .

При підстановці в вираз  $x = 1, 2, 3, \dots$  значення функції  $y$  утворюють числову послідовність  $1^n, 2^n, 3^n, \dots$ , яка є арифметичним рядом n - го порядку.

При цьому арифметичний ряд різниць першого порядку утворюють шляхом послідовного вирахування з кожного наступного члена (починаючи з другого) вихідного ряду попереднього члена ряду.

Для наведеного прикладу арифметичні ряди різниць першого, другого та n-го порядку мають вигляд

(2.18)

В арифметичному ряду різниць  $n$ -го порядку члени ряду однакові та дорівнюють  $n!$

Відповідно до завдання дослідження обчислювач степеневі функції має обчислювати в підкореновому виразі функцію  $y=x^2$  [6].

При підстановці в вираз  $x = 1, 2, 3, \dots$  значення функції  $y$  утворюють числову послідовність  $1^2, 2^2, 3^2, 4^2, \dots$ , яка є арифметичним рядом другого порядку. При цьому арифметичні ряди різниць першого і другого порядків матимуть вигляд

При проектуванні спеціалізованих обчислювачів поліноміальних функцій використовують вищевикладену методику.

2.2.2 Математична модель апаратного біт-потокowego обчислювача степеневих функцій

Степеневі обчислювачі призначені для обчислення неперервних функцій

$$,$$
(2.19)

де  $m, n$  - натуральні позитивні числа.

В пристрої для обчислення степеневих функцій операції піднесення до степеню  $m$  і добування кореня  $n$ -го степеню суміщені у часі. При цьому, за

час введення до пристрою кінцевого числа бітів  $x$ , на виході обчислювача буде сформовано  $y$  вихідних бітів, які відповідають певним номерам  $x_y$ , що обираються з вхідної послідовності  $x$ .

У випадку показників степеню  $m < n$  задані функції можуть бути апроксимовані відповідними ступінчастими функціями

$$, \quad (2.20)$$

Для степеневі функції формула (2.4) буде мати вигляд

$$. \quad (2.21)$$

Перетворимо нерівність до виду

$$. \quad (2.22)$$

При підстановці в вираз (2.22), отримаємо

$$. \quad (2.23)$$

З урахуванням похибки, отримаємо

$$(2.24)$$

Нерівність (2.24) дозволить перейти безпосередньо до структурно-функціональної моделі обчислювача степеневих функцій.

Інтерполяційний спосіб формування степеневих ступінчастих функцій дозволяє одночасно формувати паралельні коди приростів функцій  $x^m$ ,  $y^n$  в темпі надходження вхідної послідовності  $x$ , безперервно порівнювати їх поточні значення і формувати вихідні біти у пристрою в момент їх рівності.



На підставі системи нерівностей (2.25) була отримана математична модель біт-потокowego обчислювача степеневі функції

(2.28)

Абсолютна похибка обчислення .

Знайдемо зворотну функцію заданої (2.28) і, використовуючи систему нерівностей (2.25), запишемо систему нерівностей, що описують математичну модель біт-потокowego обчислювача заданої функції в різницях

(2.29)

.....

,

де .

З (2.29) можна зробити висновок, що визначення  $x_y$  може бути зведено до обчислення приростів ґратчастої функції змінної  $x$  на кожному з інтервалів  $(x_{y-1}; x_y]$  і їх порівнянні з приростами ґратчастої функції з урахуванням їх різниці обчислення, отриманої на попередньому кроці обчислень в точці  $x_{y-1}$ . Отже, першому обраному біту  $y = 1$  відповідатиме певний біт з номером  $x_1$  вхідної послідовності  $x$ , при якому буде виконана перша нерівність системи (2.25). Аналогічно другому обраному біту  $y = 2$  відповідатиме біт з номером , при якому виконається друга нерівність системи (2.29) і так далі.

### 3 СТРУКТУРНИЙ СИНТЕЗ ОБЧИСЛЮВАЧА СТЕПЕНЕВИХ ФУНКЦІЙ

В розділі розглянуто принцип побудови біт-потоківих пристроїв обчислення поліноміальних функцій на основі конвеєрних архітектур, узагальнену структурно-функціональну модель відтворення степеневих функцій, представлено структурно-функціональну модель біт-потоківого обчислювача степеневих функцій на основі структурних рішень, що входять складовими блоками в архітектуру пристрою.

#### 3.1 Принцип побудови біт-потоківих апаратних обчислювачів на основі конвеєрних архітектур

В основі концепції конвеєризації обчислень лежить розбиття процесів виконання команд на менші вкладені частини, які послідовно виконуються.

Для кожної такої частини виділяється окремий блок апаратури. Кожен блок в конвеєрному ланцюжку здійснює тільки один етап виконання команди. Таким чином, обробка виконуваного обчислення розділяється на декілька етапів, сформувавши послідовну передачу даних між блоками ланцюжка.

Під час надходження чергового тактового імпульсу, кожна команда в конвеєрі просувається на наступну стадію обробки, команда що виконувалася залишає конвеєр, звільняючи місце для наступної команди. Для зберігання даних, що мають передаватися від одного етапу до іншого, використовують внутрішні проміжні буфери. Вміст буферів оновлюється на кожному такті, по завершенню етапу виконання чергової команди. Проміжні буфери забезпечують паралельну незалежну роботу блоків конвеєрного ланцюжка. Коли наступний блок починає виконувати чергову команду в ланцюжку, попередній блок може починати оброблювати наступну команду,

завдяки цьому в конвеєрі одночасно можуть оброблятися кілька послідовних команд. Конвеєрна обробка команд не зменшує час виконання окремої команди, але, завдяки використанню конвеєрної обробки велика частина обчислювального процесу відбувається в режимі паралельного виконання команд і швидкість отримання результату обробки команд збільшується пропорційно до кількості етапів конвеєра.

Оскільки в кожному такті можуть виконуватися різні стадії обробки команд, тривалість такту вибирається, виходячи з максимального часу виконання всіх стадій. Крім того, слід враховувати, що для передачі команди з однієї стадії обробки на іншу потрібен додатковий час, необхідний для запису проміжних результатів обробки в буферні регістри.

Конвеєрні системи обробки інформації є високопродуктивними системами. Значна перевага конвеєрної обробки перед послідовною має місце в ідеальному конвеєрі, в якому відсутні конфлікти і всі команди виконуються один за одним в сталому режимі, без перезавантаження конвеєра. Конвейеризація обробки команд найбільш ефективна у випадку, коли тривалість виконання всіх фаз команди приблизно однакова.

Принцип конвеєрних обчислень може бути застосований на всіх рівнях структури обчислювального процесу починаючи з рівня логічного елемента.

При створенні конвеєрних архітектур біт-потоків обчислювачів степеневих функцій, можна організувати конвеєрні обчислення для представленої в різницевих рівняннях математичної моделі поліноміальної функції (2.16), що була досліджена в розділі 2.2.1. Конвеєр може бути створений для знаходження арифметичних рядів різниць різних порядків і може бути реалізований за допомогою функціонального перетворювача розгортуючого типу.

### 3.2 Структурно-функціональна модель обчислювача поліноміальних функцій

Оскільки одним з основних вузлів біт-потокowego обчислювача степеневих функцій, що реалізує запропонований спосіб формування приростів функцій, є пристрій формування паралельних кодів функцій, розглянемо принцип дії одного з відомих і найбільш широко використовуваних для цього технічного рішення (рисунок 3.1) – пристрою обчислення поліноміальних функцій. В основі його побудови лежить алгоритм обчислення поліномів, розглянутий в 2.2.1.

Поліноміальні обчислювачі, що базуються на алгоритмі обчислення поліномів, в загальному випадку містять  $n$  паралельних накопичуючих суматорів  $SUM1, SUM2, \dots, SUMn$ ;  $n$  груп елементів  $\&1, \&2, \&3, \dots, \&n$ ;  $n-1$  елементів затримки  $DE1, DE2, DE3, \dots, DEn$  і регістр пам'яті константи  $n!$   $RG1$ .

Час затримки елементів  $DE1, DE2, DE3, \dots, DEn$  обирається із умови

Ініціалізація компонентів модуля. В початковому стані значення в  $SUM1=0$ , а в суматорах  $SUM2, SUM3, SUM4$  записані значення  $1^n, (2^n - 1^n), (3^n - 2^n) - (2^n - 1^n)$  відповідно.

Перший вхідний біт пристрою  $x=1$  надходить до групи елементів  $\&1$  і з відповідними затримками послідовно надходить до груп елементів  $\&2, \&3, \dots, \&n$ . При цьому значення кожного суматора переноситься до наступного суматора в прямому коді.

В подальшому з надходженням на вхід пристрою наступних бітів послідовності  $x$  описані процеси в схемі будуть повторюватися циклічно. В результаті в суматорі  $SUMn$  будуть формуватися члени арифметичного ряду різниць  $(n-1)$  – го порядку. В  $SUM2$  – члени арифметичного ряду різниць 1-го порядку, а в  $SUM1$  – члени арифметичного ряду поліноміальної функції  $1^n, 2^n, 3^n, 4^n, \dots$

### Рисунок 3.1 – Структурно-функціональна модель поліноміального обчислювача з паралельним виходом

Перевагою даного технічного рішення є те, що воно дозволяє відтворювати не лише функцію  $y = x^n$  для будь-якого цілого  $n$ , але і поліноми  $n$ -го степеню з довільними коефіцієнтами шляхом зміни початкових станів суматорів пристрою, що дає можливість використати цей метод для відтворення функції, яка реалізується проєктованим обчислювачем.

