

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ІВАНОФРАНКІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
НАФТИ І ГАЗУ

На правах рукопису

Піх Володимир Ярославович

УДК 681.325

ПРОЦЕСОРИ СПЕКТРАЛЬНОГО КОСИНУСНОГО ПЕРЕТВОРЕННЯ  
ФУР'Є НА ОСНОВІ РІЗНИХ КОРЕЛЯЦІЙНИХ ФУНКЦІЙ ТА  
ТЕОРЕТИКО-ЧИСЛОВИХ БАЗИСІВ

Спеціальність 05.13.05 – Комп'ютерні системи та компоненти

Дисертація на здобуття наукового ступеня кандидата  
технічних наук

Ідентифікую всі  
приклади дисертації  
Засвідчую:

Вчений секретар  
спеціалізованої вченої  
ради К58.082.02

Науковий керівник  
доктор технічних наук, професор  
Николайчук Ярослав Миколайович



Івано-Франківськ 2016 р.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	5
ВСТУП.....	6

## РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ ТА СИСТЕМНІ ХАРАКТЕРИСТИКИ  
СПЕЦПРОЦЕСОРІВ СПЕКТРАЛЬНОГО ОПРАЦЮВАННЯ ЦИФРОВИХ  
ДАНИХ

1.1. Аналіз сфери застосування цифрових спектроаналізаторів в інформаційних системах.....	15
1.2. Теоретичні засади спектрального аналізу на основі інтегрального та дискретного перетворення Фур'є.....	21
1.2.1 Фундаментальна теорія косинусного перетворення Фур'є..	22
1.2.2 Дискретне перетворення Фур'є.....	24
1.3 Структура кореляторів як компонентів спектральних спецпроцесорів косинусного перетворення Фур'є .....	28
1.4 Дискретні кореляційні функції, їх аналітика, та системні властивості.....	38
1.5 Теоретико-числові базиси та перспективи вдосконалення кореляційних спецпроцесорів на їх основі, постановка задачі досліджень.....	40
ВИСНОВКИ ПО ПЕРШОМУ РОЗДІЛІ.....	46

## РОЗДІЛ 2

РОЗРОБКА МЕТОДУ ТА ТЕОРЕТИЧНИХ ЗАСАД ПОБУДОВА І  
КОМПОНЕНТІВ СПЕЦПРОЦЕСОРА ДИСКРЕТНОГО КОСИНУСНОГО  
ПЕРЕТВОРЕННЯ ФУРС

2.1. Теоретичні основи побудови та дослідження асимптотики дискретних автокореляційних функцій.....	47
---	----

2.2. Дослідження системних характеристик базових компонентів спектральних спецпроцесорів на основі теоретико-числового базису Хаара-Крестенсона .....	51
2.2.1 Дослідження апаратної та часової складності аналого-цифрового перетворювача у різних теоретико-числових базисах .....	51
2.2.2 Розробка методу побудови структури швидкодіючого аналого-цифрового перетворювача Хаара-Крестенсона на основі компараторів із парафазними виходами.....	58
2.2.3. Дослідження часової та апаратної складності реалізації багаторозрядних регістрів зсуву у базисах Радемахера та Хаара-Крестенсона .....	68
2.3. Структура та системні характеристики операційних пристроїв модульного матричного перемноження кодів у різних теоретико-числових базисах .....	75
2.4. Реалізація та системні характеристики накопичуючих суматорів....	86
ВИСНОВКИ ПО ДРУГОМУ РОЗДІЛІ.....	93

### РОЗДІЛ 3

#### РОЗРОБКА ТА ДОСЛІДЖЕННЯ АЛГОРИТМІВ РЕАЛІЗАЦІЇ ДИСКРЕТНОГО КОСИНУСНОГО ПЕРЕТВОРЕННЯ ФУР'Є НА ОСНОВІ РІЗНИХ АВТОКОРЕЛЯЦІЙНИХ ФУНКЦІЙ

3.1. Дослідження формалізованого алгоритму обчислення спектрів на основі центрованої нормованої автокореляції. ....	94
3.2 Розробка та дослідження формалізованого алгоритму обчислення спектрів на основі кореляційної функції автоковаріації з адаптацією характеристик базисних функцій різних теоретико-числових базисів..	95
3.3 Розрахунок системи взаємопростих модулів перетворення системи залишкових класів в базисі Крестенсона.....	98
3.4. Розробка функціональної схеми спецпроцесора спектрального перетворення сигналів у базисі Хаара-Крестенсона.....	108

ВИСНОВКИ ПО ТРЕТЬОМУ РОЗДІЛІ.....	111
РОЗДІЛ 4	
РОЗРОБКА СХЕМОТЕХНІЧНИХ РІШЕНЬ ТА РЕАЛІЗАЦІЯ СПЕКТРАЛЬНИХ СПЕЦ ПРОЦЕСОРІВ	
4.1. Розробка структури та мікроелектронних компонентів швидкодіючого аналого-цифрового перетворювача у базисі Хаара-Крестенсона.....	112
4.2 Розробка та оцінка системних характеристик швидкодіючого спецпроцесора обчислення спектру у різних теоретико-числових базисах .....	114
4.3. Розробка програмного продукту для розрахунку наборів взаємопростих модулів спектрального спецпроцесора різної розрядності .....	118
4.4. Розробка програмного забезпечення синтезу мікроелектронної реалізації компонентів спектрального спецпроцесора на програмованих логічних інтегральних схемах .....	132
ВИСНОВКИ ПО ЧЕТВЕРТОМУ РОЗДІЛІ.....	138
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ.....	140
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	143
ДОДАТК А Лістинг програми розрахунку модулів та їх параметрів для спецпроцесорів різної розрядності .....	159
ДОДАТК Б Лістинг коду програмованого спектрального спецпроцесора....	163
ДОДАТОК В Акти про впровадження дисертаційної роботи	167

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЦП – аналого-цифровий перетворювач

ЦАП – цифро-аналоговий перетворювач

АКФ – автокореляційна функція

ЦКП – цифровий кореляційний процесор

БРЗ – багаторозрядний регістр зсуву

К – компаратор

КСП – кореляційний спецпроцесор

КФ – кореляційна функція

ПК – персональний комп'ютер

СЗК – система залишкових класів

СП – спеціалізований процесор

ТЧБ – теоретико-числовий базис

ПЛІС – програмовані логічні інтегральні схеми

СКПФ- спектральне косинусне перетворення Фур'є

ДКП – дискретне косинусне перетворення

ШПФ – швидке перетворення Фур'є

ДКПФ – дискретне косинусне перетворення Фур'є

АОФ - адаптовані ортогональні функції

СКС – спеціалізовані комп'ютерні системи

## ВСТУП

Сучасні досягнення в області мікроелектроніки, мікропроцесорної техніки та розвитку фундаментальних досліджень у теорії та розробці алгоритмів опрацювання інформаційних потоків у різних теоретико-числових базисах (ТЧБ) [1 – 5] створюють сприятливі умови для відповідного вдосконалення та покращення системних характеристик високопродуктивних кореляційних та спектральних процесорів[5].

При цьому успішно вирішуються задачі підвищення ефективності формування, передавання та опрацювання інформації в розподілених та спеціалізованих комп'ютерних системах [6].

Важливою науковою задачею при створенні та впровадженні високопродуктивних спектральних процесорів є проблемна орієнтація їх системних характеристик на виконання арифметико-логічних та модульних операцій при цифровому опрацюванні сигналів у комп'ютерних мережах. Актуальним також є підвищення швидкодії рішення задач цифрової томографії, опрацювання оптичних голограм, побудови цифрових фільтрів, моніторингу спектральних характеристик маніпульованих сигналів у системах передавання даних, а також діагностування вібрацій та спектральних характеристик промислового обладнання нафтогазової та інших галузей промисловості.

У багатьох українських та зарубіжних наукових школах даний клас наукових та прикладних задач традиційно вирішується на основі двійкової системи числення ТЧБ Радемахера[9 – 12], що в значній мірі обмежує функціональні можливості процесорів при зростанні алгоритмічної складності обчислювальних задач та підвищенні швидкодії опрацювання інформаційних потоків та сигналів.

Успішне вирішення названого класу задач може бути досягнуто за рахунок розробки математичного апарату опрацювання інформації та створення відповідної архітектури процесорів та алгоритмів опрацювання цифрових даних на основі ТЧБ Хаара, Радемахера, Крестенсона, Уолша та Галуа. [1 – 3].

Успішне застосування базису Крестенсона та системи числення залишкових класів досягнуто при реалізації високопродуктивних процесорів у проблемно-орієнтованих комп'ютерних системах [15], які були реалізовані та впроваджені в системах реального часу на об'єктах нафтогазової, атомної, енергетичної та інших галузях промисловості.

Досвід ефективного застосування базису Крестенсона при рішенні широкого класу задач цифрового опрацювання сигналів та цифрових даних обґрунтовує значну актуальність наукової задачі створення високопродуктивних процесорів у названому базисі. Цьому сприяють сучасні можливості технологій їх проектування на основі програмованих логічних матриць (ПЛМ), а також на мікропроцесорних платформах відомих фірм Analog Device, Dallas Semiconductor, MAXIM, Altera, Xilinx Intel, Texas Instruments, Motorola та інших [13, 14, 16, 17].

Перспективним підходом вдосконалення методів побудови високопродуктивних процесорів є використання системи числення ТЧБ Крестенсона [18], що базується на використанні цілочисельної форми системи залишкових класів (СЗК).

Основоположниками теорії спектрального аналізу сигналів є відомі зарубіжні вчені: Л. Ейлер, Ш. Фур'є, О. Хінчин, М. Батт, Л. А. Залманзон, Н. Ахмад, К. Рао та інші. Серед українських вчених вагомий вклад у розвиток теорії та методів швидкого спектрального перетворення внесли: А. О. Мельник, Я.М. Николайчук, у тому числі у радіотехнічних системах Пастух О. А. атакож на основі оптико-електронних перетворювачів В.П. Кожемяко та інші. [26-31].

### **Актуальність теми**

Розвиток теорії універсальних та спеціалізованих процесорів тісно пов'язаний з відповідним розвитком двійкової системи числення, тобто теоретико-числового базису Радемахера [31].

Зростаючі жорсткі вимоги до швидкодії процесорів стимулювали дослідження у застосуванні інших ТЧБ відмінних від базису Радемахера. Наприклад, відомі успішні застосування базису Крестенсона, математичною

основою якого є розширені поля Галуа, для побудови високопродуктивних спецпроцесорів системи залишкових класів [5, 26, 30], а також базису Уолша, який використовується при створенні комунікаційних та сигнальних процесорів у комп'ютерних мережах [2, 36]. Однак, в даних дослідженнях мало уваги приділено вдосконаленню теорії та алгоритмів реалізації спектральних перетворень при опрацюванні сигналів, що особливо актуально при створенні високопродуктивних спектральних спецпроцесорів.

Таким чином, розробка високопродуктивних процесорів спектрального опрацювання на основі ТЧБ Крестенсона є актуальною науковою задачею, яка дозволяє вирішити завдання покращення системних характеристик обчислювальних засобів, як компонентів сучасних розподілених систем, підвищення ефективності передавання та захисту інформації в комп'ютерних мережах, а також побудови швидкодіючих спецпроцесорів кореляційного, спектрального та інших застосувань опрацювання інформації.

### **Зв'язок роботи з науковими програмами, планами і темами.**

Представлені в дисертаційній роботі результати досліджень виконані в рамках плану наукових досліджень Інституту інформаційних технологій та кафедри комп'ютерних систем та мереж Івано-Франківського національного технічного університету нафти і газу: «Синтез комп'ютерних систем і мереж для об'єктів нафтогазового комплексу і контролю за навколишнім середовищем» (номер державної реєстрації № 31202005B76A); «Синтез комп'ютерних систем та розробка програмного забезпечення для об'єктів нафтогазового комплексу» (номер державної реєстрації № 0111U005890); а також кафедри спеціалізованих комп'ютерних систем Тернопільського національного економічного університету: НДР «Розробка теоретичних засад та алгоритмів ідентифікації запусків потужних електроприводів у високовольтних ЛЕП на основі теорії розпізнавання образів» (номер державної реєстрації 0115U002340).

Усі вказані дослідження проводились за безпосередньою участю автора.



**Мета і завдання дослідження.** Метою роботи є вдосконалення методів цифрового опрацювання сигналів та розробка високопродуктивних спецпроцесорів спектрального перетворення Фур'є у базисі Хаара-Крестенсона.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

1) проаналізувати існуючий стан побудови та застосування спектральних спецпроцесорів в інформаційних системах;

2) систематизувати структури, дослідити асимптотику та системні характеристики дискретних автокореляційних функцій, як базових компонентів процесорів косинусного перетворення Фур'є;

3) провести аналіз характеристик кодових матриць ТЧБ: унітарного, Хаара, Радемахера та Крестенсона;

4) розробити метод швидкодіючого аналого-цифрового перетворення сигналів у ТЧБ Хаара-Крестенсона;

5) розробити метод дискретного косинусного перетворення Фур'є на основі адаптації характеристик ортогональних базисних функцій до експериментальної реалізації коваріаційної функції автокореляції;

6) розробити метод генерування ортогональних функцій різних ТЧБ, адаптованих до характеристик функцій кореляції;

7) розробити структуру та мікроелектронні компоненти швидкодіючого АЦП у базисі Хаара-Крестенсона;

8) розробити структуру швидкодіючого спектрального спецпроцесора з розширеними функціональними можливостями у базисі Хаара-Крестенсона;

9) розробити програмне забезпечення для розрахунку наборів взаємопростих модулів спектральних спецпроцесорів у базисі Хаара-Крестенсона різної розрядності та синтезу компонентів на ПЛІС.

**Об'єкт дослідження** – процеси цифрового опрацювання сигналів спеціалізованими процесорами спектрального перетворення у базисі Хаара-Крестенсона.

**Предмет дослідження** – методи та засоби опрацювання сигналів на основі дискретних спектральних перетворень на основі різних кореляційних функцій та теоретико-числових базисів.

Методи дослідження базуються на використанні теорії інформації, теорії чисел та кодування сигналів, теорії комп'ютерної логіки та теорії опрацювання цифрових даних, комп'ютерного моделювання та схмотехнічного проектування мікроелектронних засобів на кристалі.

Наукова новизна одержаних результатів полягає в розвитку методів та програмно-апаратних засобів для цифрового спектрального опрацювання сигналів у системі числення залишкових класів теоретико-числового базису Хаара-Крестенсона.

Основні результати і положення, що виносяться на захист, спрямовані на створення нових методів та програмно-апаратних засобів спектрального опрацювання цифрових даних у базисі Хаара-Крестенсона.

### **1. Вперше розроблено:**

1.1 метод швидкодіючого аналого-цифрового перетворення сигналів у кодовій системі Хаара-Крестенсона, який, у порівнянні з відомими аналогами, дозволяє розпаралелити формування кодів залишків у базисі Хаара згідно набору взаємо-простих модулів і не потребує додаткових міжбазисних перетворень Радемахера-Крестенсона для високопродуктивної реалізації модульних операцій множення та додавання цифрових даних у базовому кореляційному модулі спектрального спецпроцесора;

1.2 метод дискретного косинусного перетворення Фур'є на основі адаптації характеристик ортогональних базисних функцій до експериментальної реалізації коваріаційної функції автокореляції що, у порівнянні з відомими методами, дозволило зменшити кількість та спростити складність обчислювальних операцій алгоритму спектрального опрацювання сигналів;

1.3 метод генерування ортогональних функцій різних теоретико-числових базисів, адаптованих до характеристик функцій автокореляції, що дозволило реалізувати їх зберігання табличним способом та вибірку згідно параметрів

дисперсії, математичного сподівання та ступеня затухання енергії функції автоковаріації, що дозволило вилучити одну з трьох операцій множення в алгоритмі спектрального перетворення Фур'є.

**2. Удосконалено** метод формування цифрових значень ортогональних базисних функцій перетворення Фур'є шляхом переставлення їх кодів у системі залишкових класів, що спростило алгоритм спектрального перетворення та підвищило його швидкодію.

**3. Отримав подальший розвиток** метод обчислення значення спектру для однієї базисної функції шляхом розпаралелення виконання операцій множення діючих та затриманих у часі залишків цифрових відліків вхідного процесу у системі взаємо-простих модулів, множення отриманих залишків добутків на залишки відповідної ортогональної базисної функції та їх накопичуюче додавання на виході однієї точки спектру, що дозволило, у порівнянні з виконанням аналогічних операцій над двійковими кодами у базисі Радемахера, розпаралелити обчислювальні процеси та підвищити швидкодію визначення спектру.

#### **Практичне значення отриманих результатів.**

Розроблена структура швидкодуючого аналого-цифрового перетворювача (АЦП) сигналів на основі кодів Хаара-Крестенсона, який, у порівнянні з відомими аналогами, забезпечує відповідне зменшення часової складності у 2 рази, а апаратної складності у 1.5 рази по відношенню до АЦП паралельного типу у базисі Радемахера.

Розроблена структурна схема спецпроцесора для обчислення спектрального косинусного перетворення Фур'є в системі залишкових класів, який відрізняється від відомих підвищеною на 1-2 порядки швидкодією.

Реалізована та впроваджена у промисловість схемотехнічна структура високопродуктивного спецпроцесора спектрального опрацювання даних, орієнтована на мікроелектронне виконання на кристалі.