### 3.3 Узагальнена структурно-функціональна модель обчислювача степеневих функцій

На рисунку 3.2 приведена узагальнена структура моделі обчислювача степеневих функцій, яка реалізує функцію (2.20) з урахуванням заданої похибки .

Перетворений вираз відтворюваної функції в цілих степенях має вигляд

$$. \tag{3.1}$$

Оскільки для функції (3.1) вона може бути реалізована на базі розглянутого вище поліноміального обчислювача з цілочисельним показником ступеня.

У загальному випадку біт-потоківий обчислювач степеневі функції з дробовим показником містить перший Block 1 і другий Block 2 пристрої для відтворення поліномів цілих степенів  $m$  і  $n$  відповідно.

Відтворена функція з дробовим показником перетворена у вираз з цілими степенями. При розробці математичної моделі степеневого обчислювача отримано нерівність, яка реалізується в пристрою

(3.2)

З нерівності випливає, що значення вібірок бітів  $x_y$  можуть бути отримані шляхом формування ґратчастої функції і послідовному порівнянні її поточних значень з цілочисельними рівнями.

Рисунок 3.2 – Узагальнена структурно-функціональна модель обчислювача степеневих функцій

Узагальнена модель степеневого обчислювача містить компоненти:

- Block 1 - пристрій відтворення функції;
- Block 2 - пристрій відтворення функції;
- & - група елементів &;
- DE - елемент затримки.

Основним обчислювальним вузлом є суматор результату  $Sum_{рез}$ , який використовується в якості схеми порівняння паралельних кодів збільшень ґратчастої функції з приростами ґратчастої функції з урахуванням їх різниці, отриманої на попередньому кроці обчислень.

Формування сходинок відтворної функції здійснюється на виході суматора результату  $Sum_{рез}$ , в який з Block 1 надходять прямі бінарні коди чисел, а з Block 2 - додаткові коди чисел.

### 3.4 Структурно-функціональна модель біт-потокowego обчислювача степеневих функцій

Функція, що реалізується спроектованим пристроєм для обчислення степеневих функцій з абсолютною похибкою обчислення функції, має вигляд

$$, \quad (3.3)$$

Структурно-функціональна модель біт-потокowego обчислювача степеневих функцій наведена на рисунку 3.3. В пристрої виконуються операції піднесення до степеню і добування кореня, що суміщені в степеневому модулі.

Основна нерівність, на якій базується робота обчислювача функції (3.3) має вигляд

$$. \quad (3.4)$$

При розробці математичної моделі обчислювача степеневих функцій отримано систему нерівностей (2.28), що реалізується в пристрої.

З нерівності випливає, що значення в результаті обчислення степеневі функції можуть бути отримані шляхом формування ґратчастої функції і послідовному порівнянні її поточних значень з цілочисельними рівнями.

Розглянемо структуру і роботу пристрою.

Степневий модуль представляє собою конвеєрну архітектуру і містить підсумовуючий лічильник CNT, суматори SUM1, SUM2, SUM\_RES, регістри RG1, RG2, елементи затримки DE1, DE2, DE3, DE4 і групи елементів &1, &2, &3; &4; &5.

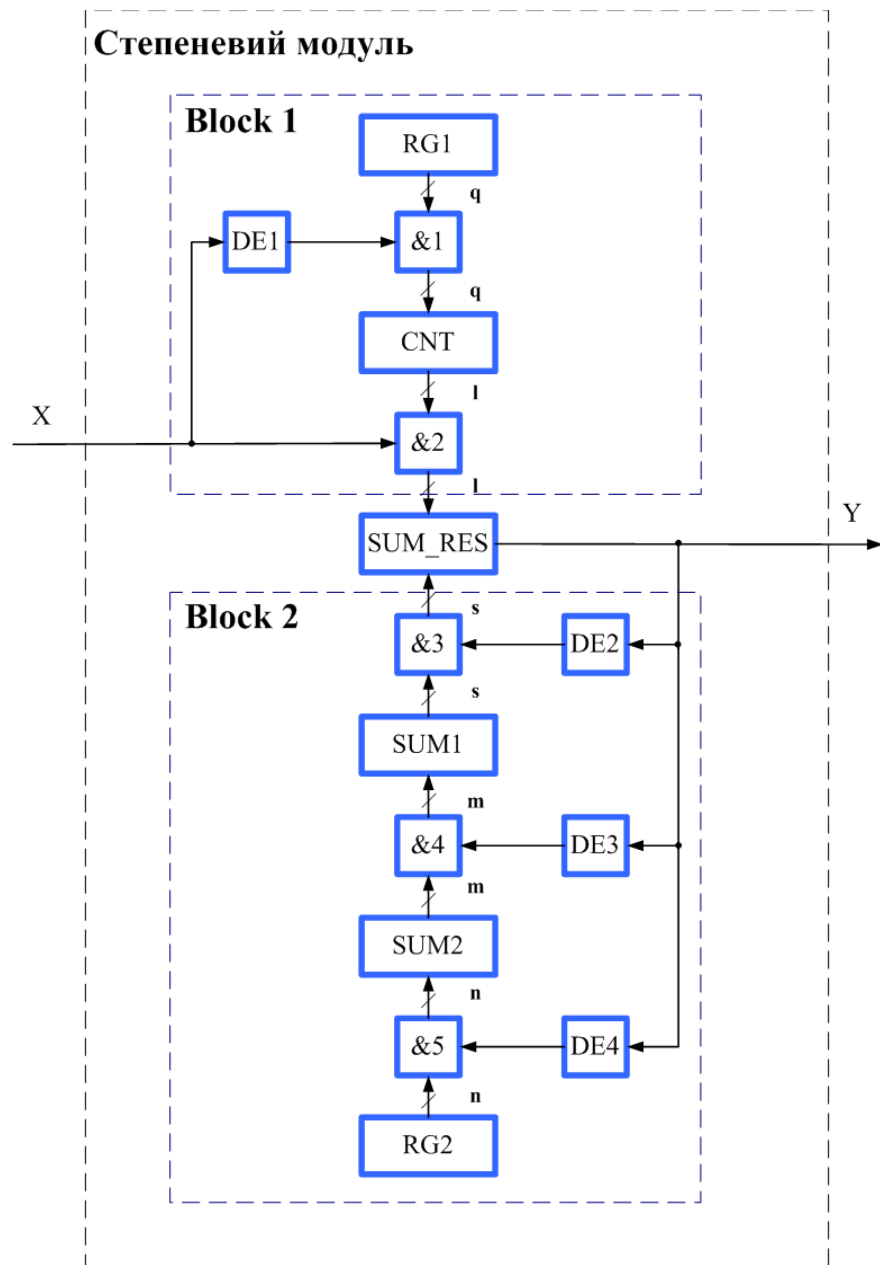


Рисунок 3.3 – Структурно-функціональна модель обчислювача степеневих функцій

Основним обчислювальним вузлом в степеневому модулі є суматор SUM\_RES, що використовується в якості схеми порівняння паралельних кодів приростів гратчастої функції з приростами гратчастої функції з урахуванням їх різниці, отриманої на попередньому кроці обчислень.

Степеневий модуль містить два блоки: Block 1, що реалізує функцію, та Block 2, що реалізує функцію .

Перерахуємо компоненти названих блоків.

В перший блок входять компоненти, які реалізують функцію :

- лічильник CNT;
- регістр RG1;
- елемент затримки DE1;
- групи елементів &1, &2.

В другий блок входять компоненти, які реалізують функцію :

- суматор SUM1;
- суматор SUM2;
- регістр RG2;
- елементи затримки DE2, DE3, DE4;
- групи елементів &3; &4, &5.

Підставляючи в ліву частину нерівності (3.4), тобто для функції, значення  $x = 1, 2, 3, \dots$ , отримаємо значення функції  $z$ , яка являє собою арифметичний ряд другого порядку, що має вигляд

$$Z: 2^3 \cdot 1^2, 2^3 \cdot 2^2, 2^3 \cdot 3^2, 2^3 \cdot 4^2, 2^3 \cdot 5^2, \dots$$

Арифметичні ряди різниць першого і другого порядків відповідно

$$2^3 \cdot (2^2 - 1^2), 2^3 \cdot (3^2 - 2^2), 2^3 \cdot (4^2 - 3^2), 2^3 \cdot (5^2 - 4^2) \dots$$

$$2^3 \cdot 2! , 2^3 \cdot 2! , 2^3 \cdot 2! , 2^3 \cdot 2! \dots$$

Отже , для функції  $2^3$  вихідний ряд чисел другого порядку

$$Z: 8, 32, 72, 128, \dots$$

Арифметичні ряди різниць першого і другого порядків

$$24, 40, 56, \dots$$

$$16, 16, \dots$$

Підставляючи в праву частину нерівності (3.4), тобто для функції значення  $y = 1, 2, 3, \dots$ , отримаємо арифметичний ряд другого порядку, що має вигляд

$$Q: 1^3, 3^3, 5^3, 7^3, 9^3 \dots$$

Арифметичні ряди різниць першого і другого порядків відповідно дорівнюють

$$3^3 - 1^3, 5^3 - 3^3, 7^3 - 5^3, 9^3 - 7^3 \dots$$

$$(5^3 - 3^3) - (3^3 - 1^3), (7^3 - 5^3) - (5^3 - 3^3), (9^3 - 7^3) - (7^3 - 5^3) \dots$$

$$\begin{aligned} &: ((7^3 - 5^3) - (5^3 - 3^3)) - ((5^3 - 3^3) - (3^3 - 1^3)), \\ &((9^3 - 7^3) - (7^3 - 5^3)) - ((7^3 - 5^3) - (5^3 - 3^3)), \dots \end{aligned}$$

Отже, для функції вихідний ряд

$$Q: 1, 27, 125, 343, 729 \dots$$

Арифметичні ряди різниць першого, другого і третього порядків

$$: 26, 98, 218, 386 \dots$$

$$: 72, 120, 168 \dots$$

$$: 48, 48 \dots$$

Block 1 та Block 2 призначені для формування кодів чисел:  $2^{3*}(2^2 - 1^2)$ ,  $2^{3*}(3^2 - 2^2)$ ,  $2^{3*}(4^2 - 3^2)$ ,  $2^{3*}(5^2 - 4^2) \dots$  та  $3^3 - 1^3$ ,  $5^3 - 3^3$ ,  $7^3 - 5^3$ ,  $9^3 - 7^3 \dots$  відповідно. Тобто різниць першого порядку числових послідовностей  $2^{3*}1^2$ ,  $2^{3*}2^2$ ,  $2^{3*}3^2$ ,  $2^{3*}4^2$  і  $1^3$ ,  $3^3$ ,  $5^3$ ,  $7^3$ ,  $9^3 \dots$  відповідно.

В процесі роботи пристрою в лічильнику CNT формуються члени арифметичного ряду різниць першого порядку числової послідовності  $2^{3*}1^2$ ,  $2^{3*}2^2$ ,  $2^{3*}3^2$ ,  $2^{3*}4^2$ ,  $2^{3*}5^2 \dots$ , в суматорах SUM1 та SUM2 - формуються члени арифметичного ряду різниць першого та другого порядку числової послідовності  $1^3$ ,  $3^3$ ,  $5^3$ ,  $7^3$ ,  $9^3 \dots$  відповідно.

Бінарний код числа, записаний в суматор CNT, надходить в SUM\_RES в прямому, а бінарний код числа, записаний в суматорі SUM1, надходить в SUM\_RES в додатковому коді.

Ініціалізація компонентів степеневого модуля:

– суматор SUM\_RES ініціалізується числом  $2^i - 1$  ( $i$  – розрядність суматора);

– лічильник CNT ініціалізується числом ;

– суматор SUM1 ініціалізується числом ;

– суматор SUM2 - ініціалізується числом ;

- реєстр RG1 ініціалізується числом , тобто числом 16;
- реєстр RG2 ініціалізується константою, утвореною в ряду різниць 2-го порядку функції , тобто числом 48.

Вихід елемента затримки DE1 з'єднаний з групою елементів &1, за допомогою якої з реєстра RG1 в CNT кожним вхідним бітом послідовності  $x$  переноситься число 16 і підсумовується з числом в лічильнику CNT.

Перенос прямого бінарного коду числа з підсумовуючого лічильника CNT в SUM\_RES здійснюється через групу елементів &2 при поданні на її елементи біту вхідної послідовності  $x$ .

Вихід пристрою з'єднаний з групою елементів &3, &4, &5 через відповідні елементи затримки DE2, DE3, DE4 таким чином, що кожен вихідний біт пристрою по ланцюгу зворотного зв'язку:

- здійснює перенос в SUM\_RES в додатковому бінарному коді число, що утворюється в SUM1 і підсумовується з вмістом суматора SUM\_RES ;
- здійснює перенос в SUM1 прямого бінарного коду числа, що утворюється в SUM2 і підсумовується з вмістом суматора SUM1;
- в SUM2 записує константу 48, що зберігається в реєстрі RG2.

Тобто в степеневому обчислювачі застосовується конвеєрні структури поліноміальних пристроїв, що забезпечують конвеєрні обчислення.

Розглянемо роботу обчислювача.

З приходом першого біту  $x_i$  на вхід пристрою виконуються наступні операції:

- прямий бінарний код числа  $2^3 \cdot 1^2$  з CNT за допомогою групи елементів &2 записується в SUM\_RES і підсумовується з його вмістом. В результаті в SUM\_RES формується число рівне При цьому SUM\_RES переповниться, на його виході з'явиться біт переповнення і в SUM\_RES залишиться число  $= 2^3 - 1 = 7$ ;

– за допомогою групи елементів &1 прямий код числа 16, записаного в RG1, переноситься в CNT і підсумовується з його вмістом. В результаті показання CNT стануть рівними  $2^3 \cdot 1^2 + 16 = 24$ . Тобто в CNT запишеться

число, відповідне першому члену арифметичного ряду різниць 1-го порядку функції . Далі при надходженні на вхід модуля бітів  $x$  в CNT будуть утворюватися члени арифметичного ряду різниць 1-го порядку функції ;

В результаті перший біт переповнення SUM\_RES буде мати місце при надходженні на вхід пристрою першого біту послідовності  $x$ . При цьому виконається перша нерівність системи (2.29) .

При відсутності біту переповнення дії повторюються з надходженням наступного вхідного біту пристрою.

Біт переповнення:

- з виходу SUM\_RES біт надходить на вихід в канал  $y$ ;
- пройшовши елемент затримки DE2, відкриває групу елементів & 3.

При цьому число, яке зберігається в SUM1 переноситься в SUM\_RES в додатковому коді і підсумовується з вмістом SUM\_RES. В результаті утворюється число . Так як , біт переповнення в SUM\_RES буде відсутній;

- пройшовши елемент затримки DE3, відкриває групу елементів &4.

При цьому з суматору SUM2 в SUM1 переноситься число, утворене в ряду різниць 2-го порядку функції і підсумовується з його вмістом. В результаті в SUM1 буде сформоване число  $5^3 - 3^3$ . Тобто в SUM1 утворюються члени ряду різниць 1-го порядку функції ;

- пройшовши елемент затримки DE4, відкриває групу елементів &5.

При цьому з регістру RG2 в SUM2 переноситься константа 48 , утворена в ряду різниць 3-го порядку функції і підсумовується з його вмістом. В результаті в SUM2 буде сформоване число  $(7^3 - 5^3) - (5^3 - 3^3)$ . Тобто в SUM2 утворюються члени ряду різниць 2-го порядку функції .

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

Другий біт переповнення SUM\_RES буде мати місце при надходженні на вхід пристрою біту з номером , при якому виконається умова і так далі.

За час вводу в обчислювач числа  $x$  на його виході з'явиться число бітів  $y$ , що відповідає заданій функції (3.3) . Оскільки в даному випадку  $m < n$ ,

число вихідних бітів  $y$  буде менше числа вхідних  $x$ , тобто в степеневому обчислювачі буде здійснюватися процес вибірки певних номерів бітів вхідної послідовності, що відповідає отриманій нерівності математичної моделі спроектованого пристрою.

### 3.5 Вибір елементної бази

FPGA має типову структуру вентиляційної матриці. ПЛІС типу FPGA фірми Xilinx виконані по SRAM КМОП технології. Характеризуються високою гнучкістю структури і достатком на кристалі тригерів. При цьому логіка реалізується за допомогою матриці так званих LUT - таблиць (Look Up Table), а внутрішні міжз'єднання - за допомогою розгалуженої ієрархії металевих ліній, комутованих спеціальними швидкодіючими транзисторами.

Програмовані логічні інтегральні схеми (ПЛІС) широко використовуються для створення цифрових систем різного призначення. Фірма Xilinx, яка є провідним світовим виробником ПЛІС, надає розробникам широкий спектр кристалів з різною технологією виробництва, ступенем інтеграції, архітектурою, швидкодією, споживаної потужністю і напругою живлення, що випускаються в різних типах корпусів і в декількох варіантах виконання, включаючи промислове, військове і радіаційно –стійке.

Кристали, що випускаються фірмою Xilinx, в повній мірі реалізують переваги ПЛІС в порівнянні з «жорсткою логікою»:

- високу швидкодію;
- можливість перепрограмування безпосередньо в системі;
- високий ступінь інтеграції, що дозволяє розмістити цифровий пристрій в одному кристалі і тим самим знизити час і витрати на трасування і виробництво друкованих плат;
- скорочення часу циклу розробки і виробництва пристрою;
- наявність потужних інструментів САПР, що дозволяють усунути можливі помилки в процесі проектування пристрою;

- порівняно низька вартість (у перерахунку на один логічний вентиль);
- можливість подальшої реалізації проектів ПЛІС для серійного виробництва у вигляді замовних НВІС, що дозволяє значно знизити їх собівартість.