Практичні результати роботи використано та впроваджено:

1) у Тернопільському конструкторському бюро радіозв'язку «СТРІЛА» при реалізації програмно-апаратних засобів моніторингу спектральних характеристик

маніпульованих сигналів у спеціалізованих низових комп'ютерних мережах (акт від 30.11.2015р.);

3) на кафедрі програмного забезпечення автоматизованих систем Інституту інформаційних технологій Івано-Франківського національного технічного університету нафти і газу при викладанні дисципліни «Алгоритми та структура даних» для студентів спеціальності 6.050103 - «Програмна інженерія» (акт від 02.12.2015р.);

4) на кафедрі спеціалізованих комп'ютерних систем Тернопільського національного економічного університету при виконанні НДР «Розробка теоретичних засад та алгоритмів ідентифікації запусків потужних електроприводів у високовольтних ЛЕП на основі теорії розпізнавання образів» державний реєстраційний номер 0115U002340 (акт від 05.12.2015р.).

**Особистий внесок здобувача.** Основний зміст роботи, наукові положення та результати сформульовано та вирішено автором самостійно. Особистий внесок здобувача полягає в аналізі сучасного стану рішення науково-технічної задачі, розробці основних ідей, методик досліджень, структурних, принципових та алгоритмічних рішень, організації експериментів, програмного моделювання функцій спектрального спецпроцесора косинусного перетворення Фур'є, а також у розробці необхідного для дослідження програмного забезпечення. У публікаціях, написаних у співавторстві, здобувачеві належить: [109] – обґрунтовано необхідність визначення спектральних характеристик об'єкту управління; [110] – запропоновано метод розрахунку спектру на основі обчислення коефіцієнту взаємкореляції з базисною функцією конкретного ТЧБ; [111] – вперше запропонований новий метод дискретного косинусного перетворення Фур'є на основі кодових матриць ТЧБ та аналітики різних кореляційних функцій; [112] – запропоновано функціональні структури процесорів спектрального перетворення Фур'є; [113] – запропоновано метод формалізації дискретного косинусного перетворення Фур'є у системі залишкових класів; [114] – розроблено набір аналітичних виразів косинусного перетворення Фур'є на основі різних кореляційних та базисних функцій; [115] – розраховані набори модулів системи

залишкових класів для спецпроцесорів базису Крестенсона; [116] – запропонований спосіб зберігання та формування кодів базисних ортогональних функцій у пам'яті; [117] – викладені аналітичні вирази та проаналізована характеристика структур кореляційних обчислювачів; [111] – систематизовані теоретико-числові бази систем ортогональних функцій, які використовуються для розрахунку спектрів сигналів; [116] – запропоновано функціональне розширення структури пристрою на основі генераторів кодів залишків ортогональних функцій у базисі Хаара-Крестенсона; [117] – викладені теоретичні основи спектрального косинусного перетворення Фур'є на основі оцінок різних кореляційних функцій та їх застосування для обчислення спектрів, обґрунтована реалізація такого класу процесорів на мікроелектронних кристалах.

**Основні результати дисертації доповідались та обговорювались на:** Міжнародній проблемно-науковій міжгалузевій конференції «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління». – Бучач-Східниця, 2010; XI-th International conference «The experience of designing and application of CAD systems in micro-electronics». – Lviv- Polyana, 2011, 2015; Міжнародній молодіжній математичній школі «Питання оптимізації обчислень» (ПОО-XXXVII) – Київ, 2011; Проблемно-науковій міжгалузевій конференції «Юриспруденція та проблеми інформаційного суспільства», (ЮПІС – 2011). - Яремча, 2011; Всеукраїнській науково-практичній конференції «Інформаційні технології в освіті, техніці та промисловості» Ів.-Франківськ, 2013; Міжнародній координаційній нараді ISCM-2014. – Тернопіль, 2014; Міжнародній конференції «Modern Problems of Radio Engineering, Telecommunications and Computer Science», TCSET - 2014. – Львів-Славське, 2014; VII-й Міжнародній школі-семінарі «Теорія прийняття рішень». – Ужгород, 2014; II-й Всеукраїнській науково-практичній конференції молодих учених і студентів - Івано-Франківськ, 2015.

**Публікації.** За матеріалами дисертації опубліковано 15 друкованих праць, серед яких 4 статті (з них 2 одноосібні), з них 4 у фахових наукових виданнях України, один патент України на корисну модель, співавтор однієї монографії на

англійській мові. 9 робіт опубліковано у збірниках матеріалів конференцій, в.т.ч. міжнародних, 2 публікації індексовані наукометричною базою Index Copernicus.

Структура та обсяг дисертації. Дисертація складається зі вступу, чотирьох розділів, висновків та додатків (170 с.). Основний зміст дисертаційної роботи викладений на 142 сторінках. Дисертація містить 65 рисунків, 29 таблиць та 157 посилання на джерела.



# РОЗДІЛ 1

## ТЕОРЕТИЧНІ ЗАСАДИ ТА СИСТЕМНІ ХАРАКТЕРИСТИКИ СПЕЦПРОЦЕСОРІВ СПЕКТРАЛЬНОГО ОПРАЦЮВАННЯ ЦИФРОВИХ ДАНИХ

### 1.1 Аналіз сфери застосування цифрових спектроаналізаторів в інформаційних системах

Аналіз світового досвіду розробки сучасного та практичного застосування спецпроцесорів спектральної обробки сигналів, які застосовуються в системах обробки інформації провідними фірмами: Ubiquiti, Rohde&Schwarz, РФ Explorer, ATEN, UNI-T, NWT (Табл. 1.1.) .

Таблиця 1.1 - Спектроаналізатори провідних фірм

Назва фірми та модель	Характеристика
<p>Компанія Rohde&amp;Schwarz пропонує спектроаналізатори.</p> 	<p>Зокрема - Advantest серії U37xx (моделі з діапазоном частот до 3, 8, 31.8 та 43 ГГц).</p> <p>РФ Explorer-3G Combo 15-2700 мГц Ручний цифровий аналізатор спектру ЖК-дисплей 15-2700 мГц 112 кГц-600 мГц 113*70*25 мм</p>
<p>ATEN</p> 	<p>AT5005 призначений для аналізу сигналу в лабораторних умовах у діапазоні частот 0.15 500 мГц</p>

Продовження таблиці 1.1.

<p style="text-align: center;"><b>Spectrum Analyzer</b></p> 	<p>Діапазон частот: 150 кГц 500 мГц (-3dB).</p> <p>Центральна частота <math>\pm 100</math> кГц</p>
<p style="text-align: center;">UNI-T</p>  <p style="text-align: center;"><b>UNI-T UTS1030</b></p>	<p>UTS1030 6.5 "жк-9 кГц ~ 2 ГГц</p> <p>Ручний аналізатор спектру</p>
<p style="text-align: center;"><b>NWT 4000</b></p> 	<p>Тип: NWT4000</p> <p>Діапазон частот: 138 мГц-4.4 ГГц (без перерви), Крок 1 кГц</p> <p>- Діапазон частот: 138 мГц-4.4 ГГц</p>
<p style="text-align: center;">H-45 Compact</p> 	<p>Спектроаналізатор</p> <p>+MPEG4+CI+DVBT2+Оптика SEL-CWDM</p>
<p style="text-align: center;"><b>AC-6400</b></p> 	<p>Память для зберігання спектрів - 1,2 Мб;</p> <p>Спектральних ліній : 400, 800, 1600, 3200;</p> <p>частотний діапазон 10- 2500 Гц;</p>



Продовження таблиці 1.1.

<p style="text-align: center;"><b>АКЭ-823</b></p> 	<p style="text-align: center;">Внутрішня пам'ять 15 Мб; Діапазон спектрального аналізу кратних частот 2 ... 1000; Разрядність АЦП складає 16- розрядів, 256 цифрових відліків за період частоти 50 Гц</p>
---	---

Перші аналізатори спектру поділяли сигнал на частотні смуги за допомогою набору аналогових фільтрів. Дисплей такого аналізатора (рис. 1.1 ) показує рівень сигналу в безлічі частотних смуг , відповідних фільтрам.

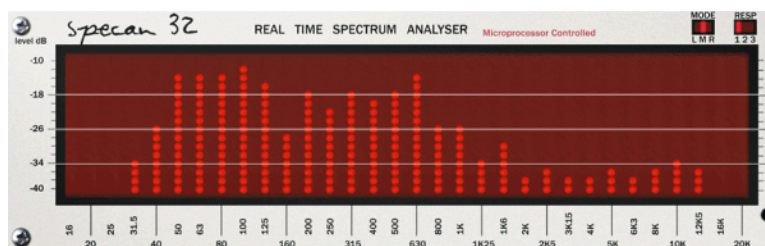


Рис . 1.1 - Трьохоктавний аналізатор Spresan32

Аналізатор Spresan32 емулює відомий прилад KlarkTeknik DN60. На рис. 1.2 наведено приклад частотних характеристик смугових фільтрів в аналізаторі, який відповідає стандарту ГОСТ 17168-82. Такий аналізатор називається трьохоктавний, оскільки в кожній октаві частотного діапазону є три смуги. Видно, що частотні характеристики смугових фільтрів перекриваються, їх крутизна залежить від порядку використовуваних фільтрів.

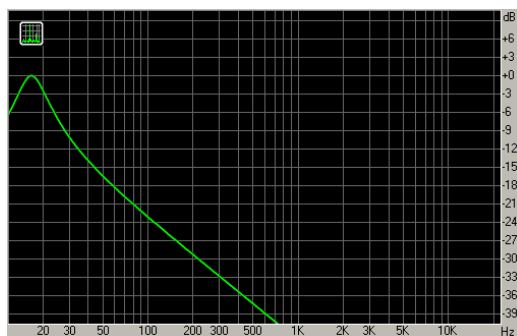


Рис. 1. 2. Частотні характеристики фільтрів трьохоктавного спектроаналізатора

Важливою властивістю спектроаналізатора є балістика - інерційність вимірників рівня в частотних смугах. Вона може регулюватися завданням швидкості наростання (атаки) і спаду рівня. Типовий час атаки і спаду в такому аналізаторі - порядку 200 і 1500 мс.

Смугові спектроаналізатори часто застосовуються для налаштування АЧХ (амплітудно-частотної характеристики) акустичних систем на концертних майданчиках. Якщо на вхід такого аналізатору подати рожевий шум (що має однакову потужність в кожній октаві), то дисплей покаже горизонтальну лінію, з можливою поправкою на варіацію шуму в часі. Якщо рожевий шум, проходячи через звукопідсилювальну систему залу, спотворився, то зміни його спектру будуть видні на аналізаторі. При цьому аналізатор, як і наше вухо, буде малочутливе до вузьких провалів АЧХ (менше 1/3 октави).

Аналізатор спектру Ubiquiti AirView2-EXT призначений для ефективного використання у радіодіапазоні 2400 МГц. Габарити пристрою не перевищують стандартні розміри флеш-накопичувачів, а підключення Ubiquiti AirView2-EXT до комп'ютера здійснюється за допомогою інтерфейсу USB.

Відмінною особливістю даної моделі є можливість підключення зовнішньої антени.

Програмне забезпечення Ubiquiti AirView2-EXT відображає графічну інформацію про заповнення радіоканалів, розподілу по частотах RF енергії, зібраної пристроєм з початку його роботи, а також динамічних енергетичних характеристиках частотного спектра 2400 - 2485 МГц.

FFT (fast Fourier transform) – алгоритм швидкого обчислення дискретного перетворення Фур'є. Завдяки чому стало можливим аналізувати спектр звукових сигналів в реальному часі.

Розглянемо роботу типового FFT-аналізатора. На вхід йому надходить цифровий аудіосигнал. Аналізатор вибирає з сигналу послідовні інтервали («вікна»), на яких буде обчислюватися спектр, і обчислює FFT в кожному вікні для отримання амплітудного спектра  $X_k$ . Обчислений спектр відображається у вигляді графіка залежності амплітуди від частоти (рис. 1.3). Аналогічно в смугових

аналізаторах, зазвичай використовується логарифмічний масштаб по осях частот і амплітуд. Але через лінійне розташування смуг FFT за частотою спектр може виглядати недостатньо детальним на нижніх частотах або надмірно осцилювати на верхніх частотах.

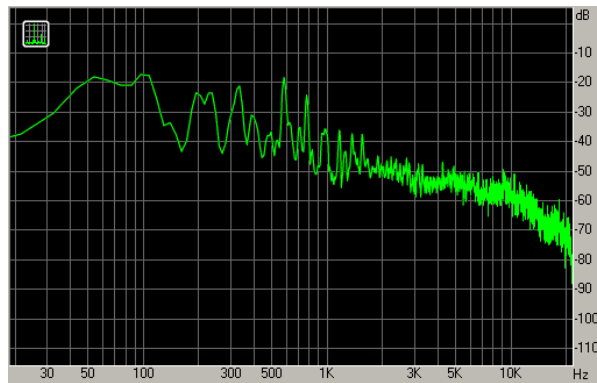


Рис. 1.3. Дисплей FFT-аналізатора

Якщо розглядати FFT як набір фільтрів, то, на відміну від смугових фільтрів трохоктавного аналізатора, фільтри FFT матимуть однакову ширину в герцах, а не в октавах. Тому рожевий шум на FFT-аналізаторі буде вже не горизонтальною лінією, а похилою, зі спадом 3 дБ / окт. Горизонтальною лінією на FFT-аналізаторі буде білий шум - він містить рівну енергію в рівних лінійних частотних інтервалах.

Параметр  $N$  - число аналізованих відліків сигналу - має вирішальне значення для виду спектру. Чим більше  $N$ , тим щільніше сітка частот, за якими FFT розкладає сигнал, і тим більше деталей по частоті видно на спектрі. Для досягнення більш високого частотного дозволу доводиться аналізувати більш довгі ділянки сигналу. Якщо сигнал в межах вікна FFT змінює свої властивості, то спектр буде відображати деяку усереднену інформацію про сигнал з усього інтервалу вікна.

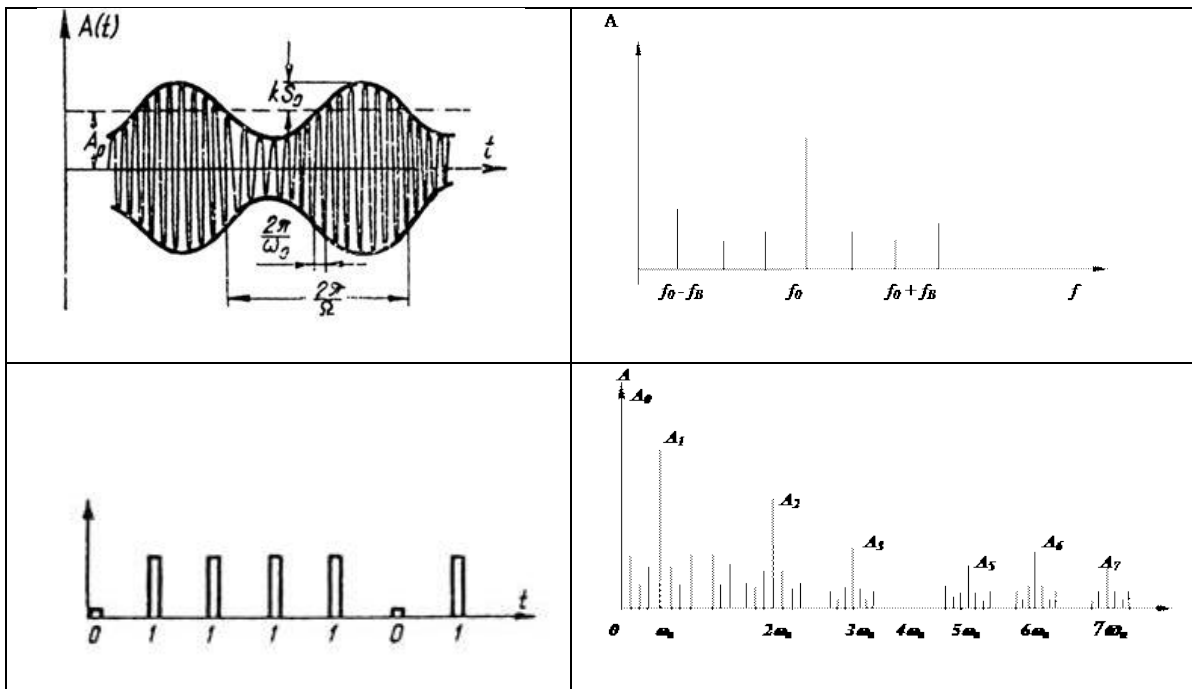
Коли потрібно проаналізувати швидкі зміни в сигналі, довжину вікна  $N$  вибирають укороченою. У цьому випадку дозвіл аналізу за часом збільшується, а по частоті - зменшується. Таким чином, дозвіл аналізу по частоті обернено пропорційний вирішенню за часом.

Задача спектрального аналізу вирішується в системі передаванні даних для формування модульованих та маніпульованих сигналів заданого спектру. Існують наступні методи модуляції сигналів:

- Амплітудна модуляція
- Дельта-модуляція
- Кодово - імпульсна модуляція (КІМ)
- Кутова швидкісна модуляція
- Часово - імпульсна модуляція
- Широтно - імпульсна модуляція (ШІМ)

У табл. 1.2 показано приклади формування спектральних характеристик маніпульованих сигналів.

Таблиця 1.2 – Амплітудна, амплітудно-імпульсна модуляція



Спектральні аналізатори широко використовуються для діагностування стану технологічного обладнання, яке характеризується вібраціями зі змінним спектром. Прикладом такого спектрального моніторингу є діагностування стану електроприводів глибинних насосів у нафтогазовій галузі [9,10].