До недавнього часу, не дивлячись на всі переваги ПЛІС Xilinx, існували обставини, що стримували їх застосування (особливо недорогих кристалів при розробці несерійних пристроїв) - необхідність додаткових витрат на придбання пакета програмних засобів проектування і програмування. Щоб усунути цю перешкоду, фірма Xilinx надала розробникам можливість використовувати безкоштовне програмне забезпечення - пакет WebPACK™ ISE™ (Integrated Synthesis Environment)

Програмне забезпечення ISE випускаються в чотирьох конфігураціях: Foundation ISE, BaseX ISE, Alliance ISE і WebPACK ISE. Вільно розповсюджувана серед цих чотирьох модифікацій WebPACK ISE, можливості і засоби якої і було застосовано в реалізації пристрою.

WebPACK ISE являє собою програмні засоби з системою наскрізного проектування, яка реалізує всі етапи створення цифрового пристрою на базі ПЛІС, включаючи програмування кристала: розробка проекту, синтез, моделювання, трасування і завантаження в кристал. Версія 3.3WP8.0 САІР WebPACK ISE призначена для проектування цифрових пристроїв на базі ПЛІС виробництва Xilinx, що відносяться як сімейств CPLD: XC9500, XC9500XL, XC9500XV, XCR22V10, XCR3000 (XPLA1\_3, XPLA2), XCR3000XL (XPLA3), XCR5000 (XPLA1\_5), так і FPGA: Spartan™ -II, Virtex™ -E (тільки кристал XCV300E), Virtex-II (кристали 2V40, 2V80 і 2V250).

У проекті був використаний кристал сімейства Xilinx Spartan 3E серії XC3S500E, технічні характеристики якого наведені в таблиці 3.1. Таблиця містить основні характеристики (логічні комірки, число системних вентилів, розмірність масиву CLB, к-ть CLB та ін.).

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

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

В пристрої Spartan-3E поєднано риси гнучкою, регулярної архітектури, яка охоплює матрицю конфігуруються логічних блоків (CLB), оточену програмованими блоками введення - виведення, пов'язаних між собою багатою ієрархією швидких, багатосторонніх ресурсів між'єднань. Достаток таких ресурсів дозволяє реалізовувати надзвичайно громіздкі і складні проекти. Проекти можуть працювати з системної частотою синхронізації до 200 МГц, включаючи блоки вводу / виводу (Input / Output - I / O).

Кристал містить декілька блоків ОЗУ кожен по 4Кбіта, також можлива реалізація 16 біт пам'яті на кожному 4-х входовому функціональному генераторі.

Крім цього чіпи Spartan-3E відрізняє цілий ряд переваг:

- низька вартість розроблюваних пристроїв;
- висока швидкість проектування;
- велика розмірність чіпа (до 200,000 системних вентилів);
- висока швидкодію.

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

Базовим будівельним елементом CLB блоку є логічна комірка. Логічна комірка включає 4-х входовий функціональний генератор, логіку

прискореного перенесення і запам'ятовуючий елемент. Вихід кожного функціонального генератора в кожній логічній комірці приєднаний до вихідної лінії CLB-блоку і до D-входу тригера. Кожен CLB-блок в серії Spartan-3E містить чотири логічні комірки, організовані у вигляді двох однакових секторів (Slice). На додаток до чотирьох базових логічних комірок, CLB-блок серії Spartan-3E містить логіку, яка дозволяє комбінувати ресурси функціональних генераторів для реалізації функцій від п'яти або шести входів.

Функціональні генератори реалізовані у вигляді 4-х входових функціональних таблиць (LUT). Крім використання в якості функціональних генераторів, кожний LUT-елемент може бути також використаний як синхронна пам'ять типу RAM розмірністю 16x1 біт. Більш того, з двох LUT-елементів в рамках одного сектора (Slice) можна реалізувати синхронну RAM-пам'ять розмірністю 16x2 біта або 32x1 біт, або двопортову синхронну RAM-пам'ять розмірністю 16x1 біт.

На LUT-елементі мікросхеми Spartan-3E може бути реалізований 16-бітний зсувний регістр, який ідеально підходить для захоплення високошвидкісних або пакетних потоків даних. Цей режим може також використовуватися для запам'ятовування даних в таких додатках, як цифрова обробка сигналів (Digital Signal Processing - DSP). Запам'ятовуючі елементи в кожному секторі CLB-блоку Spartan-3E можуть бути зконфігуровані як динамічні тригери (чутливі до фронту сигналу) D-типу, або як засувки, чутливі до рівня сигналу. D-вхід тригера може управлятися або від функціонального генератора в рамках того ж сектора CLB-блоку, або безпосередньо від входів даного сектора. Мультиплексор F5 в кожному секторі об'єднує виходи перетворювача. Ця комбінація дозволяє реалізувати будь-яку функцію 5-ти змінних або деякі функції до дев'яти змінних. Точно так же мультиплексор F6 об'єднує виходи з усіх чотирьох LUTів в CLB. Це дозволяє реалізувати будь-яку функцію 6-ти змінних або деякі функції до 9 змінних. У кожного CLB є чотири прямих шляхи. Ці шляхи забезпечують

додаткові лінії введення даних або додаткову маршрутизацію, яка дозволяє заощадити логічні ресурси. Спеціальна логіка перенесення забезпечує можливість швидкого перенесення при реалізації арифметичних функцій. CLB-блок Spartan-3E підтримує дві окремі ланцюжки перенесення, одна на кожен сектор (Slice) CLB. Розмірність ланцюжків перенесення - два біти на CLB-блок. У FPGA Spartan-3E вбудована особлива блокова пам'ять. Вона створена на додаток до розподіленої пам'яті невеликої ємності (Select RAM), реалізованої на функціональних таблицях (Look Up Table RAM - LUTRAM).

## 4 АПАРАТНА РЕАЛІЗАЦІЯ ДОСЛІДЖУВАНОЇ МОДЕЛІ ОБЧИСЛЮВАЧА

У даному розділі досліджено обчислювальний процес в компонентах пристрою, розроблено граф-схему алгоритму роботи обчислювача, розроблено апаратну модель на мові VHDL з використанням автоматної моделі опису. На етапі верифікації отриманого рішення використано автоматизоване тестове середовище (тест-бенч). Модель реалізована в програмовану логічну інтегральну схему Xilinx Spartan 3E.

### 4.1 Специфікація досліджуваного обчислювача

Для автоматизації досліджуваної моделі була створена експериментальна апаратна реалізація пристрою з бітовим потоком даних, що обчислює функцію .

В якості наочної демонстрації прикладу роботи пристрою пропонується перевірити його працездатність. У таблиці 4.1 вхідні та вихідні параметри відтворюваної функції у.

Таблиця 4.1 – Вихідні дані

X	M	N
10	2	3

Опис інтерфейсу біт-потокowego пристрою наведено у лістингу 4.1.

Лістинг 4.1.– Опис інтерфейсу біт-потокowego пристрою

```
entity powerfunc is
    generic(
        width1: natural := 16;
```

```

        width2: natural := 24);
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 entity powerfunc;

```

На вхід обчислювача подається біт-потоківна послідовність  $x$ , яка є вхідним інформаційним сигналом. На виході пристрою формується біт-потоківна послідовність  $y$ , яка відтворює неперервну задану функцію з похибкою, що не перевищує половини одиниці молодшого розряду при виконанні операції піднесення до дробового степеню.

#### 4.2 Результати обчислення заданої функції та обчислювальний процес в компонентах пристрою

Розглянемо етапи розрахункової частини при виконанні апаратної реалізації досліджуваного обчислювача з заданими параметрами та обчислювальний процес в компонентах пристрою.

Нижче приведемо результати обчислень заданої функції для вказаних вище значень  $x$ .

При подачі на вхід обчислювача вхідної біт-потоківної послідовності  $x_{\max}=10$  значення функції будуть наступні:

- при  $x = 1$
- при  $x = 2$
- при  $x = 3$
- при  $x = 4$

- при  $x = 5$
- при  $x = 6$
- при  $x = 7$
- при  $x = 8$
- при  $x = 9$
- при  $x = 10$

Значення вибірок визначаються з метою знаходження номерів біт, які обираються з вхідного бітового потоку  $x$  та подаються на вихід пристрою на основі виразу (2.21)

(4.1)

При підстановці в (4.1) значень  $y = 1, 2, 3, 4, 5$  отримаємо значення вибірок  $= 1, 2, 4, 7, 10$  відповідно.

Нерівність, яка реалізується в пристрої .

Підставляючи в ліву і праву частину нерівності значення  $x, y = 1, 2, 3, \dots, 10$ , отримаємо арифметичні ряди 2-го та 3-го порядків відповідно.

Для функції  $2^3 x^2 y$ , арифметичний ряд 2-го порядку має вид

8 , 32 , 72 , 128 , 200 , 288, 392 , 512 , 648 , 800

Арифметичні ряди різниць першого і другого порядків

24 , 40 , 56 , 72 , 88 , 104 , 120 , 136 , 152

16 , 16 , 16 , 16 , 16 , 16 , 16 , 16.

Підставляючи у вираз значення  $y = 1, 2, 3, \dots, 10$ , отримаємо послідовність значень функції , яка є арифметичним рядом 3-го порядку

1 , 27 , 125 , 343 , 729 , 1331 , 2197 , 3375 , 4913 , 6859

Арифметичні ряди різниць першого другого і третього порядків для цієї послідовності мають вид

: 26 , 98 , 218 , 386 , 602 , 866 , 1178 , 1538 , 1946

: 72 , 120 , 168 , 216 , 264 , 312 , 360 , 408

: 48 , 48 , 48 , 48 , 48 , 48 , 48.

Визначимо числа, якими ініціалізуються компоненти обчислювача.

$$RG1 = 16;$$

$$CNT = 1 * 2^3 = 8;$$

$$SUM\_RES = 2i - 1;$$

$$SUM1 = 3^3 - 1^3 = 26;$$

$$SUM2 = = 72;$$

$$RG1 = = 16;$$

$$RG2 = = 48.$$

Розглянемо обчислення в досліджуваному пристрої (таблиця 4.2). Результати обчислення функції у і поява бітів переповнення у на виході пристрою співпадають.

Таблиця 4.2 – Обчислювальний процес в компонентах пристрою

X	SUM_RES	Біт переповнення	CNT	SUM1	SUM2	CNTres
1	-1+8 = 7	1	8+16 = 24	26+72 = 98	72+48 = 120	1
	7-26 = -19					
2	-19+24=5	1	24+16 = 40	98+120 = 218	120+48 = 168	2
	5-98 = -93					
3	-93+40 = -53		40+16=56			
4	-53+56=3	1	56+16 = 72	218+168=386	168+48=216	3
	3-218 = -215					
5	-215+72 = -143		72+16 = 88			
6	-143+88 = -55		88+16 = 104			
7	-55+104 = 49	1	104+16 = 120	386+216 = 602	216+48 = 264	4
	49-386 = -337					
8	-337+120 = -217		120+16 = 136			
9	-217+136 = -81		136+16 = 152			
10	-81 +152 = 71	1	152+16 = 168	602+264 = 866	264+48 = 312	5
	71-602 = -531					

### 4.3 Граф-схема алгоритму роботи біт-потокowego пристрою

Автоматна модель досліджуваного спеціалізованого обчислювача представлена композицією управляючого і операційного автоматів.

За структурно-функціональною моделлю пристрою і її математичним описом була розроблена змістовна граф-схема алгоритму його роботи (рисунок 4.2).

Рисунок 4.2 – Змістовна граф-схема алгоритму роботи пристрою

Модель працює за наступним алгоритмом:

а) при скиданні пристрою (сигнал  $reset = 1$ ) регістри пристрою встановлюються в такі значення:

- 1)  $CNT = 8$ ;
- 2)  $SUM\_RES = -1$ ;
- 3)  $SUM1 = 26$ ;
- 4)  $SUM2 = 72$ ;
- 5)  $RG1 = 16$ ;
- 6)  $RG2 = 48$ ;
- 7)  $count = 0$ ;

б) при надходженні чергового біту:

- 1) значення  $SUM\_RES$  збільшується на значення суматора  $CNT$ ;
- 2) значення  $CNT$  збільшується на 16;

в) якщо значення регістра  $SUM\_RES$  невід'ємне, то

- 1) на виході пристрою генерується вихідний біт;
- 2) значення лічильника бітів  $count$  збільшується на одиницю;
- 3) значення регістра  $SUM\_RES$  віднімається значення регістра  $SUM1$ ;
- 4) до значення регістра  $SUM1$  додається значення регістра  $SUM2$ ;
- 5) до значення регістра  $SUM2$  додається 48;

г) пункт в) повторюється до тих пір, поки значення регістра  $SUM\_RES$  невід'ємне.

#### 4.4 Граф переходів управляючого автомату пристрою

В спеціалізованому обчислювачі арифметичний блок представлено композицією операційного і управляючого автоматів. Управляючий автомат описується графом переходів, що був отриманий в результаті синтезу граф – схеми алгоритму для автомата Мура (рисунок 4.3) [11]. Граф переходів автомата має три стани:  $a_0$ ,  $a_1$ ,  $a_2$ .

Переходи між станами в досліджуваній моделі відбуваються наступним чином:

- по сигналу  $reset = 1$  автомат переходить в початковий стан  $a_0$ . Автомат знаходиться в стані  $a_0$  доки немає сигналу  $impulse$ , який надходить з вхідного буфера;

- в стані  $a_1$  за допомогою автомата значення суматора  $SUM\_RES$  збільшується на значення суматора  $CNT$ , а значення суматора  $CNT$  збільшується на 16. Автомат входить в стан  $a_2$ , коли на виході апроксиматора з'являється сигнал наявності біта на виході результуючого суматора;

- у стані  $a_2$  автомат збільшує на одиницю значення  $Count$ , значення суматора  $SUM1$  віднімається від значення суматора  $SUM\_RES$ , до значення суматора  $SUM1$  додається значення регістра  $SUM2$ , і з  $RG2$  число 48 додається до значення суматора  $SUM2$ . В цьому стані також виводиться сигнал для вихідного буфера для генерації біту на виході пристрою.

Рисунок 4.3 – Граф переходів автомата Мура арифметичного блоку

#### 4.5 Структурно-блокова схема досліджуваного обчислювача

На рисунку 4.4 представлена структурно-блокова схема пристрою. У схемі використовується два основних блоки: детектор вхідного біту і блок спеціалізованого обчислювача.

Детектор вхідного біту . Для того, щоб виявити біт на вході  $x$ , використовується детектор бітів. Часова діаграма детектування вхідного біту представлена на рисунку 4.5.

Біт детектується по двом сусіднім тактам сигналів  $clock$ .

Детектор виявить біт, якщо на черговому такті значення сигналу  $x = 0$ , а на наступному такті значення  $x = 1$ . В результаті цього на виході встановиться відповідний сигнал  $impulse = 1$ .

Рисунок 4.4 – Структурно-блокова схема пристрою

Рисунок 4.5 – Часова діаграма детектування вхідного біту

В протилежному випадку сигнал  $impulse$  буде встановлено в 0. Далі цей біт надходить до обчислювача.

Таким чином, видно, що для виявлення вхідного біту, необхідно, щоб сигнал  $clock t_{clk}$  був меншій або дорівнював періоду вхідного біту.

Якщо період вхідного біту  $t_1$  менше до періоду синхронізації, такі біти не зможуть бути задетектовані детектором бітів ( $t_1 < t_{clk}$ ).

В таблиці 4.3 представлений опис входів детектора вхідного біту.

Таблиця 4.3 – Вхідні і вихідні параметри для детектора вхідного біту

Сигнал	Опис	Режим
--------	------	-------

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

Часова діаграма роботи детектора вхідного біту представлена на рисунку 4.6. До моменту 225 нс сигнал ready = 0. В цей час арифметичний блок не готовий приймати вхідні біти, оскільки ще не завершена робота над обробкою попередніх бітів. Детектор бітів накопичує останні за допомогою регістра counter. На моменті 240 нс сигнал ready змінює своє значення на ready = 1, що говорить про те, що арифметичний блок готовий приймати вхідний біт. Детектор вхідних бітів виставляє на вихід одиницю і зменшує значення сигналу counter на одиницю. Це відбувається до моменту 400 нс, коли внутрішній лічильник стає рівним нулю.

Блок спеціалізованого обчислювача. Блок «Powerfunc» виконує операції піднесення аргументу  $x$  до дробового степеню з заданою абсолютною похибкою обчислення, тобто в біт-потоківному обчислювачі результат ділення округлюється до найближчого цілого числа. На виході обчислювача формується вихідна біт-потоківна послідовність. Після того, як з'явився сигнал Ready на виході арифметичного блоку, рівний '1', це означає, що даний блок видає сигнал готовності для прийняття пристроєм наступного вхідного біту. Результатом його роботи є сигнал DataOut, який являє собою результат обчислення заданої функції.

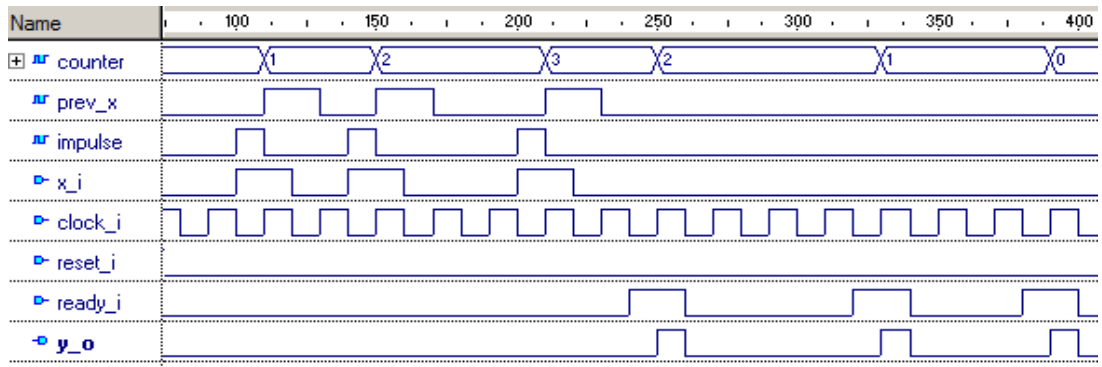


Рисунок 4.6 – Часова діаграма роботи вхідного буфера

Вхідні і вихідні параметри для даного блока представлені в таблиці 4.4

Таблиця 4.4 – Вхідні та вихідні параметри для блока «Powerfunc»

Сигнал	Опис	Режим
Data	Біт-поточкова вхідна послідовність	IN
Reset	Установка пристрою в початковий стан	IN
Clock	сигнал синхронізації	IN
DataOut	Біт-поточковий вихідний сигнал, який є результатом обчислення функції	OUT
Count	Значення функції в двійковому N-розрядному коді на виході суматора	OUT

Інтерфейс даного блока наведений на рисунку 4.7.