Важливою функцією застосування спектральних аналізаторів є контроль допустимих норм вібрацій та спектральних характеристик підшипників потужних компресорів магістральних газопроводів[42,69]. Широке застосування спектральні аналізатори отримують в авіаційній промисловості при діагностуванні реактивних турбін.[78] Відоме також ефективне застосування спектральних аналізаторів при дослідженні та вимірюванні характеристик рідинних потоків, пов'язаних з виявленням поривів та витоків в окремих ділянках продуктопроводів [47].

## 1.2 Теоретичні засади спектрального аналізу на основі інтегрального та дискретного перетворення Фур'є

Сучасні методи опрацювання сигналів базуються на різних теоретичних засадах оцінки параметрів гармонічних сигналів та різних способів їх перетворення. Аналіз періодичних сигналів найчастіше виконується на основі частотних та ортогональних перетворень. Широке застосування для рішення цього класу задач отримали дискретні перетворення в базисах ортогональних функцій Радемахера, Хаара, Уолша, Крестенсона та Галуа [28]. Серед названих перетворень найбільш широке практичне застосування отримало дискретне комплексне перетворення Фур'є [29]. Слід зауважити, що при аналізі неперіодичних сигналів вказані перетворення не є достатньо ефективними. Це зумовлено складністю аналітики та асимптотики алгоритмів, а також, як показано в роботі [30], зниження рівнів чутливості інформаційних параметрів.

Теорія спектрального аналізу викладена у фундаментальних роботах Н. Вінера, О.О. Харкевича, А. М. Колмогорова, В.К. Задіраки та інших вчених. У той же час реалізація програмно-апаратних засобів спектрального аналізу далека від досконалості і не повною мірою відповідає сучасним вимогам досягнення мінімальної апаратної та часової складності. Це пов'язано, у першу чергу, з великою обчислювальною складністю, реалізацією алгоритмів спектрального аналізу на основі відомого комплексного перетворення Фур'є та швидких перетворень Фур'є [32]. Використання косинусного перетворення Фур'є в практиці створення засобів спектрального аналізу набуло набагато меншого поширення у зв'язку з необхідністю послідовного обчислення автокореляційної та взаємкореляційної функцій, що потребує центрування та нормування вхідних сигналів, а також виконання множення та сумування добутків чисел з різними знаками. При цьому існуючий досвід побудови швидкодіючих процесорів та сучасної мікроелектронної елементної бази обумовлює актуальність створення процесорів косинусного перетворення Фур'є для спектрального аналізу та діагностування на його основі складних технологічних об'єктів.

### 1.2.1 Фундаментальна теорія косинусного перетворення Фур'є

Фундаментальна теорія косинусного перетворення Фур'є базується на основі інтегральних оцінок спектральних амплітуд [2]:

$$a(\omega) = \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt, \quad b(\omega) = \int_{-\infty}^{\infty} f(t) \sin(\omega t) dt, \quad (1.1)$$

де функція  $f(t)$  виражається через комплексну функцію  $F(j\omega)$ :

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) \exp(j\omega t) d\omega, \quad (1.2)$$

яка розраховується згідно виразу:

$$F(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} f(t) \exp(-j\omega t) dt \quad (1.3)$$

і має зміст спектральної густини.

Спектральна функція  $F(j\omega)$  виражається через парні  $a(\omega)$  і непарні  $b(\omega)$  гармонічні функції:

$$F(j\omega) = a(\omega) - jb(\omega), \quad (1.4)$$

звідки отримується вираз для функції спектрального перетворення:

$$f(t) = \frac{1}{\pi} \int_0^{\infty} [a(\omega) \cos(\omega t) - b(\omega) \sin(\omega t)] d\omega. \quad (1.5)$$

При чому, якщо функція  $f(t)$  є дійсною, то використання комплексної функції  $F(j\omega)$  є надлишковим і пряме та обернене перетворення Фур'є може здійснюватися лише над множиною дійсних чисел. Можливість реалізації косинусного перетворення Фур'є у вигляді операцій над дійсними "особливо цілими числами", що визначає перспективу його застосування для побудови високопродуктивних спецпроцесорів спектрального опрацювання сигналів у мікроелектронному виконанні.

У теорії спектрального аналізу випадкових процесів визначено, що кореляційна функція і спектральна щільність становлять пару перетворень Фур'є, яка відома під назвою теореми Вінера-Хінчина [18]. Неперервний випадок:

$$S_{xx}(f) = \int_{-\infty}^{\infty} r_{xx}(\tau) e^{-j2\pi f\tau} d\tau, \quad r_{xx}(\tau) = E[x(t)x^*(t-\tau)], \quad (1.6)$$

та дискретний випадок:

$$S_{xx}(f) = \sum_{k=-\infty}^{\infty} r_{xx}[k] e^{-j2\pi kf}, \quad r_{xx}[k] = E[x[n]x^*[n-k]], \quad (1.7)$$

де  $S_{xx}(f)$  – спектральна щільність потужності функції  $x(t)$  для неперервного випадку, а для дискретного – спектральна щільність потужності з дискретними значеннями  $x[n]$ . Слід зауважити, що автокореляційна функція  $r_{xx}$  визначена через математичне сподівання від добутку і що перетворення Фур'є від  $x(t)$  не існує в загальному випадку, так як стаціонарні випадкові функції не інтегровані в квадратичному просторі.

Формулу Вінера-Хінчина можна виразити через косинусне перетворення Фур'є. Легко показати, що коваріаційна функція і спектральна щільність симетричні:  $R_x(\tau) = R_x(-\tau)$  та  $R_x(s) = R_x(-s)$ .

Використовуючи формулу Ейлера, одержують [18]:

$$R_x(\omega) = \int_{-\infty}^{\infty} R_x(\tau) \cos \omega \tau d\tau = 2 \int_0^{\infty} R_x(\tau) \cos \omega \tau d\tau; \quad (1.8)$$

$$R_x(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R_x(\omega) \cos \omega \tau d\omega = \frac{1}{\pi} \int_0^{\infty} R_x(\omega) \cos \omega \tau d\omega. \quad (1.9)$$

Можна також визначити спектральні щільності для стаціонарних випадкових послідовностей (випадкових процесів з дискретним часом). Для скалярної випадкової величини в стаціонарному випадку можна визначити

$$R_x(nT) = \Phi_s(kT, iT) = R_x(k - iT) = E\{x(kT)x^*(iT)\}, \quad (1.10)$$

де  $x(iT)$  приймає тільки цілі значення. Дискретну спектральну щільність визначають як дискретне перетворення Фур'є або подвійне перетворення коваріаційної функції випадкової послідовності.

Таким чином, дискретний варіант теореми Вінера-Хінчина можна записати в наступному вигляді:

$$R_x(z) = T \sum_{n=-\infty}^{\infty} R_x(nT) z^{-n}, \quad (1.11)$$

де  $z^{-n}$  – одна з ортогональних функцій ТЧБ.

Ефективний спосіб розрахунку спектру базується на обчисленні коефіцієнту взаємкореляції між нормованою функцією автокореляції досліджуваного процесу і базисною функцією конкретного ТЧБ.

Значні функціональні обмеження обчислювальних процесів перетворення Фур'є та слабка збіжність рядів Фур'є призвели до розвитку та успішного застосування інших ТЧБ для обчислення спектрів сигналів. При цьому для вдосконалення характеристик спецпроцесорів необхідно розв'язати задачу сумісного застосування теорії кореляційних функцій та базисних функцій різних ТЧБ шляхом адаптації базисних функцій до асимптотики кореляційних функцій досліджуваних сигналів. Крім того, виклад теорії кореляційних функцій в джерелах інформації стосується майже виключно мультиплікативної форми автокореляційної функції, яка серед інших формул не характеризується найкращими показниками в плані реалізації кореляційних спецпроцесорів з мінімаксними характеристиками.

Способи кодування даних визначаються теоретико-числовими базисами, які застосовуються для їх представлення. Найбільш широко вживаними ТЧБ в сучасних комп'ютерних системах є ортогональні базиси Радемахера, Хаара, Крестенсона та Галуа, які породжують відповідні системи числення.

### 1.2.2 Дискретне перетворення Фур'є

Дискретне перетворення Фур'є (ДПФ) є базовим алгоритмом цифрової обробки сигналів у частотній області. Завдяки наявності ефективних алгоритмів його обчислення – алгоритмів швидкого перетворення Фур'є (ШПФ) – ДПФ широко використовується для цілей цифрової фільтрації та спектрально-кореляційного аналізу сигналів.



Для сигналу, заданого у вигляді дискретної послідовності  $S(n)$ , пряме й обернене дискретне перетворення Фур'є (ДПФ) має вигляд:

$$S(k) = \sum_{n=0}^{N-1} s(n) \exp\left[-j \frac{2\pi nk}{N}\right], k = \overline{0, N-1}; \quad (1.12)$$

$$S(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) \exp\left[j \frac{2\pi nk}{N}\right], n = \overline{0, N-1}; \quad (1.13)$$

де  $k$  – номер гармоніки із частотою  $\int k$ ,  $N$  – обсяг вибірки.  $S(k)$ , визначений як комплексний спектр сигналу, можна представити

$$S(k) = A(k) - jB(k) = C(k)e^{-j\varphi(k)}; \quad (1.14)$$

де амплітудно-частотна (АЧХ) і фазо-частотна характеристика (ФЧХ) сигналу відповідно визначаються:

$$C(k) = \sqrt{A^2(k) + B^2(k)}; \quad (1.15)$$

$$\varphi(k) = \arctg \frac{B(k)}{A(k)} + 2\pi m; \quad (1.16)$$

Обчислення перетворень Фур'є вимагає дуже великого числа множень (приблизно  $N^2$ ) і обчислень синусів. Існує спосіб виконати ці перетворення значно швидше: приблизно за  $N \cdot \log_2 N$  операцій множення. Цей спосіб називається швидким перетворенням Фур'є. Алгоритм ШПФ – це спосіб швидкого

обчислення ДПФ  $X(jk) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)kn}$ , що дозволяє усунути притаманну

ДПФ надлишковість. Вони ґрунтуються на властивостях комплексної експоненти  $e^{-j(2\pi/N)kn}$ , для зручності позначають  $W_N^{kn}$  ( $W_N^{kn} = e^{-j(2\pi/N)kn}$ ), її симетрії  $W_N^{(N-k)n} = W_N^{(N-k)n} = (W_N^{kn})^*$  і періодичності  $W_N^{(N-k)(N+n)} = W_N^{kn}$  з періодом, рівним довжині оброблюваної реалізації сигналу  $N$  (числу точок ШПФ). Відповідно до останньої властивості експоненти  $W_N^{pkn} = W_{N/p}^{kn}$  відповідає період  $N/p$ , де  $p$  – цілі

числа, на які ділиться  $N$ . Використання даних властивостей в алгоритмах ШПФ виключає велике число повторюваних при обчисленні ДПФ операцій.

Дискретне перетворення Фур'є (ДПФ) є базовим алгоритмом цифрової обробки сигналів в частотній області. Завдяки наявності ефективних алгоритмів швидкого перетворення Фур'є (ШПФ) шляхом реалізації дискретного перетворення Фур'є (ДПФ) ці перетворення широко використовуються для цілей цифрової фільтрації і спектрально-кореляційного аналізу сигналів.

Для сигналу, заданого у вигляді дискретної послідовності  $S(n)$ , пряме і зворотне ДПФ має вигляд [2]:

$$S(k) = \sum_{n=0}^{N-1} s(n) \exp\left[-j \frac{2\pi nk}{N}\right]; k = \overline{0, N-1}; \quad (1.17)$$

$$S(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) \exp\left[j \frac{2\pi nk}{N}\right]; n = \overline{0, N-1},$$

де  $k$  – номер гармоніки з частотою  $f_k$ ,  $N$  – об'єм вибірки.

Функція  $S(k)$  визначена як комплексний спектр сигналу, можна подати

$$S(k) = A(k) - jB(k) = C(k)e^{-j\varphi(k)},$$

де амплітудно-частотна (АЧХ) і фазо-частотна характеристика (ФЧХ) сигналу відповідно визначається згідно виразів:

$$C(k) = \sqrt{A^2(k) + B^2(k)}; \quad \varphi(k) = \arctg \frac{B(k)}{A(k)} + 2\pi m.$$

Обчислення перетворення Фур'є потребують  $N^2$  множень и обчислення синусів. Існує спосіб виконати ці перетворення значно швидше: за  $N \cdot \log_2 N$  операцій множення згідно алгоритму ШПФ: [2]

$$X(jk) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)kn}.$$

При цьому в алгоритмах обчислення ШПФ використовують симетрію функції  $W_N^{(N-k)n} = W_N^{(N-k)n} = (W_N^{kn})$ , а також її періодичність  $W_N^{(N+k)(N+n)} = W_N^{kn}$  з періодом, який рівний довжині опрацьованої реалізації сигналів, де  $N$  число точок

ШПФ. Причому властивості експоненти  $W_N^{pkn} = W_{N/p}^{kn}$  відповідає періодам  $N/p$ , де  $p$  - цілі числа, на які ділиться  $N$ .

ШПФ може виконуватися на основі різних алгоритмів Кулі-Тьюкі, Рейдера-Бренера та ін. Ці алгоритми об'єднуються на тому що комплексні фазові множники  $W_N^r$  замінюють дійсними або уявними, що приводить до декомпозиції алгоритму і спрощує обчислення. В табл. 1 приведені дані кількості операцій множення  $\alpha_m$  та додавання  $\beta_m$  для алгоритмів ШПФ<sub>2t-f</sub> і ШПФ<sub>4t-f</sub> обчисленні згідно виразу 1.18 і табл. 1.3 [94]:

$$\begin{aligned} a_m &= 2^m(2m-7)+12; \beta_m = 3 \cdot 2^m(m-1)+4; \\ a_m &= 2^{m-1}(3m-10)+8; \beta_m = 2^{m-1}(7m-10)+8; \\ a_m &= 2^{m-1}(3m-10)+8; \beta_m = 2^{m-2}(7m-26/3)+8/3; \\ a_m &= 2^{m-3}(9m-86/3)+16/3; \beta_m = 2^{m-3}(25m-86/3)+16/3. \end{aligned} \quad (1.18)$$

Таблиця 1.3 Системні характеристики ШПФ

Безпосереднє повертання вектора					
№ п/п	$N = 2^m$	ШПФ <sub>2t-f</sub>		ШПФ <sub>2t-f</sub>	
		$\alpha_m$	$\beta_m$	$\alpha_m$	$\beta_m$
1	16	28	148	24	144
2	64	332	964	264	920
3	256	2316	5380	1800	5080
4	1024	13324	27652	10248	25944
Ефективний поворот вектора					
№ п/п	$N = 2^m$	ШПФ <sub>2t-f</sub>		ШПФ <sub>2t-f</sub>	
		$\alpha_m$	$\beta_m$	$\alpha_m$	$\beta_m$
1	16	24	152	20	148
2	64	264	1032	208	976

Продовження таблиці 1.3

3	256	1800	5896	1392	5488
4	1024	10248	30728	7856	28336

Результати представлені в табл. 1.3 дозволяють оцінити граничну швидкодію процесора ШПФ у базисі Радемахера, наприклад при об'ємі вибірки  $N = 2^{10}$ , числі точок спектра 32 і розрядності процесора 16 біт. Швидкодія виконання операцій множення двійкових чисел  $\alpha_m \nu$  на основі матричної схеми Брауна складає  $16 \times 5 \times 2 = 160$  мікротактів, а операцій додавання  $16 \times 5 = 80$  мікро тактів  $= \alpha_m \nu$ , де  $\nu$  швидкодія переключення мікроелектронного вентиля. Таким чином число мікрооперацій розрахунку ШПФ на 32 точки складає:  $(13324 \times 160 + 27652 \times 80) \times 32 = 13901824 \nu$ . При тактовій частоті генератора процесора 1 ГГц швидкодія виконання 100 обчислень спектра згідно ШПФ рівна  $\approx 60$  с., що недопустимо для багатьох промислових об'єктів при розрахунку спектральних характеристик в режимі онлайн.

### 1.3 Структура кореляторів як компонентів спектральних спецпроцесорів косинусного перетворення Фур'є

Значний внесок у створенні розвитку теорії кореляційного аналізу сигналів на основі різних кореляційних функцій зробили наступні відомі вчені R. Kenefic, J. Varchak, A. Bell, D. DeFatta [7-9], Е.С. Вентцель, Ф.Г. Ланге, Г.Я. Мірський, Е.І. Цветков, М.Г. Серебряніков, А.А. Первозванський, Я.М. Николайчук [10-15]. У Роботах названих вчених розроблені теоретичні засади алгоритмів обчислення кореляційних функцій, характеристики впливу частоти дискретизації та квантування на точність обчислень а також застосування кореляційного аналізу у різних сферах. Недостатньо глибоко були досліджені характеристики цифрових процесорів, які реалізували визначення кореляційних функцій на основі різних аналітичних виразів, що стримувало можливості широкого застосування

косинусного перетворення Фур'є для спектрального аналізу та розпізнавання образів.

Наявність операцій обчислення автокореляційних та взаємокореляційних функцій у косинусному перетворенні Фур'є визначає необхідність аналізу стану розробки цифрових кореляторів на сьогоднішній час. У роботах [16-18] викладена систематизація аналітики та асимптотики різних алгоритмів обчислення автокореляційних та взаємокореляційних функцій наступного типу: знакова, релейна, коваріаційна, кореляційна, структурна, модульна та еквівалентності, а також базові структури відповідних спецпроцесорів. Особливістю таких спецпроцесорів є використання різних теоретико-числових базисів (ТЧБ) для представлення числових даних [27-31], а також глибокий рівень розпаралелення обчислювальних операцій, що приводить до регулярності структури та спрощення їх мікроелектронної реалізації.

Сучасні досягнення у створенні процесорів реалізації кореляційних функцій висвітлені в патентних матеріалах [19-22]. Аналіз архітектури цифрових кореляторів та їх мінімакських системних характеристик стосовно часової та апаратної складності створює основу їх ефективного застосування шляхом розширення функціональних можливостей для спектрального аналізу та реалізації процесорів дискретного косинусного перетворення Фур'є.

В табл. 2 приведені структури кореляторів, які є компонентами спецпроцесорів спектрального косинусного перетворення Фур'є.

Аналіз методів для опрацювання сигналів та розробок різних пристроїв а також різних пристроїв для аналізу технологічних процесів показує, що цифрові кореляційні процесори знаходять широке застосування із жорсткими вимоги до системних характеристик, таких як:

- максимальна швидкодія;
- мінімальна апаратна, часова та алгоритмічна складність;
- мінімальні габарити;
- висока інформативність.

Тому задача розробки нових методів побудови та вдосконалення спецпроцесорів кореляційного опрацювання сигналів, та розширення функціональних можливостей є актуальною науково-технічною задачею.

Для досягнення та забезпечення необхідних характеристик кореляторів інтенсивно розвивалися різні структурні варіанти їх побудови, які забезпечували досягнення необхідних системних характеристик. Наприклад: максимальна швидкодія, мінімізована апаратна складність, регулярність архітектури, виключення операцій множення в алгоритмах обчислення кореляційних функцій та інше. Тому при аналізі та вдосконаленні характеристик кореляційних процесорів доцільно виконати аналіз відомих вітчизняних та зарубіжних патентно-захищених розробок.

Удну з перших структур цифрового багатоканального корелятора розроблено співробітником Інституту автоматики та електрометрії Сибірського відділення АН СРСР В.І. Юшиним, де з метою підвищення швидкодії шляхом запам'ятовування дискретів сигналів запропоновано комбінований спосіб затримки даних і їх переміщення в регістрі зсуву. Корелятор має два аналого-цифрові перетворювачі, синхронізатор, регістр зсуву, два комутатори інформаційних потоків, блок перемноження і накоплюючі суматори, кількість яких рівна числу точок обчислюваної кореляційної функції.

В той же час операція множення виконується послідовно для кожного зсунутого цифрового значення вхідних даних. Наявність одного блоку перемноження у структурі описаного корелятора суттєво обмежує його швидкодію.

У 80-х роках цифрові корелятори та їх сімейство в мікроелектронному виконанні розроблено під керівництвом професора Я.М. Николайчука [44, 64-66]. Особливості даних кореляторів полягали у використанні унітарного теоретико-числового базису, що призвело до регулярності архітектури такого класу процесорів, дозволило забезпечити високий рівень паралелізму операцій та реалізацію операцій множення в кожному каналі на основі простих логічних схем.

Широкого значення для обчислення кореляційних функцій спецпроцесорами набули архітектури у базисі Радемахера на основі паралельної базової архітектури [14], яка у формалізованому вигляді приведена на рис. 1.4 [92].

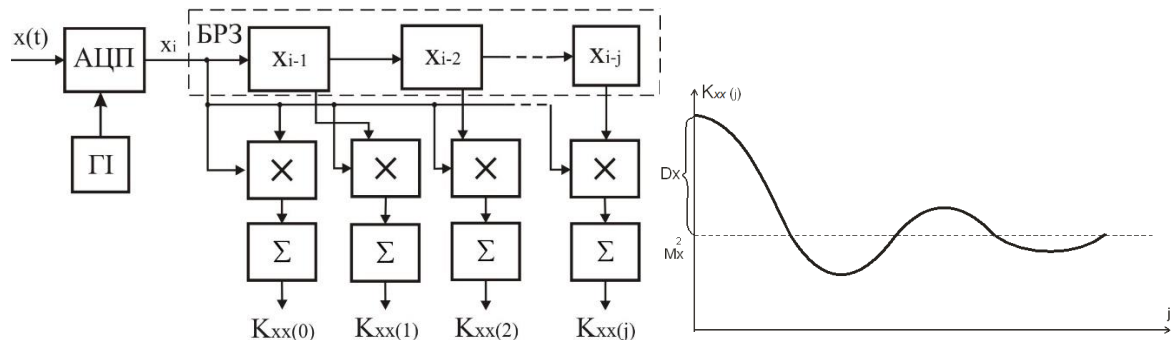


Рисунок 1.4 - Формалізована архітектура цифрового корелятора

Кореляційний спецпроцесор з даною архітектурою, який на виході АЦП формує паралельні коди цифрових відліків у двійковій системі числення базису Радемахера реалізує алгоритм обчислення автоковаріаційної функції [84, 88]:

$$K_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot x_{i-j}, \quad j = 0, 1, 2, \dots, \quad (1.19)$$

де  $D_x = \frac{1}{n-1} \sum (x_i - M_x)^2$  - дисперсія;  $M_x = \frac{1}{n} \sum_{i=1}^n x_i$  - математичне сподівання.

На рис. 1.4 представлена архітектура корелятора, що включає:

- $x(t)$  – вхідний аналоговий сигнал;
- $x_i$  – квантовані з інтервалом  $\delta = \text{const}$  у діапазоні  $0 \leq X_i \leq A$  та дискретизовані з інтервалом  $\Delta t$  ( $\text{const}$ ) в часі цифрові відліки представлені у базисі Радемахера;
- $x_{i-1}, x_{i-2}, \dots, x_{i-j}$  – цифрові відліки із затримкою в часі у регістрі зсуву  $G_j$ ;
- $\times$  – матричний перемножувач;
- $\Sigma$  - накопичуючий суматор.

Асимптотика кореляційної функції  $K_{xx}(j)$  відповідає аналітичним виразам [33]:

$$K_{xx}(0) = D_x + M_x^2; \quad K_{xx}(\infty) = M_x^2. \quad (1.20)$$

Формалізована архітектура цифрового корелятора, яка представлена на рис. 1.4 містить наступні компоненти:

- 1 – АЦП, який формує цифрові коди  $x_i$  у різних ТЧБ;
- 2 – багаторозрядний регістр зсуву (БРЗ) запам'ятовує потік цифрових відліків  $x_i, x_{i-1}, x_{i-2}, \dots, x_{i-j}$ , які подаються на входи відповідних перемножувачів ( $\times$ );
- 3 – цифровий перемножувач, формує парні добутки  $x_i \cdot x_{i-j}$ , коди яких додаються в накопичуючих суматорах ( $\Sigma$ );
- 4 – накопичуючий суматор ( $\Sigma$ ) виконує додавання парних добутків  $x_i \cdot x_{i+j}$  та операцію ділення шляхом відкидання молодших розрядів отриманого двійкового коду.

Така архітектура корелятора характеризується високою швидкістю АЦП паралельного типу, глибоким розпаралеленням виконання операцій та високою регулярністю архітектур. Крім того автоковаріаційна функція  $K_{xx}(j)$  характеризується високою інформативністю оскільки її асимптотика дозволяє визначити математичне сподівання  $M_x$  та дисперсії  $D_x$  згідно виразів :

$$M_x = \sqrt{K_{xx}(\infty)}; D_x = K_{xx}(0) - K_{xx}(\infty). \quad (1.21)$$

Результати обчислення  $K_{xx}(j)$  представлені в квадратичному просторі, що забезпечує менший об'єм вибірки ( $n > 256$ ) цифрових даних по відношенню до інших функцій кореляції.

Реалізація кореляційного процесора у базисі Радемахера характеризується наступними недоліками:

- значна апаратна складність АЦП, в якому використовується дешифратор унітарного коду в двійковий код ТЧБ Радемахера;
- наявність наскрізних переносів при виконанні арифметичних операцій у двійковій системі числення у блоках перемножувачів та накопичуючих суматорів;
- відсутність можливості реалізації спектрального косинусного перетворення Фур'є на основі коваріаційної функції  $K_{xx}(j)$  оскільки дана функція характеризується нецентрованістю та ненормованістю;



Аналогічно описуються архітектурні особливості та характеристики кореляційних спецпроцесорів з використанням математичних виразів та різних кореляційних функцій (табл. 1.3) та різних ТЧБ, що демонструє таблиця 1.4.

Таблиця 1.4 – Мультибазисні кореляційні спецпроцесори на основі класифікованих різних кореляційних функцій та ТЧБ

ТЧБ	Унітарний (U)	Хаара (H)	Радемахера (R)	Крестенсона(K)	
Дискретні автокореляційні функції	$H_{xx}(j)$	UH	HH	RH	CH
	$B_{xx}(j)$	UB	HB	RB	CB
	$K_{xx}(j)$	UK	HK	RK	CK
	$R_{xx}(j)$	UR	HR	RR	CR
	$\rho_{xx}(j)$	Up	Hp	Rp	Cp
	$C_{xx}(j)$	UC	HC	RC	CC
	$G_{xx}(j)$	UG	HG	RG	CG
	$F_{xx}(j)$	UF	HF	RF	CF

Існуючі спецпроцесори аналізованого класу приведені на рис. 1.3. Структура коваріаційного корелятора  $K_{xx}(j)$  у базисі Радемахера ідентифікується символом – RK (табл.1.4). Найбільш поширені структури мультибазисних кореляторів, які наведені в таб. 1.5. відповідно ідентифікуються наступними символами:

- UG унітарно-модульний (табл.1.4);
- UC унітарно-структурний (табл.1.4);
- HF Хаара-еквівалентний (табл.1.4);
- UG матрично-унітарно-модульний (табл.1.4).

Системні характеристики кореляційних спецпроцесорів описуються функціоналом у вигляді кортежу [98]:

$$- S_x = F(W, k, n, m, \Delta, V, M), \quad (1.22)$$

де  $W$  – тип кореляційної функції,  $k$  – розрядність АЦП,  $n$  – об'єм вибірки,  $m$  – число точок кореляційної функції,  $\Delta$  - точність обчислень,  $V$  – швидкодія,  $M$  – об'єм пам'яті.

У табл. 1.5 зображено структури цифрових кореляторів, які реалізують алгоритми обчислення структурної, модульної та еквівалентної функції автокореляції, а також матрично-модульної взаємкореляційної функції.

Таблиця 1.5 - Структура та аналітика цифрових кореляторів з використанням різних ТЧБ

№	Тип корелятора	Структура	Аналітичне представлення
1	2	3	4
1.	Пристрій для обчислення модульної функції (унітарний ТЧБ)		$G_{xx}(j) = \text{res}\left(\sum_{i=1}^n  x_i - x_{i+j}  \right) \text{mod } P$
2.	Пристрій для обчислення структурної функції (унітарний ТЧБ)		$C_{xx}(j) = \frac{1}{n} \sum_{i=1}^n (x_i - x_{i+j})^2$

Продовження таблиці 1.5

<p>3.</p>	<p>Пристрій для обчислення функції еквівалентності (ТЧБ Хаара)</p>		$F_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \tilde{Z}_{xx}$
<p>4.</p>	<p>Модульно-матричний цифровий корелятор (унітарний ТЧБ)</p>		$C_{xy}(0) = \frac{1}{n} \sum_{i=1}^n  x_i - y_i $
<p>5</p>	<p>Корелятор на основі АЦП послідовного наближення, у якому відсутні модулі перемноження</p>		$K_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot x_{i+j}$
<p>6</p>	<p>Пристрій обчислення знакової автоковаріаційної функції на основі кодів унітарного ТЧБ</p>		$H_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \text{sign}^\circ(x_i) \times \text{sign}^\circ(x_{i+j})$

7	Автоковаріаційний спецпроцесор у базисі Хаара-Крестенсона		$K_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot x_{i+j}$
---	---	--	--

Особливістю даних структур є застосування багатоканального регістра зсуву, на виході АЦП формується паралельний унітарний код, а також відбувається перемноження цифрових даних та їх додавання в накопичуючих суматорах.

Аналіз цифрових кореляторів показує, що за останні двадцять років велика кількість фірм займаються розробкою та тиражуванням кореляційних процесорів та інформаційних систем на їх основі [47-50].

Проведена класифікації кореляційних засобів цифрового опрацювання інформаційних потоків, яка дозволяє систематизувати та класифікувати їх типи згідно ознак різних аналітичних виразів функцій кореляції та способів кодування цифрових даних у різних ТЧБ для різних сфер застосування (рис. 1.7) [55-77].

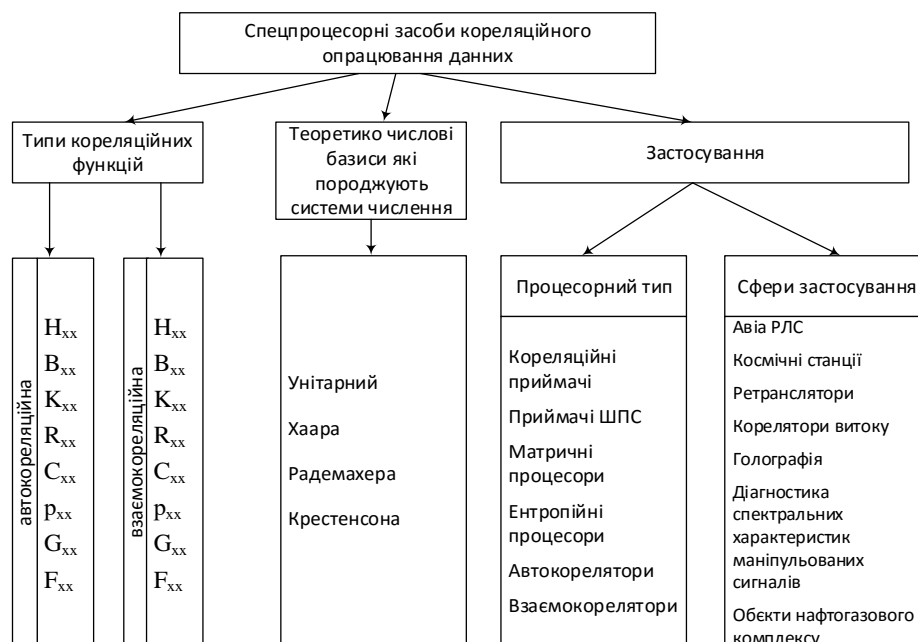


Рисунок 1.7 – Класифікація кореляційних засобів цифрового опрацювання інформаційних потоків

Аналіз застосування цифрових кореляторів, теоретичних засад обчислення та властивостей кореляційних функцій, класів процесорних засобів кореляційної обробки інформаційних потоків, показує перспективу розвитку та вдосконалення методів цифрового спектрального опрацювання сигналів шляхом застосування дискретних функцій кореляції в якості базових компонентів спектральних спецпроцесорів. Також слід виділити наступні базові компоненти цифрових кореляторів:

- одноканальні та багатоканальні аналого-цифрові перетворювачі;
- накопичуючі суматори та шифратори даних ;
- квадратори та перемножувачі дискретних даних у різних ТЧБ;
- багатоканальні багаторозрядні регістри зсуву (БРЗ) даних .

У результаті проведеного аналізу встановлено (рис. 1.7), що цифрові кореляційні спецпроцесори можуть бути реалізовані на основі різних дискретних автокореляційних функцій [36]: релейної -  $H_{xx}$ , знакової -  $V_{xx}$ , коваріаційної –  $K_{xx}$ , кореляційної –  $R_{xx}$ , структурної –  $C_{xx}$ , нормованої кореляційної -  $\rho_{xx}$ , модульної –  $G_{xx}$ , еквівалентності -  $F_{xx}$  . Для обґрунтування ефективності їх застосування для побудови спектральних спецпроцесорів з різними проблемно-орієнтованими мінімаксними характеристиками, необхідно виконати дослідження аналітики, асимптотики, інформаційних та інших характеристик названих кореляційних функцій.

Незважаючи на широке застосування структур цифрових кореляторів в унітарному базисі, вони показують свої функціональні недоліки:

- велика апаратна складність багаторозрядних регістрів зсуву за рахунок представлення даних в унітарному базисі у діапазоні  $2^n$ , де число рівнів квантування вхідних АЦП при 8 та 10 біт АЦП – відповідно 256 та 1024 рівнів квантування;

- низька швидкодія обчислень та, відповідно, нездатність опрацювання інформаційних потоків у режимі on-line, а також велика часова складність при цифровому опрацюванні унітарних кодів у діапазоні 256 та 1024 бітів.

#### 1.4 Дискретні кореляційні функції, їх аналітика, та системні властивості

Розробкою теоретичних засад побудови спецпроцесорів кореляційної обробки інформаційних потоків займалися відомі зарубіжні та українські вчені: Грібанов Ю.І., Золотухін Г.А., Николайчук Я.М., Іщеряков С.М., Кожемяко В.П., Сегін А.І. [14, 16, 17, 19, 20, 4]. В роботах [78-80] розроблені теоретичних та реалізація кореляційного опрацювання сигналів на основі решітчастих функцій, які формуються на виході аналого-цифрового перетворювача в процесі перетворення аналогових сигналів у цифрові.