В якості ефективного середовища моделювання та верифікації обрано програмний пакет Active-HDL - середовище розробки, моделювання та верифікації проектів для [програмованих логічних інтегральних схем](#), розроблене фірмою [Aldec](#). Він дозволяє автоматизувати процес введення проекту і Test Bench до нього, а також ефективно здійснювати аналіз проектних рішень. Програма також дозволяє створювати описи пристроїв за допомогою [мов опису апаратури](#), а також за допомогою структурних схем,

створювати графічні моделі кінцевих автоматів та конвертувати HDL опис в графічні структурні схеми і назад.

#### Рисунок 4.7 – Інтерфейс блока «Powerfunc»

Для синтезу проекту вибраний програмний продукт фірми Synplicity Synplify 7.0.2. Продукт Synplify є інструментом синтезу RTL, спеціально розробленим для ПЛІС і CPLDs. Інструменти синтезу також включають в себе вбудований RTL Debug (Ідентифікувати набір інструментів), який є двокомпонентною системою, що є важливою частиною процесу проектування HDL.

Продукт Synplify 7.0.2. має такі переваги порівняно з програмами синтезу інших виробників:

- порівняно висока швидкість синтезу;
- візуальне відтворення результатів синтезу;
- наявність бібліотек під сучасну елементну базу найбільших світових виробників.

Хоча створення проекту за допомогою мовного опису на МОА вимагає більше часу і знань специфіки використовуваної МОА, він є найбільш гнучким варіантом.

#### 4.6 Опис блоків проекту МОА

В даному проекті в якості мови апаратних засобів була вибрана мова VHDL. Такий вибір обумовлений наявністю ефективних програм для реалізації моделювання та синтезу (Active-HDL , Synplify 7.0.2.).

Оскільки спроектований пристрій має в собі багато компонентів, для опису проекту мовою VHDL, представимо складний проект в вигляді ієрархічної структури, що дає змогу декомпонувати проект і деталізувати його опис на більш низьких рівнях. Таким чином буде спрощений процес написання окремих компонентів проекту і процес їх відладки. На основі структурно-функціональної схеми (рисунок 3.2), була визначена ієрархія проекту, що представлена на рисунку 4.8.

Рисунок 4.8 – Файлова ієрархія проекту

Файлова структура містить:

Powerfunc.top.vhdl,  
Powerfunc.detector.vhdl, Powerfunc.vhdl,  
Powerfunc.ua.vhdl, Powerfunc.oa.vhdl.

#### 4.7 Опис проекту МОА

Проект складається з великої кількості файлів що ієрархічно пов'язані між собою. Найвищим рівнем ієрархії є файл Powerfunc.top.vhdl, який об'єднує файли нижніх рівнів. Файл складається з компонентів, що поєднані між собою за допомогою операторів port map, які є зв'язуючими файли нижньої ієрархії. Також він є реалізацією на мові опису апаратури специфікації всього пристрою. Файл складається з entity, в середині якого описані всі вхідні і вихідні сигнали даного продукту. Опис файлу «Toplevel» наведено в лістингу 4.4.

#### Лістинг 4.4 – Опис файлу «Toplevel»

```

entity powerfunc_toplevel is
generic(
    width1: natural := 16;
    width2: natural := 24);
port(
    reset_i: in std_logic;
    clock_i: in std_logic;
    x_i: in std_logic;
    y_o: out std_logic;
    count: out std_logic_vector(width1-1 downto 0));
end powerfunc_toplevel;

```

В лістингу 4.5 представлені компоненти нижнього рівня: Powerfunc.vhdl, Powerfunc.ua.vhdl, Powerfunc.oa.vhdl, Powerfunc.detector.vhdl.

#### Лістинг 4.5 – Опис файлів нижнього рівня

```

architecture struct of powerfunc_toplevel is
-- Component declaration of the "aproximator(struct)" unit defined in
-- file: "./src/aproximator.vhd"
component powerfunc
    generic(
        width1: natural := 16;
        width2: natural := 24);
        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 " detector (beh)" unit defined in
-- file: "./src/ detector.vhd"
component powerfunc _ detector
    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;

```

Далі описані з'єднання блоків " Powerfunc.top" (Лістинг 4.6).

Лістинг 4.6 – Опис з'єднання блоків

```

signal detector _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
    detector _1 : powerfunc _ detector
    generic map(
        width => width1
    )
    port map(

```

```

    x_i => x_i,
    clock_i => clock_i,
    reset_i => reset_i,
    ready_i => ready,
    y_o => detector_out
);
powerfunc_111 : powerfunc
generic map(
    width1 => width1,
    width2 => width2)
port map(
    x_i => detector_out,
    ready_o => ready,
    clock_i => clock_i,
    reset_i => reset_i,
    y_o => powerfunc_out,
    sum_o => sum_t);

```

В лістингу приведений фрагмент програми, що описує роботу операційного автомату блоку «Powerfunc.oa»:

Лістинг 4.7 – Опис роботи операційного автомата

```

process (clock_i, reset_i)
begin
if (reset_i = '1') then
    counter <= CONV_STD_LOGIC_VECTOR(8, width1);
    sum_1 <= CONV_STD_LOGIC_VECTOR(-1, width2);
    sum_2 <= CONV_STD_LOGIC_VECTOR(26, width2);
    sum_3 <= CONV_STD_LOGIC_VECTOR(72, width1);
else

```

```
    if (falling_edge(clock_i)) then
        if (sum_plus_a_i = '1') then
            sum_1 <= sum_1 + counter;
            counter <= counter + 16; -- add our constant
        else
            if (sum_minus_b_i = '1') then
                count <= count + 1;
                sum_1 <= sum_1 - sum_2;
                sum_2 <= sum_2 + sum_3;
                sum_3 <= sum_3 + 48;
            end if;
        end if;
    end if;
end process;
end beh;
```

#### 4.8 Верифікація та тестування роботи пристрою

Для проведення верифікації, апаратної реалізації, що була отримана, був розроблений test bench— тестове середовище мовою VHDL. Test bench являє собою модуль описаний мовою VHDL, у якого відсутній інтерфейс, але він має примірник тестового модуля і здійснює подачу тестових впливів і аналіз тестових реакцій модуля, що тестується. На рисунку 4.9 відображені ієрархічне відношення test bench і модуля, що тестується та їх взаємодія.

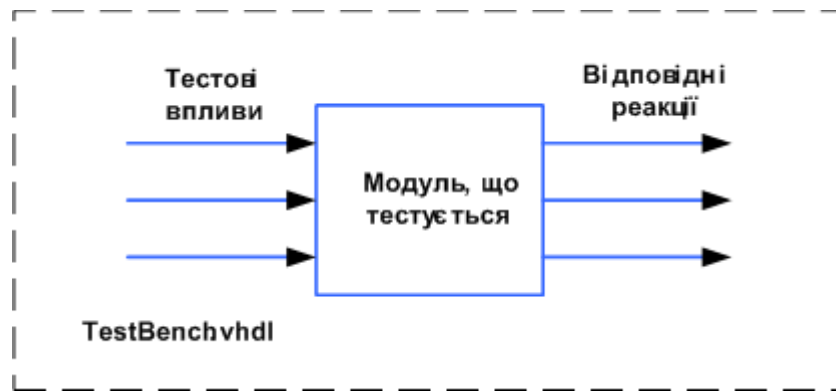


Рисунок 4.9 – Відношення тест-бенча і модуля, що тестується

Подібно до цифрової схеми, структура тестової програми при цьому описується так само за допомогою мови VHDL.

На рисунках 4.10 та 4.11 представлені результати моделювання поведінкової моделі пристрою. Відповідно до плану верифікації відбувається остання. Більш детальне представлення верифікації режимів роботи пристрою представлено на часовій діаграмі.