В роботі [108] систематизовані методи та аналітичні функції кореляційних перетворень сигналів на основі дискретних інтегральних оцінок (таб. 1.6).

Таблиця 1.6 - Кореляційні функції на основі різних дискретних інтегралів

№	Кореляційна функція
1	<p>знакова</p> <p>Автокореляційна функція <math>H_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \text{sign}(x_i) \times \text{sign}(x_{i+j})</math></p> <p>Взаємкореляційна функція <math>H_{xy}(j) = \frac{1}{n} \sum_{i=1}^n \text{sign}(x_i) \times \text{sign}(y_{i+j})</math></p>
2	<p>релейна</p> <p>Автокореляційна функція <math>B_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \times \text{sign}(x_{i+j})</math></p> <p>Взаємкореляційна функція <math>B_{xy}(j) = \frac{1}{n} \sum_{i=1}^n x_i \times \text{sign}(y_{i+j})</math></p>
3	<p>коваріаційна</p> <p>Автокореляційна функція <math>K_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \times x_{i+j}</math></p> <p>Взаємкореляційна функція <math>K_{xy}(j) = \frac{1}{n} \sum_{i=1}^n x_i \times y_{i+j}</math></p>

Продовження таблиці 1.6

4	<p style="text-align: center;">Кореляційна</p> <p>Автокореляційна функція <math>R_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \times x_{i+j}</math></p> <p>Взаємкореляційна функція <math>R_{xy}(j) = \frac{1}{n} \sum_{i=1}^n x_i \times y_{i+j}</math></p>
5	<p style="text-align: center;">Нормована кореляційна</p> <p>Автокореляційна функція <math>\rho_{xx}(j) = \frac{R_{xx}(j)}{D_x}</math></p> <p>Взаємкореляційна функція <math>\rho_{xy}(j) = \frac{R_{xy}(j)}{\sqrt{D_x \times D_y}}</math></p>
6	<p style="text-align: center;">структурна</p> <p>Автокореляційна функція <math>C_{xx}(j) = \frac{1}{n} \sum_{i=1}^n (x_i - x_{i+j})^2</math></p> <p>Взаємкореляційна функція <math>C_{xy}(j) = \frac{1}{n} \sum_{i=1}^n (x_i - y_{i+j})^2</math></p>
7	<p style="text-align: center;">модульна</p> <p>Автокореляційна функція <math>G_{xx}(j) = \frac{1}{n} \sum_{i=1}^n  x_i - x_{i+j} </math></p> <p>Взаємкореляційна функція <math>G_{xy}(j) = \frac{1}{n} \sum_{i=1}^n  x_i - y_{i+j} </math></p>
8	<p style="text-align: center;">еквівалентності</p> <p>Автокореляційна функція <math>F_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \tilde{Z}_{xx}</math></p> <p>Взаємкореляційна функція <math>F_{xy}(j) = \frac{1}{n} \sum_{i=1}^n \tilde{Z}_{xy}</math></p>

Як показано у таб. 1.6, існує вісім відомих аналітичних виразів для розрахунку кореляційних функцій, в яких використовуються знакові, центровані та нецентровані значення цифрових відліків, а також різні оцінки дискретних інтегралів. Відповідно їх алгоритми обчислення будуть характеризуватися різним ступенем інформативності, часової та апаратної складності.

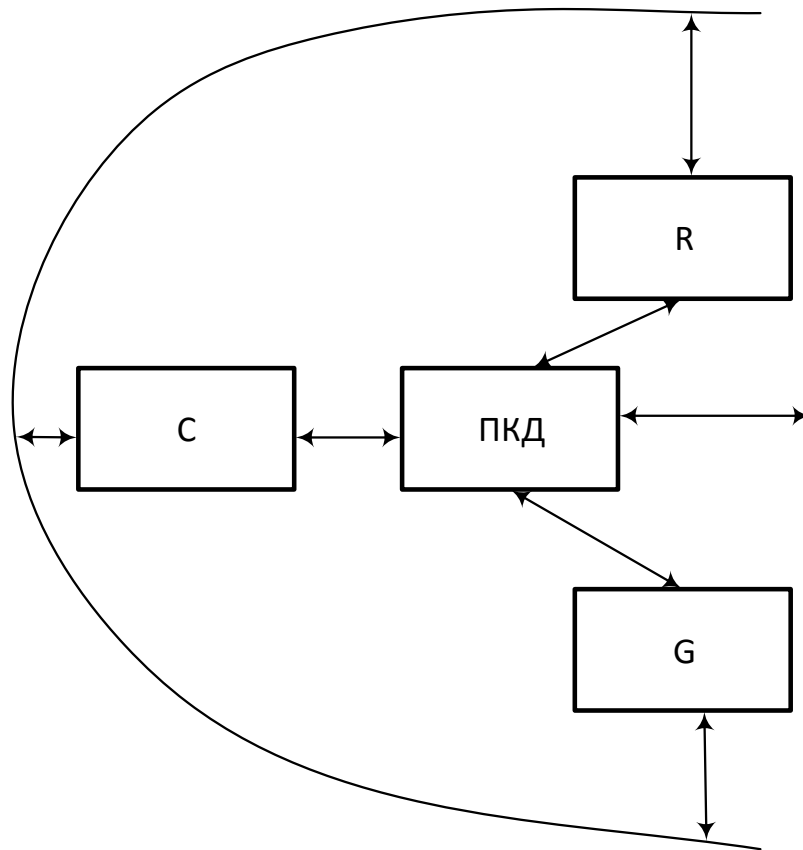
Аналіз аналітичних виразів кореляційних функцій, які подані в таблиці 1.6, показує, що найменшою алгоритмічною складністю характеризується функція еквівалентності  $F_{xx}(j)$ , у якій в якості базисної операції згортки використовується операція порівняння модульних значень  $x_i, \dots, x_{i+j}$  та сумування менших з них.

Аналітика найбільш продуктивних цифрових кореляційних функцій ( $C_{xy}, G_{xy}, F_{xy}$ ) реалізована у вигляді спецпроцесорів приймання ШПС в унітарному ТЧБ.

### 1.5 Теоретико-числові бази та перспективи вдосконалення кореляційних спецпроцесорів на їх основі, постановка задачі досліджень

Аналіз сучасних методів цифрової обробки інформації показує, що алгоритми кореляційного аналізу реалізуються на основі мультиплікативних функцій виключно в базисі Радемахера та в двійковій системі числення. Також використання теоретико-числових базисів: унітарного, Хаара, Крестенсона, Крейга, Радемахера, Уолша, Галуа, [89-92] є перспективними при побудові спецпроцесорів кореляційної обробки даних. Серед вищезгаданих базисів особливу роль складають бази Крестенсона та Галуа, які доцільно застосовувати не тільки для побудови окремих монобазисних процесорів, а також проблемно-орієнтованих мультибазисних процесорів як це показано в роботах [92], (рис. 1.6).





ПКД – пам'ять колективного доступу, R – процесор Радемахера,  
 С – процесор Крестенсона, G – процесор Галуа

Рисунок 1.6 - Архітектура мультибазисного спецпроцесора

Суттєвою перевагою та особливістю таких RCG - процесорів є можливість використання в якості комунікаційного модуля ПКД, яка представляє собою асоціативну багато портову пам'ять з паралельним доступом [93]. За рахунок оперативного перерозподілу арифметико-логічних операцій, які максимально ефективно виконуються окремими монобазисними модулями процесора відбувається підвищення швидкодії обчислень на один, два порядки. Таким чином, використання базисів Радемахера, Крестенсона та Галуа може бути ефективним для побудови високопродуктивних спецпроцесорів спектрального опрацювання даних.

Сучасний розвиток техніки, теорії та методології побудови процесорів комп'ютерних систем обумовлені високим насиченням як теоретичними так і ідейними можливостями застосування базису Радемахера для побудови

арифметико-логічних компонентів процесорів, до яких ставляться все вищі вимоги щодо швидкодії та покращення функціональних можливостей.

У табл. 1.7 приведені загальні характеристики кодових матриць ТЧБ, які найширше використовуються для кодування та цифрової обробки даних в інформаційних системах : Радемахера, Крестенсона, Унітарного, Хаара а також Радемахера-Крестенсона та Хаара-Крестенсона. [92-94, 108].

Таблиця 1.7 – ТЧБ, які породжують системи числення

Базис	Кодові матриці	N	V
1	2	3	4
Унітарний	$M_{uni} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix}$	$n = \frac{N^2}{2}$	$V = N^2$
Хаара	$M_{har} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix}$	$n = N$	$V = N^2$
Радемахера	$M_{Rad} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix}$	$n = \frac{N \cdot \log_2 N}{2}$	$V = N \cdot \log_2 N$
Крестенсона	$M_{Cres} = \begin{vmatrix} P_1 & P_2 & \dots & P_n \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ 0 & 3 & \dots & 3 \\ 1 & 4 & \dots & 4 \\ 2 & 0 & \dots & 5 \\ 0 & 1 & \dots & 6 \\ \dots & \dots & \dots & \dots \\ a_1 & a_2 & \dots & a_n \end{vmatrix}$	$n = \prod_{i=1}^m P_i$	$V = \sum_{i=1}^m \log_2 (P_i - 1)$

Продовження таблиці 1.7

<p>Радемахера-Крестенсона</p>	<table border="1"> <tr> <td>P<sub>1</sub></td> <td>P<sub>2</sub></td> <td>·</td> <td>P<sub>n</sub></td> </tr> <tr> <td>01</td> <td>000</td> <td>·</td> <td>00...00</td> </tr> <tr> <td>10</td> <td>001</td> <td>·</td> <td>00...01</td> </tr> <tr> <td>11</td> <td>010</td> <td>·</td> <td>00...10</td> </tr> <tr> <td>01</td> <td>011</td> <td>·</td> <td>.....</td> </tr> <tr> <td>10</td> <td>100</td> <td>·</td> <td>01...10</td> </tr> <tr> <td>11</td> <td>000</td> <td></td> <td>01...11</td> </tr> </table>	P <sub>1</sub>	P <sub>2</sub>	·	P <sub>n</sub>	01	000	·	00...00	10	001	·	00...01	11	010	·	00...10	01	011	·	.....	10	100	·	01...10	11	000		01...11	$n = \frac{N}{2} \cdot \sum_{i=1}^n \log_2 p_{(i-1)}$	$V = N \cdot \sum_{i=1}^n \log_2 p_{(i-1)}$
P <sub>1</sub>	P <sub>2</sub>	·	P <sub>n</sub>																												
01	000	·	00...00																												
10	001	·	00...01																												
11	010	·	00...10																												
01	011	·	.....																												
10	100	·	01...10																												
11	000		01...11																												
<p>Хаара-Крестенсона</p>	<table border="1"> <tr> <td>P<sub>1</sub></td> <td>P<sub>2</sub></td> <td>·</td> <td>P<sub>n</sub></td> </tr> <tr> <td>100</td> <td>10000</td> <td>·</td> <td>100...00</td> </tr> <tr> <td>010</td> <td>01000</td> <td>·</td> <td>010...00</td> </tr> <tr> <td>001</td> <td>00100</td> <td>·</td> <td>001...00</td> </tr> <tr> <td>100</td> <td>00010</td> <td>·</td> <td>.....</td> </tr> <tr> <td>010</td> <td>00001</td> <td>·</td> <td>000...10</td> </tr> <tr> <td>001</td> <td>10000</td> <td></td> <td>000...01</td> </tr> </table>	P <sub>1</sub>	P <sub>2</sub>	·	P <sub>n</sub>	100	10000	·	100...00	010	01000	·	010...00	001	00100	·	001...00	100	00010	·	.....	010	00001	·	000...10	001	10000		000...01	$n = \sum_{i=1}^n p_i$	$V = N \cdot \sum_{i=1}^n p_i$
P <sub>1</sub>	P <sub>2</sub>	·	P <sub>n</sub>																												
100	10000	·	100...00																												
010	01000	·	010...00																												
001	00100	·	001...00																												
100	00010	·	.....																												
010	00001	·	000...10																												
001	10000		000...01																												

У таб. 1.7: n – число інформативних елементів, N – діапазон представлених чисел, V – об’єм кодової матриці, P<sub>1</sub>, P<sub>2</sub>...P<sub>i</sub>...P<sub>m</sub> – набір взаємо-простих модулів СЗК базису Крестенсона [108].

На діаграмах показані характеристики надлишковості кодових матриць різних ТЧБ та часова складність операцій множення.

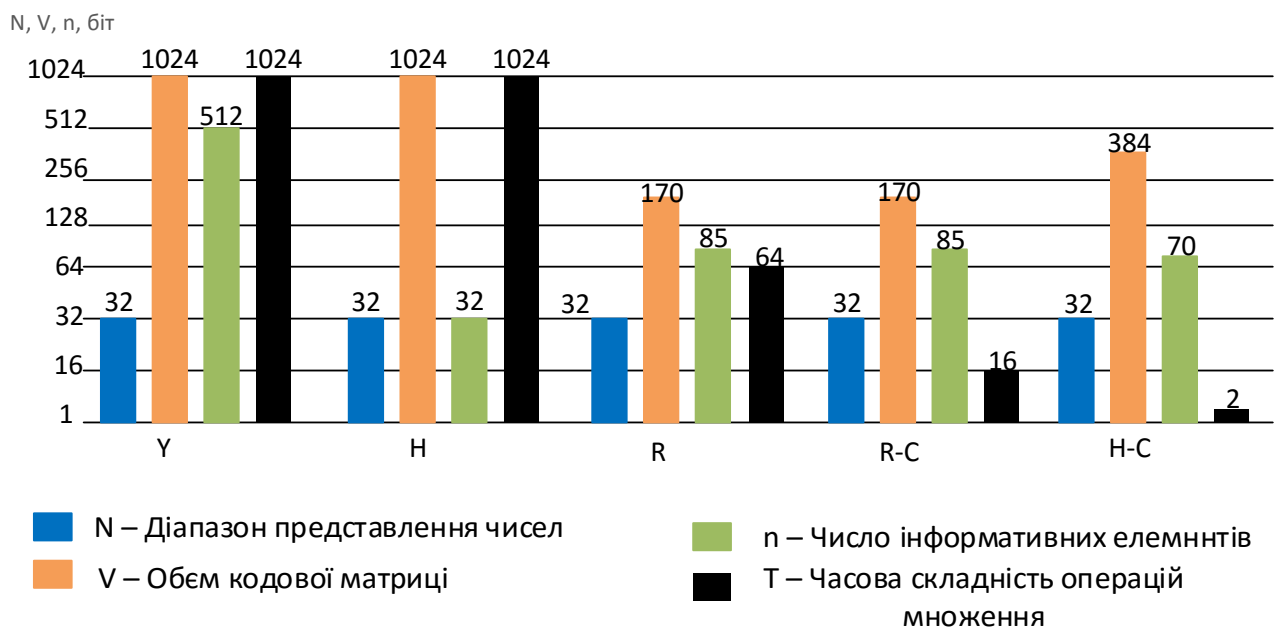


Рисунок 1.7 - Діаграма характеристик надлишковості ТЧБ:

Y- унітарний базис, H- базис Хаара, R- базис Радемахера, R-C - базис Радемахера-Крестенсона, H-C- базис Хаара-Крестенсона.

Аналіз рис.1.7 показує, що найбільш надлишковим ТЧБ є унітарний, в якого кодова матриця  $V = N^2$ . Аналогічною надлишковістю характеризується базис Хаара, в якому набагато менше активних елементів у порівнянні з унітарним базисом. ТЧБ Радемахера та Крестенсона, характеризуються набагато меншими об'ємами кодових матриць  $V = N \log_2 N$ . Дані базиси відповідно породжують позиційну двійкову систему числення та непозиційну систему числення залишкових класів.

Проведений аналіз теорії спектрального косинусного перетворення Фур'є та систематизація кореляційних спецпроцесорів на основі асимптотики різних дискретних кореляційних функцій дозволяє сформулювати задачу дослідження та вдосконалення процесорів спектрального опрацювання сигналів з покращеними системними характеристиками та розширеними функціональними можливостями в наступному вигляді:

- 1) проаналізувати існуючий стан побудови та застосування спектральних спецпроцесорів в інформаційних системах;
- 2) систематизувати структури, дослідити асимптотику та системні характеристики дискретних автокореляційних функцій, як базових компонентів процесорів косинусного перетворення Фур'є;
- 3) провести аналіз характеристик кодових матриць ТЧБ Унітарного, Хаара, Радемахера та Крестенсона;
- 4) розробити метод швидкодіючого аналого-цифрового перетворення сигналів у ТЧБ Хаара-Крестенсона;
- 5) розробити метод дискретного косинусного перетворення Фур'є на основі адаптації характеристик ортогональних базисних функцій до експериментальної реалізації коваріаційної функції автокореляції;
- 6) розробити метод генерування ортогональних функцій різних ТЧБ, адаптованих до характеристик функцій кореляції;
- 7) розробити структуру та мікроелектронні компоненти швидкодіючого АЦП у

базисі Хаара-Крестенсона;

8) розробити структуру швидкодіючого спектрального спецпроцесора, з розширеними функціональними можливостями у базисі Хаара-Крестенсона;

9) розробити програмне забезпечення для розрахунку наборів взаємо-простих модулів спектральних спецпроцесорів у базисі Хаара-Крестенсона різної розрядності та синтезу компонентів на ПЛІС.

Теоретичне обґрунтування виконання алгоритмів у системі числення залишкових класів ТЧБ Крестенсона, дослідження та реалізація спецпроцесорів спектрального опрацювання сигналів у базисі Хаара-Крестенсона є предметом дослідження даної дисертації.

## ВИСНОВКИ ПО ПЕРШОМУ РОЗДІЛІ

1. Виконана систематизація процесорів цифрового спектрального опрацювання сигналів на основі різних архітектур та сфери застосувань у широкому класі задач цифрового опрацювання даних шляхом застосування різних теоретико-числових базисів, що дозволило обґрунтувати перспективу розробки та реалізації швидкодіючих спектральних спецпроцесорів у системі числення залишкових класів у базисі Хаара-Крестенсона.

2. Викладені теоретичні засади спектрального аналізу на основі інтегрального та дискретного перетворень Фур'є, які дозволили оцінити переваги та функціональні обмеження відомих алгоритмів реалізації швидкого та косинусного перетворень Фур'є, при використанні двійкової системи числення базису Радемахера.

3. Класифіковані структури цифрових кореляторів, які є базовими компонентами спектральних спецпроцесорів косинусного перетворення Фур'є, реалізованих на основі різних аналітичних виразів автокореляційних функцій та досліджені їх системні характеристики часової складності, що дозволило обґрунтувати доцільність реалізації алгоритму автоковаріаційної згортки сигналів у структурі спектрального спецпроцесора.

4. Проаналізовані властивості кодових систем ортогональних функцій теоретико-числових базисів типу: унітарного, Хаара, Радемахера та Крестенсона, в результаті встановлено ефективність застосування базису Хаара-Крестенсона для реалізації швидкодіючих модульно-матричних обчислювальних операцій множення та додавання компонентами спектральних спецпроцесорів.

5. Виконана постановка задачі дисертаційних досліджень.

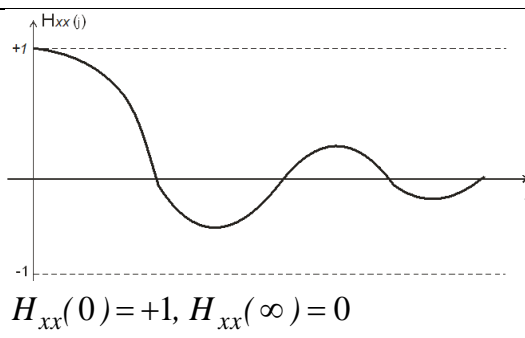
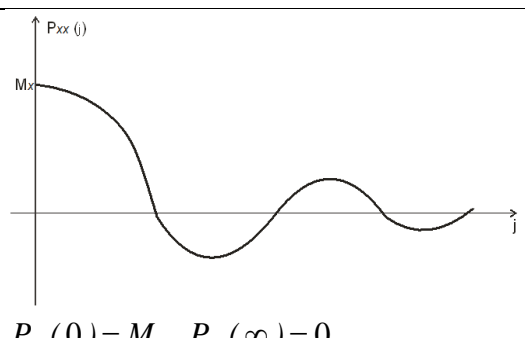
РОЗДІЛ 2  
РОЗРОБКА МЕТОДУ ТА ТЕОРЕТИЧНИХ ЗАСАД ПОБУДОВИ  
КОМПОНЕНТІВ СПРЕЦПРОЦЕСОРА ДИСКРЕТНОГО КОСИНУСНОГО  
ПЕРЕТВОРЕННЯ ФУР'Є

2.1 Теоретичні основи побудови та дослідження асимптотики дискретних автокореляційних функцій

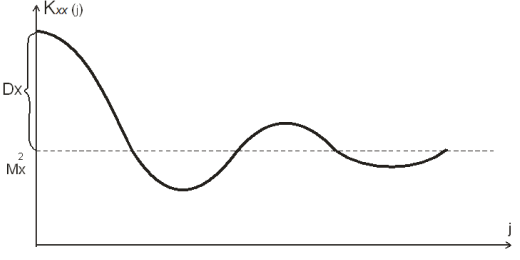
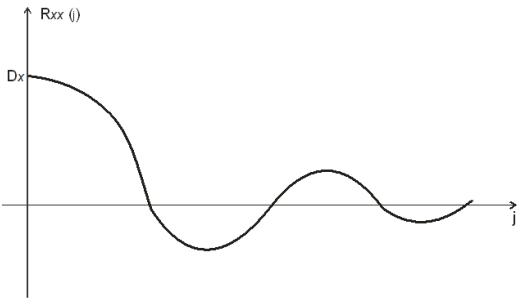
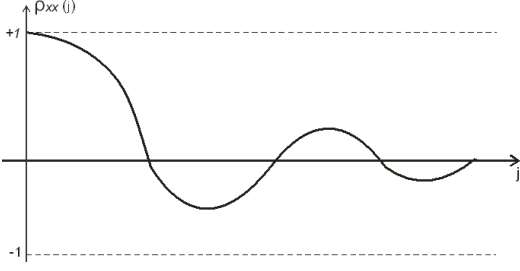
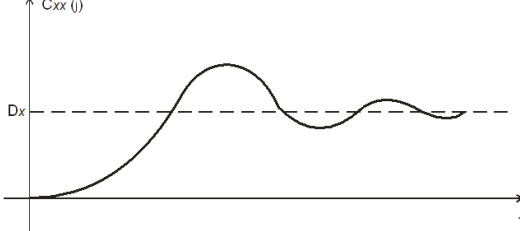
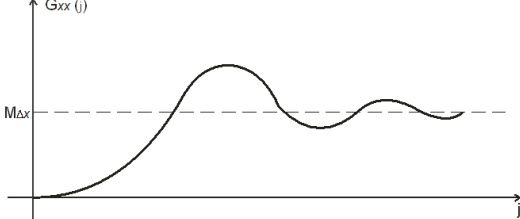
У результаті обчислення різних АКФ існує можливість визначення важливих характеристичних оцінок, які можуть бути ефективно використані при реалізації алгоритмів спектрального аналізу сигналів та зменшення апаратної та часової складності спецпроцесорів.

В роботах [110,111,112] досліджені характеристики, приведені таблиці аналітики та асимптотики наступних АКФ (табл. 2.1).

Таблиця 2.1 - Аналітика та асимптотика відомих дискретних автокореляційних функцій

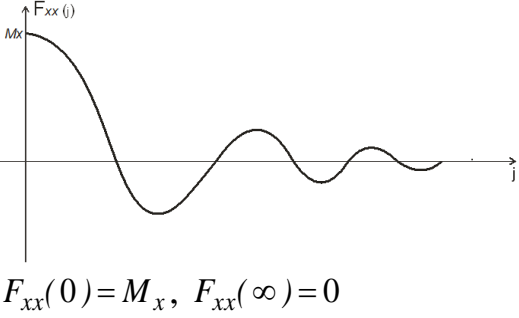
№ п/п	Кореляційна функція	Графічне зображення та асимптотика	Аналітичний вираз
1	знакова	 <p style="text-align: center;"><math>H_{xx}(0) = +1, H_{xx}(\infty) = 0</math></p>	$H_{xx}(j) = \frac{1}{n} \sum_{i=1}^n \overset{o}{\text{sign}} x_i \cdot \overset{o}{\text{sign}} x_{i+j},$ $\overset{o}{\text{sign}} x_i = \begin{cases} +1, & x_i \geq 0 \\ -1, & x_i < 0 \end{cases}$
2	релейна	 <p style="text-align: center;"><math>P_{xx}(0) = M_x, P_{xx}(\infty) = 0</math></p>	$P_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot \overset{o}{\text{sign}} x_{i+j}$

Продовження таблиці 2.1

3	коваріаційна	 <p><math>K_{xx}(0) = D_x + M_x^2, K_{xx}(\infty) = M_x^2</math></p>	$K_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot x_{i+j}$
4	кореляційна	 <p><math>R_{xx}(0) = D_x, R_{xx}(\infty) = 0</math></p>	$R_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i \cdot x_{i+j}$
5	нормована кореляційна	 <p><math>\rho_{xx}(0) = +1, \rho_{xx}(\infty) = 0</math></p>	$\rho_{xx}(j) = \frac{R_{xx}(j)}{D_x},$ $D_x = \frac{1}{n} \sum_{i=1}^n (x_i - M_x)^2$
6	структурна	 <p><math>C_{xx}(0) = 0, C_{xx}(\infty) = D_x</math></p>	$C_{xx}(j) = \frac{1}{n} \sum_{i=1}^n (x_i - x_{i+j})^2$
7	модульна	 <p><math>G_{xx}(0) = 0, G_{xx}(\infty) = M_x</math></p>	$G_{xx}(j) = \frac{1}{n} \cdot \sum_{i=1}^n  x_i - x_{i+j} $



Продовження таблиці 2.1

8	еквівалентності	 <p style="text-align: center;"><math>F_{xx}(0) = M_x, F_{xx}(\infty) = 0</math></p>	$F_{xx}(j) = \frac{1}{n} \sum_{i=1}^n Z_{xx}^{\vee}$ $Z_{xx}^{\vee} = \begin{cases} x_i, & x_i < x_{i+j} \\ x_{i+j}, & x_i \geq x_{i+j} \end{cases}$
---	-----------------	--	--

Аналіз (табл. 2.1) аналітичних виразів та асимптотики АКФ типу: знакової  $H_{xx}$ , релейної  $R_{xx}$ , коваріаційної  $K_{xx}$ , центрованої кореляційної  $R_{xx}$ , нормованої кореляційної  $\rho_{xx}$ , структурної  $S_{xx}$ , модульної  $G_{xx}$  та еквівалентності  $F_{xx}$  показує, що існує два типи дзеркальних відображень характеристик АКФ з асимптотикою:

- 1)  $H_{xx}, R_{xx}, R_{xx}, \rho_{xx}, K_{xx}, F_{xx} = 0$  при  $j \rightarrow \infty$ ;
- 2)  $S_{xx} = D_x, G_{xx} = M_x$  при  $j \rightarrow \infty$ .

Дані АКФ характеризуються зменшенням інформативності при зростанні значення  $j$  оскільки відповідно зменшується точність представлення їх цифрових відліків у квадратичному та лінійних просторах. Найвищою інформаційністю серед вищезгаданих АКФ володіє функція еквівалентності, яка представлена у лінійному просторі, тому може бути розрахована на основі найменшого об'єму вибірки вхідних цифрових даних  $N \geq 128$ .

Особливі характеристики має знакова  $H_{xx}$  АКФ, оскільки вона центрована і нормована, тобто має діапазон представлення  $\pm 1$ , що дозволяє на її базі безпосередньо розраховувати спектри згідно дискретного косинусного перетворення Фур'є. У той же час така аналітика та асимптотика  $H_{xx}$ , коли втрачається інформація про цифрові значення вхідних сигналів, які представляються числами  $\pm 1$  потребує практично найбільшого об'єму вибірки  $N \geq 1024$ . Ця АКФ характеризується найпростішим алгоритмом обчислення обумовленого представленням цифрових значень опрацьовуваних сигналів у вигляді  $\pm 1$ , що виключає складну операцію перемноження текучих та зсунутих на  $j$  тактів цифрових ознак сигналів.

Більш значною інформативністю характеризується релейна АКФ  $R_{xx}$ , яка в нульовій точці має значення математичного сподівання  $M_x$ , що виключає необхідність додаткового обчислення цієї оцінки при спектральному аналізі.

Центрована кореляційна АКФ  $R_{xx}$ , яка характеризується мультиплікативними властивостями і дозволяє визначити значення дисперсії в нульовій точці, найчастіше використовується при аналітичному вирішенні задач спектрального аналізу.

Позитивною якістю цієї оцінки є визначення дисперсії  $D_x$  у нульовій точці. Дана АКФ також характеризується середньою оцінкою інформативності оскільки її енергія згасає у квадратичному просторі при зростанні зсуву  $j$  і тому потребує об'єму вибірки  $N \geq 512$ . Складність алгоритму обчислення даної АКФ обумовлена необхідністю попереднього розрахунку математичного сподівання  $M_x$  та визначення центрованих значень вхідних сигналів  $\dot{x}_i$ , а також виконання ускладнених перемножень знакозмінних цифрових даних, що відповідно ускладнює реалізацію цифрових процесорів.

АКФ  $\rho_{xx}$  характеризується найскладнішим алгоритмом обчислення у процесі адаптації до базисної функцій перетворення Фур'є, оскільки вимагає послідовного виконання операцій визначення математичного сподівання, центрування, розрахунку дисперсії, обчислення центрованої АКФ  $R_{xx}$ , та її нормування шляхом ділення на дисперсію  $D_x$ . В той же час дана АКФ в результаті вищезгаданих обчислень є нормована і центрована і також забезпечує можливість безпосередньо розраховувати спектри згідно косинусного перетворення Фур'є без адаптації ортогональних базисних функцій ГЧБ.

Другий клас АКФ  $S_{xx}$  та  $G_{xx}$  не є центрованими, мають значення нуль у нульовій точці і відповідно представляються у квадратичному та лінійному просторах з асимптотикою  $D_x$  та  $M_x$  при  $j \rightarrow \infty$ . Такі АКФ широко використовуються в техніці побудови цифрових приймачів модульованих та маніпульованих сигналів, а також при вирішенні задачі розпізнавання образів. Інформативність цього класу АКФ зростає при зростанні  $j$ , тому потребує зменшеного об'єму вибірки  $N \geq 256$ .

Алгоритми обчислення таких АКФ простіші по відношенню до функцій  $R_{xx}$  та  $\rho_{xx}$ , тому що в алгоритмах їх обчислення відсутні операції множення.

Аналіз аналітики та асимптотики коваріаційної АКФ  $K_{xx}$  показує, що дана функція в нульовій точці має значення  $D_x + M_x^2$ , а при  $j \rightarrow \infty$  відповідно асимптотично наближається до  $M_x^2$ . Така АКФ є мультиплікативною, представлена у квадратичному просторі з високою інформативністю, не потребує додаткових операцій визначення математичного сподівання  $M_x$ , центрування та дисперсії  $D_x$  і забезпечує високу інформативність обчислень при об'ємі вибірки  $N \geq 256$ .

Таким чином проведений аналіз аналітики та асимптотики різних відомих типів АКФ дозволяє обґрунтувати перспективність застосування коваріаційної АКФ при реалізації алгоритмів спектрального косинусного перетворення Фур'є, а також побудови спеціалізованих процесорів на основі синтезу їх компонентів.

Функціональним обмеженням таких АКФ є нецентрованість та ненормованість, що потребує розробки алгоритмів адаптації базисних ортогональних функцій ТЧБ до реалізації коваріаційної АКФ, на основі яких розраховуються спектри сигналів.

## 2.2 Дослідження системних характеристик базових компонентів спектральних спецпроцесорів на основі теоретико-числового базису Хаара-Крестенсона

### 2.2.1 Дослідження апаратної та часової складності аналого-цифрового перетворювача у різних теоретико-числових базисах

Аналого-цифрові перетворювачі широко використовуються у сучасних інформаційних системах [94]. Існують різні типи, АЦП серед яких треба виділити одноканальні та багатоканальні АЦП. Багатоканальні АЦП (БАЦП) практично застосовуються в інформаційних системах моніторингу та управління багатопараметричними технологічними об'єктами. Розрізняють наступні типи БАЦП:

- 1) з аналоговим вхідним комутатором;

- 2) з дискретним цифровим комутатором;
- 3) з певним числом одноканальних АЦП;
- 4) з паралельною розгорткою в унітарному ТЧБ.

Суттєвими недоліками БАЦП з вхідними комутаторами є поява ефектів старіння інформації, які полягають у тому що різні процеси реєструються у різні моменти часу, що приводить до декореляції їх характеристик а також до суттєвого спотворення спектральних характеристик, які обчислюються в реальному масштабі часу. Тому застосування БАЦП з вхідними комутаторами є неефективним при їх застосуванні у якості базових модулів взаємокореляторів та спектральних процесорів.

БАЦП з паралельною розгорткою в унітарному ТЧБ характеризується спрощеною апаратною реалізацією, оскільки у кожному каналі використовується один компаратор, на входи яких подаються потенціали виходів сенсорів різних технологічних параметрів і використовується один цифро-аналоговий перетворювач та двійковий лічильник, який тактується генератором імпульсів.

Перевагою такого БАЦП є одночасний старт процесу вимірювання багатьох технологічних параметрів. У той же час завершення процесу аналого-цифрового перетворення відбувається в різні моменти часу на інтервалі дискретизації. Таким чином БАЦП даного класу також характеризується ефектом старіння інформації, що обмежує можливість їх застосування для взаємокореляційного та спектрального аналізу.

Успіхи мікроелектроніки та реалізації одноканальних АЦП на кристали визначає перспективу їх застосування для побудови спектральних спецпроцесорів. У табл. 2.2 наведені структурні схеми одноканальних АЦП, які реалізовані у різних ТЧБ: унітарному, Радемахера та Крестенсона. На виході різних типів АЦП можуть формуватися унітарні, число-імпульсні, порозрядні, рекурентні та паралельні коди згідно табл. 2.2.

Оскільки при розробці спектральних процесорів найважливішою системною характеристикою є швидкодія то доцільним є дослідження часової та апаратної

складності одноканальних АЦП паралельного типу, які можуть бути застосовані у якості базових компонентів досліджуваного класу спецпроцесорів.