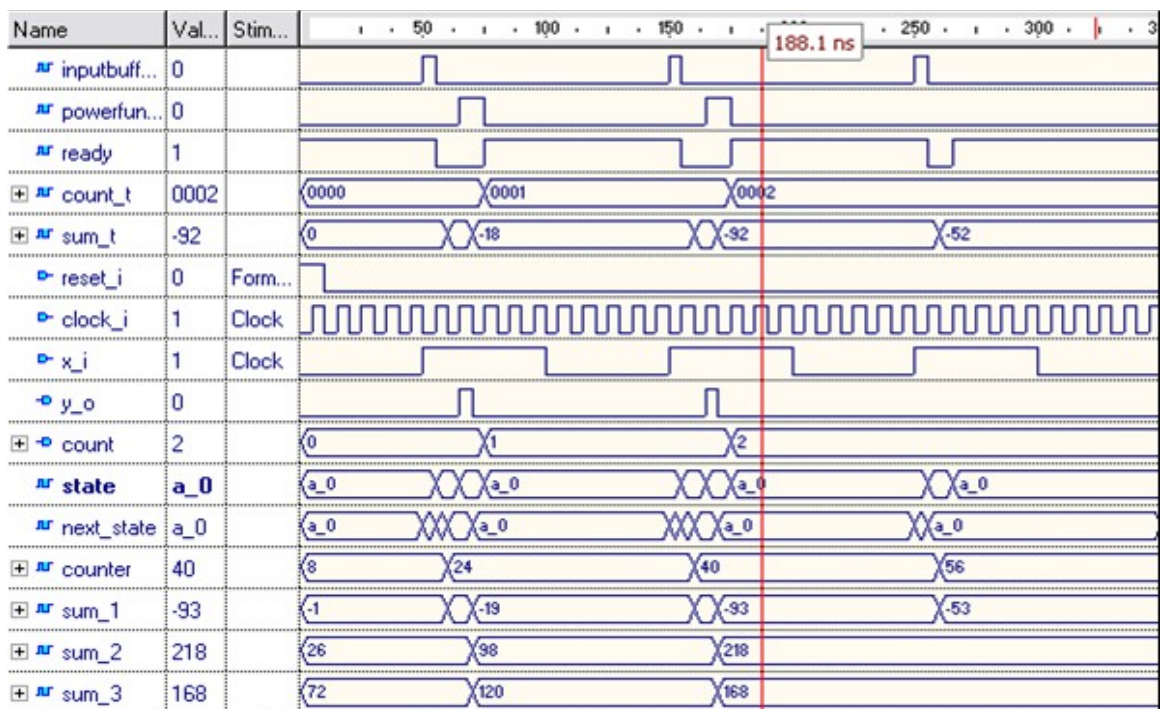


Рисунок 4.10 – Результати моделювання поведінкової моделі пристрою

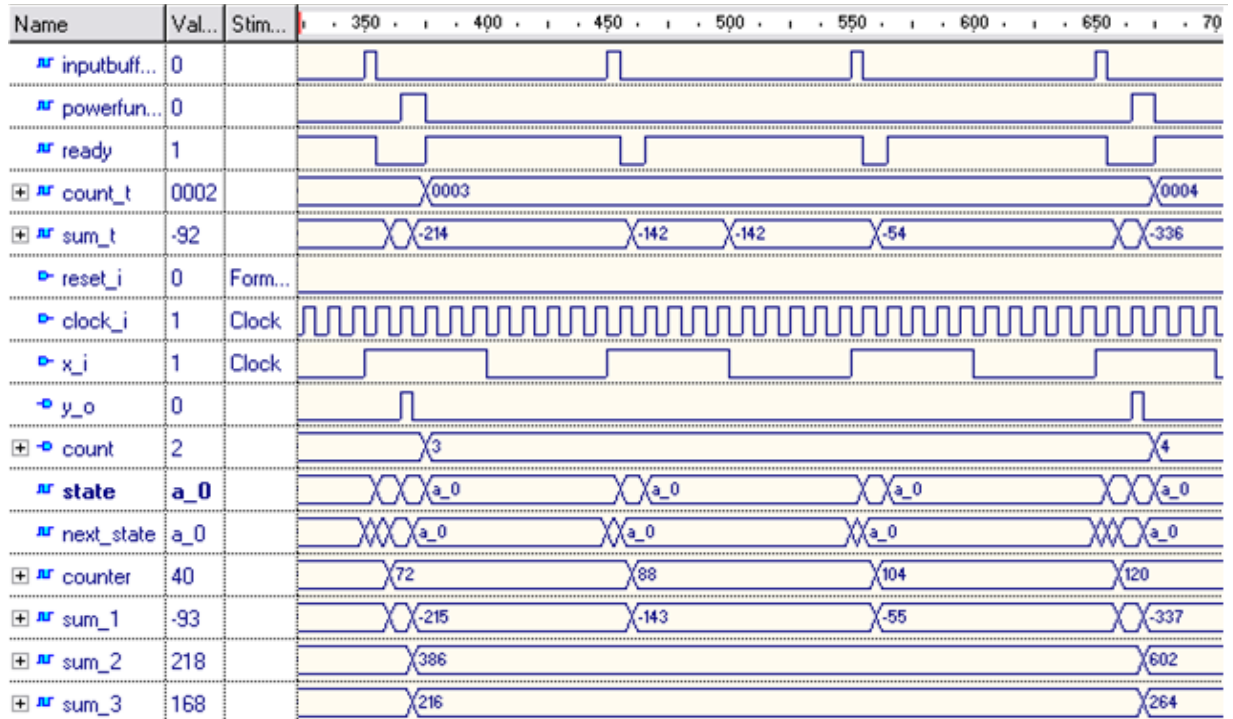


Рисунок 4.11 – Результати моделювання поведінкової моделі пристрою обчислення степеневих функцій

Отже, результати експерименту співпадають з результатами обчислень, наведеними в таблиці 4.2, що підтверджує правильність роботи пристрою.

Часова діаграма демонструє, що при подачі на вхід обчислювача 10 бітів вхідного бітового потоку  $x$  на виході пристрою згенеровано 5 бітів вихідного бітового потоку  $y$ . Значення в регістрах компонентів пристрою співпадають з даними обчислювального процесу.

Після перевірки режимів верифікації даного пристрою та порівняння їх з очікуваними, можна приступати до наступного етапу проектування - імплементції моделі пристрою.

#### 4.9 Імплементція моделі пристрою

Велика вартість мікросхем FPGA з вбудованою RAM в порівнянні з вартістю замовлених мікросхем обмежує використання FPGA для виготовлення дослідних зразків або дрібносерійної продукції. Цей недолік

FPGA усунутий фірмою Xilinx випуском на початку 1998 року нової серії мікросхем FPGA - серії Spartan.

Архітектура серії Spartan заснована на архітектурі серії XC4000 і включає аналогічні КЛБ (конфігуровані логічні блоки) і організацію з'єднань між ними.

Завдяки поліпшенням в технології, зниження споживаної потужності, поліпшень в розміщенні проекту в кристалі, серія Spartan може працювати на системній частоті 80МГц, внутрішній частоті більш 150МГц, і вартість кристалів Spartan порівнянна з вартістю замовних кристалів.

Для зниження вартості в архітектуру Spartan в порівнянні з архітектурою XC4000 були внесені зміни. Були прибрані деякі рідко використовувані властивості серії XC4000, такі як підтримка асинхронного RAM, розподілені дешифратори і можливість паралельної завантаження. Були додані декілька вбудованих ланцюгів тестування для спрощення перевірки роботи проекту і зниження вартості зовнішнього тестового обладнання. Також, малі розміри кристала дозволяють виробляти корпусування в більш дешеві, меншого розміру корпусу з числом контактів від 90 до 304. Спрощений САПР дозволяє знизити час проектування і тестування, тим самим знижується вартість кінцевого продукту.

Сімейство Spartan-3 програмованих логічних інтегральних схем (ПЛІС) є подальшим розвитком сімейства Spartan-IIЕ У приладах Spartan-3 збільшені логічні ресурси, ємність внутрішньої пам'яті, загальне число призначених для користувача блоків введення-виведення і системну продуктивність, покращено керування синхронізацією.

ПЛІС Spartan-3 надає повний і гнучкий контроль над частотою, фазою і розфазуванням синхронізуючих бітів. Контроль здійснюється за допомогою модулів автопідстроювання затримки (DLL), що входять до складу цифрових блоків управління синхронізацією (DCM).

Збільшення відношення логічної ємності до кількості блоків введення-виведення дозволило істотно знизити собівартість кристалів в перерахунку на один логічний осередок.

Завдяки своїй низькій вартості, ПЛІС FPGA сімейства Spartan-3E ідеально підходять для застосування в різних областях, таких як широкосмуговий доступ, домашні мережі, засоби візуалізації (монітори / проектори) і цифрове телевізійне обладнання.

Розробка проекту на базі ПЛІС серії Spartan-3 підтримується пакетом програмного забезпечення Xilinx ISE починаючи з версії 6.1i.

Синтез проводився з використанням системи синтезу Synplify 7.0 Pro фірми Synplicity. Результатом синтезу є RTL схема вентильного рівня, схема Technology View і файл звіту.

Була виконана імплементація проекту, заздалегідь створений UCF-файл наведено нижче (лістинг 4.8).

#### Лістинг 4.8 – UCF- файл

```
NET "rst" LOC = "H13" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "St" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "LED<7>" LOC = "F9" | IOSTANDARD = LVTTTL | SLEW = SLOW
| DRIVE = 8 ;
NET "LED<6>" LOC = "E9" | IOSTANDARD = LVTTTL | SLEW = SLOW
| DRIVE = 8 ;
NET "LED<5>" LOC = "D11" | IOSTANDARD = LVTTTL | SLEW =
SLOW | DRIVE = 8;
NET "LED<4>" LOC = "C11" | IOSTANDARD = LVTTTL | SLEW =
SLOW | DRIVE = 8;
```

```
NET "LED<3>" LOC = "F11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8;
```

```
NET "LED<2>" LOC = "E11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8;
```

```
NET "LED<1>" LOC = "E12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8;
```

```
NET "LED<0>" LOC = "F12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8;
```

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

На рисунках 4.12, 4.13, 4.14 наведені RTL - схема пристрою, RTL схема пристрою нижнього рівня пристрою та RTL схема арифметичного блока пристрою.

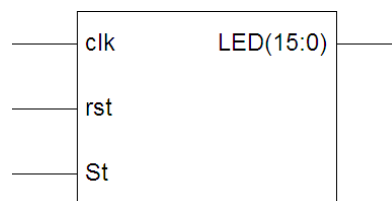


Рисунок 4.12 – RTL схема пристрою

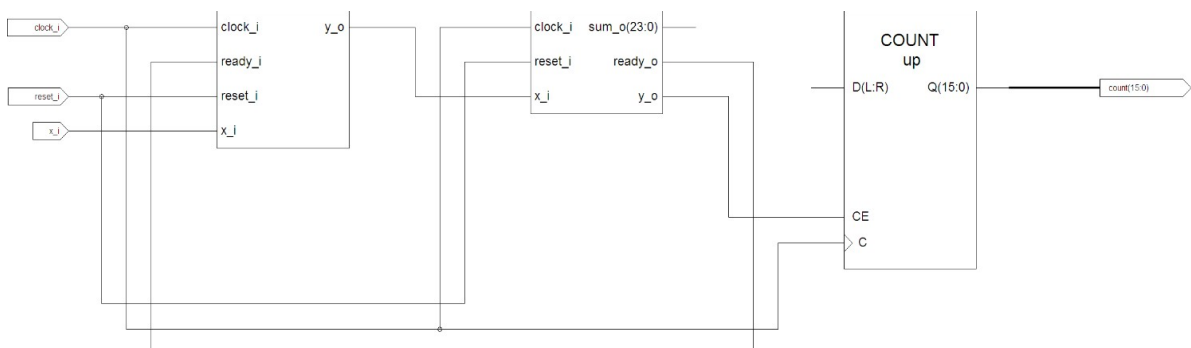


Рисунок 4.13 – RTL схема пристрою нижнього рівня

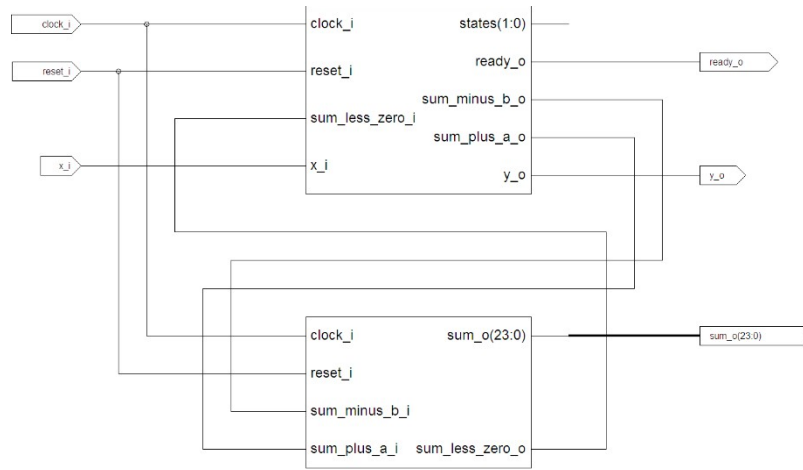


Рисунок 4.14 – RTL схема арифметического блока пристрою

## ВИСНОВКИ

Атестаційна робота присвячена розробці і дослідженню апаратного обчислювача степеневих функцій, аргументом якого є бітовий потік даних.

В роботі проведений аналіз особливостей функціонального перетворення бітових потоків в апаратних обчислювачах математичних функцій.

Розглянутий в атестаційній роботі метод ступінчастої апроксимації забезпечує мінімально можливий час відтворення функцій, що визначається часом введення аргументу в обчислювач при мінімально можливій похибці обчислення поточних значень функцій для цілих значень аргументу, що не перевищує половини одиниці його молодшого біту. Застосування розглянутого методу дозволяє здійснювати функціональне перетворення в реальному масштабі часу, по мірі надходження бітового потоку даних на вхід обчислювача, а також дозволяє підвищити точність і час обчислення заданої функції.

На основі методу ступінчастої апроксимації неперервних степеневих функцій розроблено математичну модель біт-потокowego обчислювача степеневих функцій, що представляє собою систему різницевих нерівностей. В якості основного компонента порівняння при реалізації нерівностей математичної моделі використано накопичуючий суматор.

Розроблено структурно-функціональну модель пристрою, в якому здійснено синтез операцій піднесення до степеню і добування кореня в одному пристрої.

Була створена експериментальна апаратна реалізація досліджуваної моделі обчислювача, виконані необхідні розрахунки, розроблена змістовна граф-схема алгоритму, яка була закодована для синтезу автомата Мура, розроблено граф переходів управляючого автомату. Апаратна модель

досліджуваного обчислювача, що представлена композицією управляючого і операційного автоматів забезпечує надійність функціонування пристрою. Було здійснено опис проекту для введення в САПР. За граф-схемою алгоритму та графом переходів з використанням стандартних шаблонів коду, було розроблено модель пристрою мовою опису апаратури VHDL.

Для верифікації отриманого рішення, був розроблений і використаний тест-бенч. Отримано поведінкову модель пристрою, та шляхом наочної демонстрації прикладу його роботи було продемонстровано, що результати моделювання поведінкової моделі обчислювача степеневих функцій співпадають з результатами обчислень в компонентах пристрою. Модель була синтезована засобами САПР Xilinx. В якості оптимальної платформи для імплементації досліджуваного пристрою, що відповідає вимогам, висунутим до САПР, було обрано кристал типу FPGA Spartan 3E серії XC3S500E.

Наукова новизна роботи полягає в розробці математичної моделі, архітектури та апаратної реалізації біт-потокowego обчислення степеневих функцій на технологічній платформі ПЛІС з використанням САПР на основі мов опису апаратури, для підвищення ефективності проектування біт-потокowego пристрою.

Біт-потоковой обчислювач степеневих функцій може бути застосований в системах управління, контролю, інформаційно вимірювальних системах та в системах, що потребують здійснення первинної математичної обробки даних, отриманих з датчиків фізичних величин.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шкіль, О. С. Декомпозиція математичної моделі біт-потокowego обчислювача ірраціональних функцій / О. С. Шкіль, Б. Д. Ларченко, Л. В. Ларченко. // Радіоелектроніка та інформатика. – 2019. – №4. – С. 34–39.
2. Стахів, М. Ю. Цифрові функціональні перетворювачі розгортуючого типу з покращеними характеристиками : автореф. дис. канд. техн. наук : 05.13.05 / М. Ю. Стахів; Нац. ун-т «Львів.політехніка». – Л., 2013. – 21 с.
3. Буренева, О. И. Многофункциональный бит-потокový преобразователь / О. И. Буренева, О. А. Жирнова // Известия ЛЭТИ. – 2019. – №10. – С. 46–53.
4. Попов, А. Ю. Проектирование цифровых устройств с использованием ПЛИС: учеб. / А. Ю Попов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. – 80 с.
5. Ларченко, Л. В. Функціональне перетворення імпульсних потоків в апаратних обчислювачах математичних функцій / Л. В. Ларченко, Е. М. Кулак, Б. Д. Ларченко. // Радіоелектроніка та інформатика. – 2019. – №3. – С. 27–34.
6. Ларченко, Л. В. Специализированный вычислитель для извлечения корня квадратного из суммы квадратов. / Л. В. Ларченко, А. В. Хаханова // Радиоэлектроника и информатика. – 2010. – № 1(48). – С.71–74.
7. Комлев, О. С. Вибір апаратних засобів при побудові обчислювачів цифрової обробки сигналів. / О. С. Комлев, В. А. Логинов // Московський енергетичний інститут – 2006. – ВС/NW, №1 (8): 3.2.
8. Специализированный модуль для воспроизведения степенных функций : тез. докл. науч.-практ. конф. (2019) // 23-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті / 2019. – С.51–52

9. Shkil, A. S. Bit-Stream Power Function Online Computer / A. S. Shkil, L. V. Larchenko, B. D. Larchenko. // [18 IEEE East-West Design & Test Symposium \(EWDTS\)](#). – 2020. – С.423–428.
10. Матвійків, О. М. Інженерне проектування складних об'єктів і систем. Навчальний посібник. / О. М Матвійків, С. Ткаченко, В. И. Хаханов Національний університет "Львівська політехніка". – Л., 2016 – 261 с.
11. Автоматизоване проектування спеціалізованого біт-поточкового обчислювача степеневих функцій на платформі FPGA : тез. докл. науч.-практ. конф. (2020) // 24-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті / 2020. – С.26–27.