На рис. 2.1 приведена класифікація одноканальних АЦП паралельного типу у різних ТЧБ

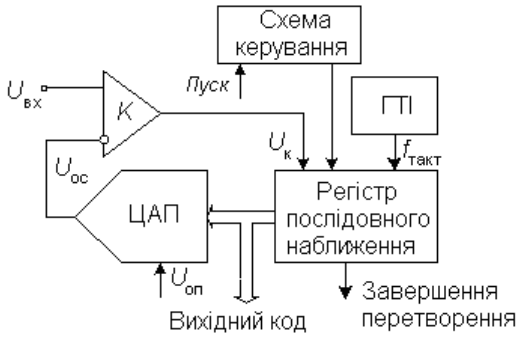
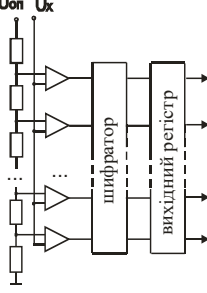
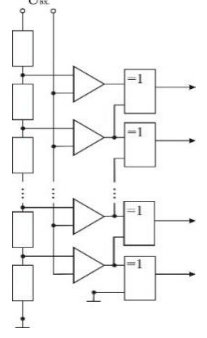


Рисунок 2.1 - Класифікація АЦП

Таблиця 2.2 – АЦП, їх структури та вихідні коди у різних ТЧБ

№	Тип АЦП	Структура	Параметри вихідних кодів, ТЧБ
1	2	3	4
1	АЦП розгортуючого типу		Паралельний код в базисі Крестенсона $b_1$ $b_2$ $\dots$ $b_k$ $c$ $S_1$ $S_2$ $\dots$ $S_k$
2	АЦП розгортуючого типу		Паралельний код у базисі Хаара $0$ $\dots$ $H_{(0)}$ $\dots$ $0$

## Продовження таблиці 2.2

3	АЦП порозрядного зрівноваження		Послідовний та паралельний код базису Радемахера, $R_k$ $R_k$ $R_{k-1}$ $R_{k-1}$ ...                            ... $R_v$ $R_v$ ...                            ... $R_1$ $R_1$
4	Паралельний АЦП		Паралельний код базису Радемахера $R_k$ $R_{k-1}$ ..
5	Паралельний АЦП		Паралельний код у базисі Хаара 0 ... $H_{(v)}$ ... 0

Оцінки часової складності одноканальних АЦП з паралельними вихідними кодами у різних ТЧБ приведені у табл. 2.3.

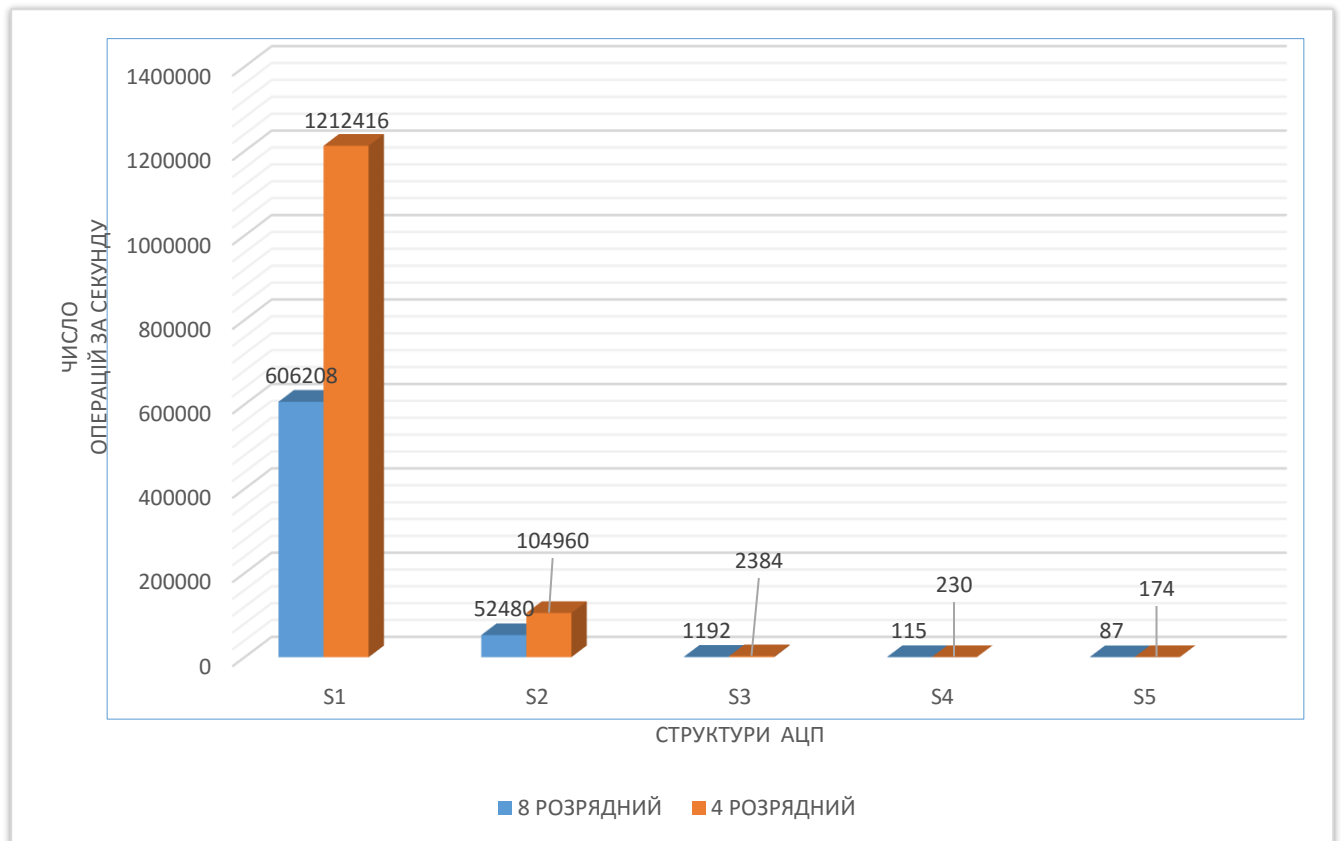
Таблиця 2.3 - Оцінки часової складності АЦП

№	Типи АЦП (ТЧБ)	Аналітичний вираз розрахунку часу затримки	Швидкість мГц
1	2	3	4
3	АЦП розгортуючого типу у базисі Крестенсона	$T=2^k(t_{\Gamma}+t_{\Delta}+t_{\text{ЦАП}}+t_{\text{К}}+t_{\text{ЛЕ}}+t_{\text{ЛК}})$	2,24
4	Паралельний АЦП розгортуючого типу (Хаара)	$T=2^k(t_{\Gamma}+t_{\Delta}+t_{\text{ЦАП}}+t_{\text{К}})$	2,61

## Породовження таблиці 2.3

5	АЦП порозрядного зрівноваження (R - Радемахера)	$T=(t_k+t_p+t_r+t_{цАП})k$	87,78
6	Паралельний АЦП (R - Радемахера)	$T=t_k+t_{ш}+t_p$	967,11
7	Паралельний АЦП (H - Хаара)	$T=t_k+t_{ЛЕ}$	986,43

Швидкодія досліджуваних типів однаканальних АЦП з паралельними вихідними кодами розрахована на основі табл 2.3 а оцінки часової складності приведені на діаграмі (рис. 2.2.) у залежності від розрядності АЦП.



S1-АЦП розгортуючого типу (С - Крестенсона);

S2- АЦП розгортуючого типу (H - Хаара);

S3-АЦП порозрядного зрівноваження (R - Радемахера);

S4- паралельний АЦП (R - Радемахера);

S5- паралельний АЦП (H - Хаара);

Рисунок 2.2 – Діаграма часової та апаратної складності різних типів АЦП

З діаграми поданої на рису. 2.2 видно, що максимальною швидкістю і відповідно мінімальною часовою складністю характеризуються паралельні АЦП у базисі Радемахера та Хаара. Найнижчою швидкістю характеризується АЦП

розгортуючого типу. Таким чином застосування АЦП паралельного типу у базисі Хаара-Крестенсона є обґрунтованим для їх використання в якості компонентів високопродуктивних спектральних перетворювачів.

Апаратна складності різних типів АЦП розраховується згідно виразу [34].

$$A_j = \sum_{i=1}^l \beta_i P_{ij}, \quad (2.1)$$

де  $\beta_i$  – ваговий коефіцієнт, який враховує апаратну складність  $i$ -го компонента АЦП, який визначається методом експертної оцінки;  $P_{ij} = 0, 1, 2, \dots$  - кількість даних компонентів в структурі  $j$ -го АЦП.

У табл. 2.4 приведені компоненти досліджуваних типів АЦП, згідно аналітичні розрахунків та вагові коефіцієнтів компонентів [34].

Таблиця 2.4 - вагові коефіцієнти апаратної складності компонентів АЦП

Pj	Компонент АЦП	Позначення	Ваговий коефіцієнт	
1	2	3	4	
1	Комунікаційний зв'язок	Z	0,001	0,001
2	Вентиль	V	0,1	0,1
3	Логічний елемент	ЛЕ	0,2	0,2
4	Прецизійний резистор	R	1	1
5	D-тригер	T	0,5	0,5
6	Компаратор	K	5	5
7	Генератор тактових імпульсів	Г	5	5
8	Шифратор	Ш	k	8
9	Пристрій управління(синхронізатор)	ПУ	20	20
10	Цифро-аналоговий перетворювач	ЦАП	3k	24
11	Лічильник Радемахера асинхронний)	ЛРа	$k(T+V+2Z)$	4,8
12	Лічильник Радемахера (синхронний)	ЛРс	$k(T+V+Z+kZ)$	4,872
13	Лічильник Крестенсона (модульний)	ЛС	$k(T+V+Z+kZ)+V+kZ$	4,98
14	Лічильник Джонсона у базисі Хаара	ЛД	$k(T+Z)+Z$	4,009

На основі аналітичного виразу (2.1), вагових коефіцієнтів представлених у табл. 2.4 та таблиці інцидентності апаратної складності АЦП (табл 2.5) отримані характеристики апаратної складності АЦП типу  $S_1$ -  $S_7$ .



Таблиця 2.5 - Таблиця інцидентності апаратної складності АЦП [34]

Компоненти АЦП	Типи АЦП						
	АЦП розгортуючого типу (U-унітарний)	АЦП розгортуючого типу (H – Хаара)	АЦП розгортуючого типу (C – Крестенсона)	АЦП розгортуючого типу паралельний (H – Хаара)	АЦП порозрядного зрівноваження (R – Радемахера)	Паралельний АЦП (R – Радемахера)	Паралельний АЦП (H – Хаара)
Z	K+7	K+5	K+7	K+8	K+8	$5 \cdot 2^k$	$6 \cdot 2^k$
V	-	-	1	-	-	$2^k$	$2^k$
ЛЕ	1	-	-	1	-	-	-
R	-	-	-	-	-	$2^k$	$2^k$
T	-	-	-	-	-	-	-
K	1	1	1	1	1	$2^k$	$2^k$
БР	-	-	-	-	1	1	-
Г	1	1	1	1	1	-	-
ЦК	-	-	-	-	-	-	-
Ш	-	-	-	-	-	1	-
ПУ	-	-	-	-	1	-	-
ЦАП	1	1	1	1	1	-	-
ЛRa	-	-	-	-	-	-	-
ЛRc	1	1	1	1	-	-	-
ЛC	-	-	1	-	-	-	-
ЛG	-	-	-	-	-	-	-
ЛD	-	-	-	1	-	-	-
K	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
2	16,55	16,44	42,83	27,79	37,13	27,62	24,64
4	42,89	42,78	75,55	45,17	44,17	104,48	98,56
8	75,81	75,7	141,47	80,17	70,25	1560,96	1553,92
12	121,39	121,28	208,03	115,49	180,29	25202,4	25231,36

Аналіз результатів розрахунків (табл 2.5) показує що АЦП розгортуючого типу  $S_1$ -  $S_5$  характеризуються низькою апаратною складністю оскільки в їх структурі використовується обмежене число компараторів, а АЦП паралельного типу мають апаратну складність на 2 порядки вищу за рахунок великого числа компараторів та інших компонентів.

Враховуючи необхідність застосування швидкодіючих АЦП у якості компонентів спектрального спецпроцесорів задача зменшення апаратної

складності АЦП є актуальною з метою їх спрощення при збереженні їх високої швидкодії.

2.2.2 Розробка методу побудови структури швидкодіючого аналого-цифрового перетворювача Хаара-Крестенсона на основі компараторів із парафазними виходами

В основу побудови швидкодіючого АЦП паралельного типу реалізовано у базисі Хаара-Крестенсона покладений метод запропонований у роботі [122], який характеризується глибоким розпаралеленням операцій перетворення аналогових сигналів у цифровий код згідно наступного графа (рис 2.3).

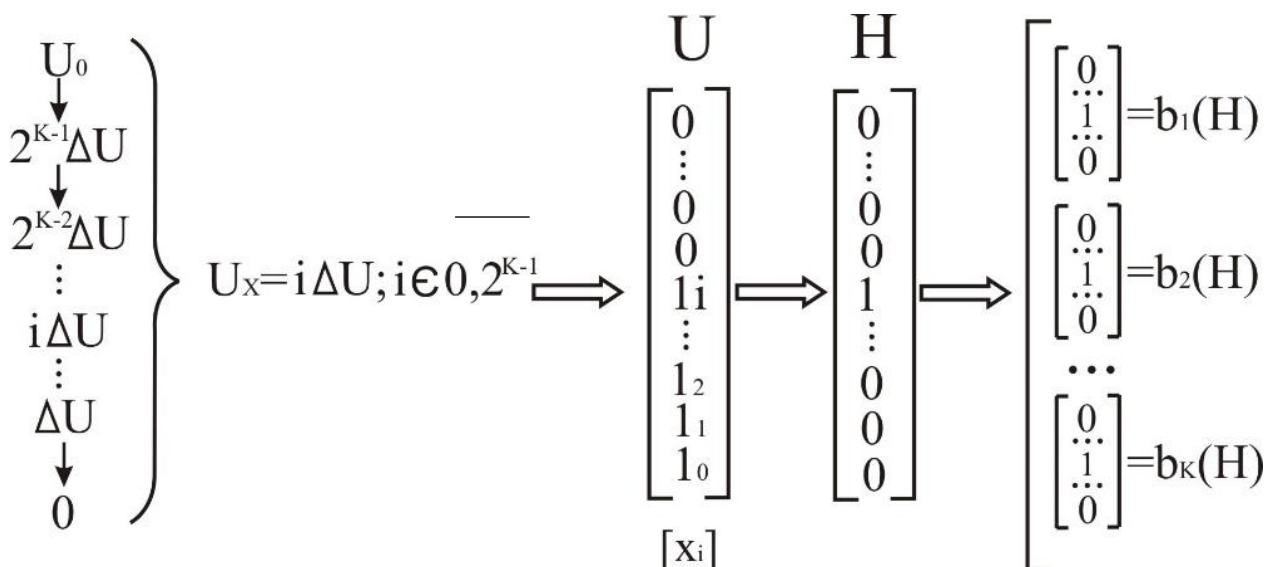


Рисунок 2.3 - Граф мультибазисного методу аналого-цифрового перетворення в паралельний код Хаара-Крестенсона

У процесі перетворення вхідного потенціалу  $U_x$  виконується його порівняння з  $i$ -тим квантованим значенням  $\Delta U$ :  $U_x = i\Delta U; i \in \overline{0, 2^k-1}$ , де  $k$  – розрядність АЦП. В подальшому на виході компараторів формується паралельний унітарний код, який на виході логічних елементів Виключаюче АБО перетворюється у паралельний код Хаара, який на виході вентиляльної матриці перетворюється у код Хаара-Крестенсона.

На рис. 2.4 показана відома структура такого класу АЦП.

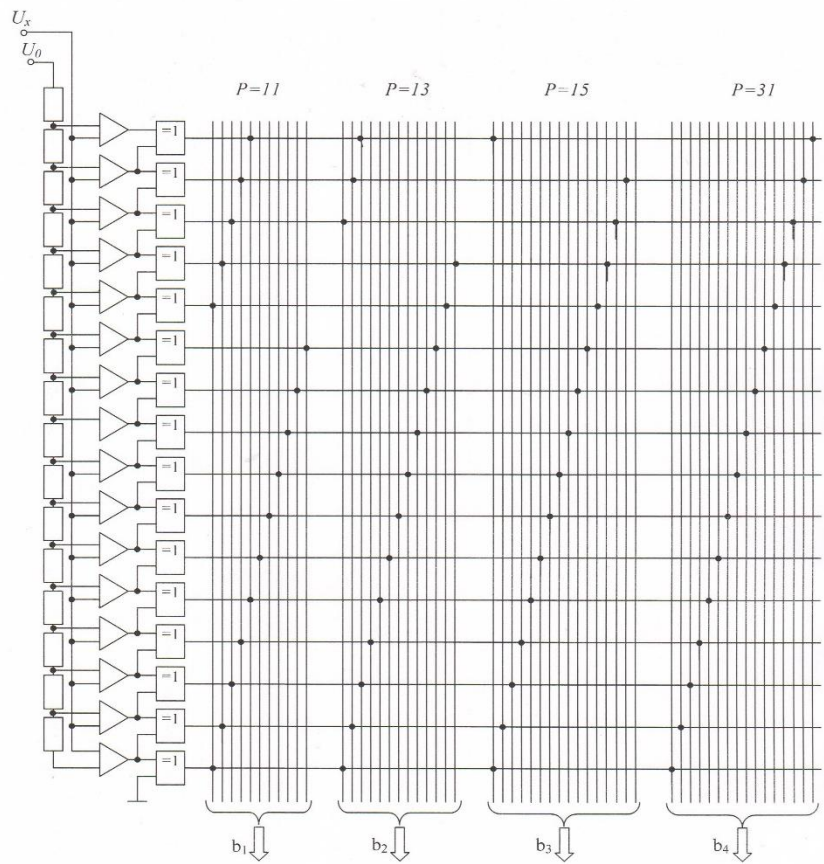


Рисунок 2.4 - Структура аналого-цифрового перетворювача з матричним шифратором для формування вихідних кодів Хаара-Крестенсона

Недоліком такої структури АЦП є підвищена апаратна та часова складність. Апаратна складність розраховується згідно виразу:  $A=2^{k-1}(R+K+5LE+4V)$  при  $k=4$ ,  $A=240v$ . Відповідно часова складність розраховується згідно виразу  $\tau_{\text{АЦП}} = \tau_K + \tau_{\text{ЛЕ}} + \tau_V$ , де  $\tau_K = 5v$  - тривалість переключення компаратора,  $\tau_{\text{ЛЕ}} = 3v$  - тривалість переключення елемента виключаюче АБО,  $\tau_V = 1v$  тривалість переключення електронного вентиля  $v$ , тобто  $\tau_{\text{АЦП}} = 9v$ .

У даному АЦП застосовані звичайні однофазні компаратори (рис. 2.5).

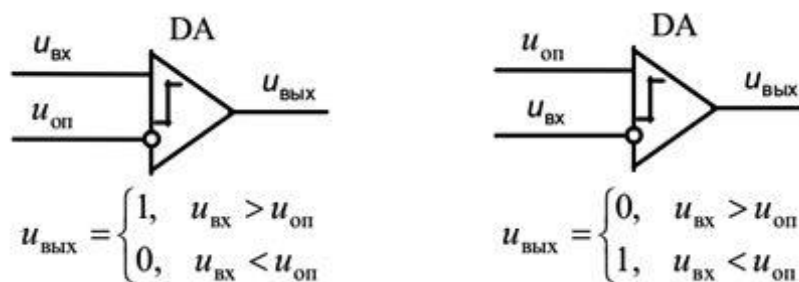


Рисунок 2.5 - Функціональна схема компаратора з однофазним виходом

При такій кількості входів і вхідних сигналів можливі два варіанти. У першому випадку, показаному в лівій частині рисунка, опорна напруга подається на інверсний вхід а вхідний - на прямий. Якщо при цьому вхідна напруга перевищить опорну, то на виході компаратора з'явиться логічна одиниця. В іншому випадку формується логічний нуль.

У другому варіанті, показаному в правій частині рисунка, опорний потенціал подається на прямий вхід, а вхідний - на інверсний. В цьому випадку, якщо вхідний потенціал більший ніж опорний то на виході компаратора логічний нуль, в іншому випадку - логічна одиниця.

Розвиток сучасної мікроелектронної технології та схемотехніки в напрямку підвищення швидкодії та розширення функціональних можливостей компараторів, які відображені у роботах [53] дозволяє обґрунтувати вибір компараторів на камоп транзисторах з тривалістю переключення до  $\tau_K = 2\tau$  а також наявністю вихідних прямих та інверсних парафазних сигналів (рис. 2.6) .

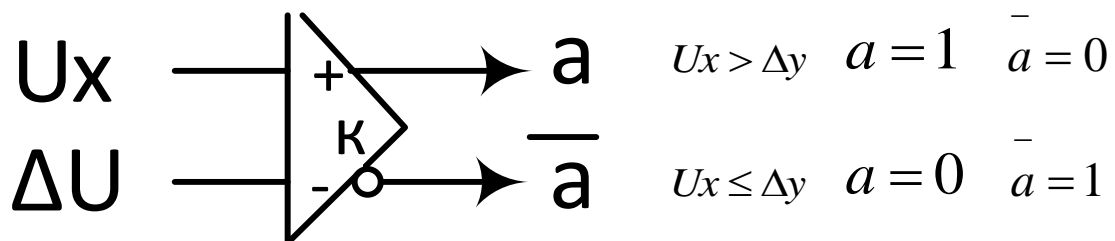


Рисунок 2.6 - Функціональна схема компаратора з парафазними виходами

На рис 2.6  $a$  та  $\bar{a}$  – відповідно прямий та інверсний виходи компаратора.

Сучасні компаратори також мають імпульсний вхід: порівняння вхідних сигналів відбувається тільки в момент подачі відповідного імпульсу.

Компаратор показаний на рис. 2.6 має парафазні виходи, як у тригера:  $a$  - верхній вихід прямий,  $\bar{a}$  - вихід інверсний.

Тривалість стрибка імпульсу (його фронту) повинна бути достатньою для того, щоб вхідний сигнал встиг пройти через декодер до того, як встигне спрацювати елемент пам'яті. Застосування імпульсів збільшує заводозахищеність

компаратора, оскільки перешкода може змінити стан компаратора лише в короткий час імпульсу.

Іншим компонентом даного типу АЦП, який має затримку сигналів  $\tau_{ЛЕ} = 3-5 \nu$  є класична мікроелектронна реалізація логічного елемента Виключаюче АБО, яка показана на рис. 2.7.

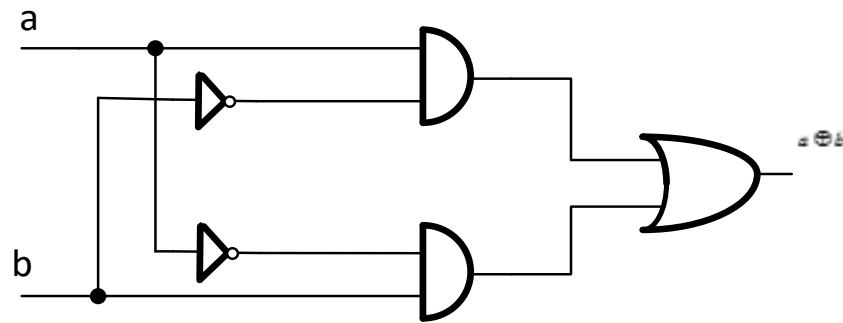


Рисунок 2.7 - Структура елемента виключаюче АБО на логічних елементах І, АБО, НЕ

Аналіз часової та апаратної складності даного елемента показує, що часова складність  $\tau_{ЛЕ} = 3\nu$ ; а апаратна складність  $A=5\nu$ ;

Системні характеристики елемента Виключаюче АБО можна значно покращити при реалізації на основі вхідних парафазних сигналів [130] та об'єднання виходів логічних елементів І-НЕ (рис. 2.8).

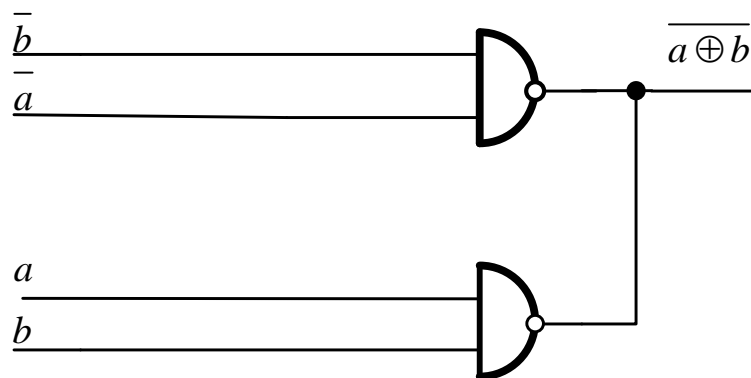


Рисунок 2.8 - Структура логічного елемента Виключаюче АБО на основі вхідних парафазних сигналів та об'єднання інверсних виходів елементів І-НЕ.

У результаті застосування компараторів з парафазними виходами та реалізації логічного елемента Виключаюче АБО на логічних елементах І-НЕ з парафазними входами які з'єднані між собою інверсними виходами отримуємо компонент АЦП, який відповідає одному рівню квантування згідно наступної структури (рис. 2.9).

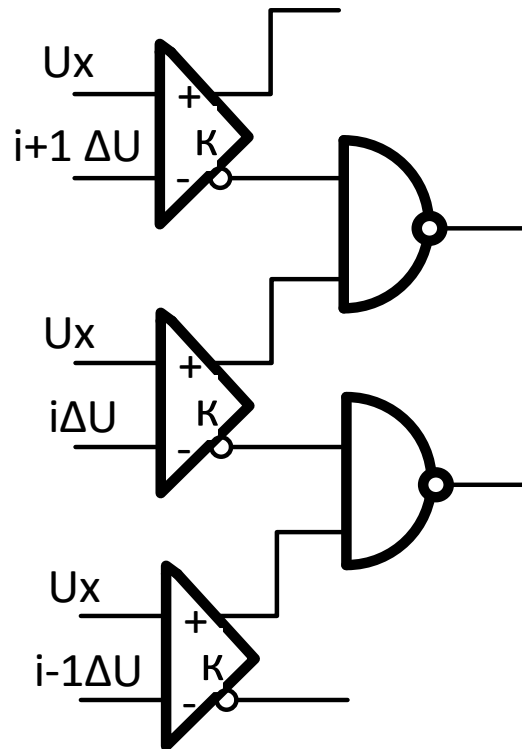


Рисунок 2.9 - Структура компонента I-того рівня квантування АЦП Хаара-Крестенсона на основі міжрівневих парафазних зав'язків

При такій реалізації компонента АЦП Виключаюче АБО підвищується швидкодія АЦП з часовою складністю  $\tau_{\text{АЦП}} = 7v$  що дозволяє підвищити швидкодію АЦП у 2 рази, а апаратну складність, яка рівна  $A=208$ , що відповідає зменшенню апаратної складності у 3 рази.

Реалізація компонента АЦП на рис. 2.8 перетворювача паралельного унітарного коду у код Хаара є універсальною і відповідає симетричній схемі перетворення тому є структурно надлишковою.

Запропонована реалізація компонента перетворювача унітарного коду у паралельний код Хаара, яка враховує переключення компараторів в одиничний стан починаючи з нульової позиції на базі одного логічного елемента І-НЕ, крім того з матриці шифратора кодів базису Крестенсона вилучені вентиля, які замінюються входами багатовходових логічних елементів І-НЕ згідно структури на рис. 2.11.

Таким чином системні характеристики такої реалізації АЦП Хаара-Крестенсона з використанням новітніх компараторів з парафазними виходами та

нової реалізації Виключаюче АБО на елементах І-НЕ будуть наступні  $\tau_K = 2 \nu$  ,  $\tau_{ЛЕ} = 1 \nu$  ,  $\tau_K + \tau_{ЛЕ} = 2+1+1=4 \nu$  .

В роботі отримав подальший розвиток метод побудови швидкодіючого АЦП з вихідними кодами у базисі Хаара-Крестенсона, формалізація якого у вигляді графа приведена на рис. 2.10.

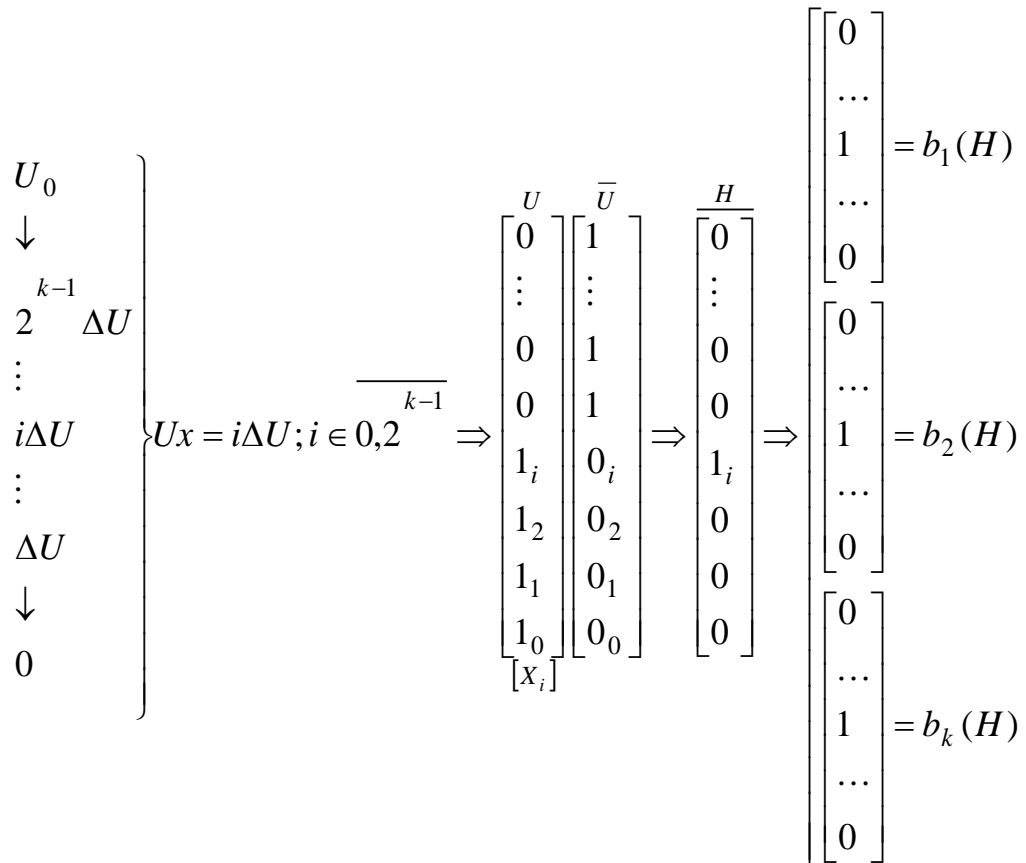


Рисунок 2.10 – Формалізація швидкодіючого АЦП з вихідними кодами у базисі Хаара-Крестенсона

Запропонований метод відрізняється тим, що у процесі перетворення аналогового сигналу  $U_x$  в паралельний унітарний код одночасно формується прямий та інверсний паралельний код унітарного ТЧБ, який в подальшому перетворюється у прямий код базису Хаара, що перетворюються у код базису Хаара-Крестенсона.

Проведений аналіз системних характеристик, розробленого АЦП Хаара-Крестенсона та його структури, яка приведена на рис. 2.5 дозволяє проаналізувати вклад різних компонентів у покращення його параметрів апаратної та часової

складності. Таким чином на рис. 2.11 показаний фрагмент розробленої структури АЦП Хаара-Крестенсона, який відповідає одному кванту амплітуди вхідних аналогових сигналів,

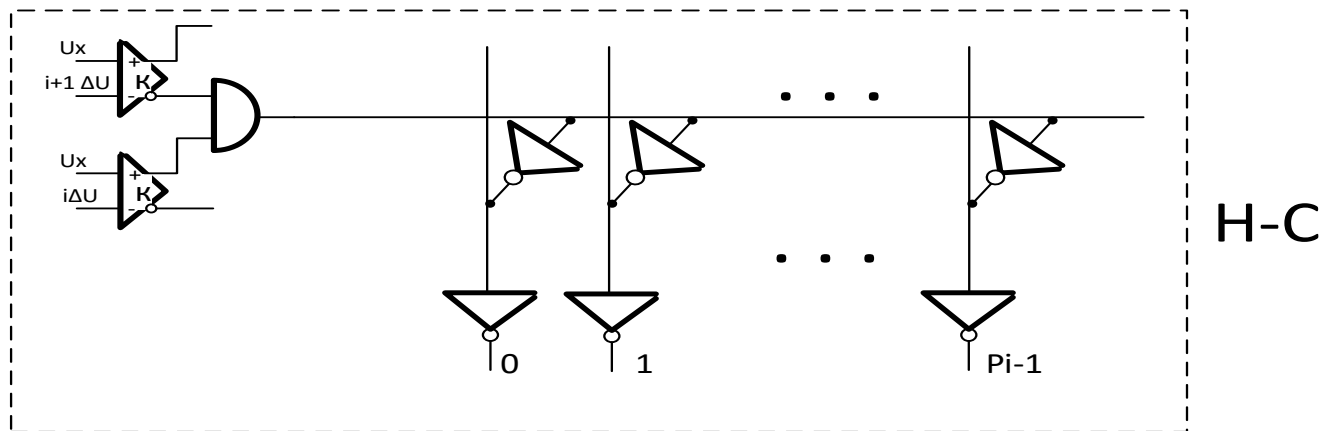


Рисунок. 2.11 - Структура АЦП з парафазними компараторами по одному рівні квантування

На основі запропонованого компонента АЦП для одного рівня квантування (рис 2.11) розраховується апаратна складність згідно виразу:

$$\tau_{\text{АЦП}} = \tau_K + \tau_{\text{ЛЕ}} + 2\tau_V;$$

де  $\tau_{\text{АЦП}}$ - сумарний час затримки сигналу на виході АЦП;

$\tau_K$  – тривалість переключення компаратора,  $\tau_K = 2\nu$  [155];

$\tau_{\text{ЛЕ}}$  – сумарний час переключення логічного елемента виключаючого АБО  $\tau_{\text{ЛЕ}} = 1\nu$ ;

$\tau_V$  – тривалість переключення вентиля, який реалізується на логічному елементі НЕ  $\tau_V = 1\nu$ ;

$\nu$  – швидкодія мікроелектронної елементної бази, на якій реалізується структура АЦП ( $\nu = 0,1; 1; 2$  нс);

$$\text{Отже } \tau_{\text{АЦП}} = 2 + 1 + 2 = 5\nu.$$

Аналіз отриманої оцінки апаратної складності суми компонентів одного рівня квантування запропонованого АЦП Хаара-Крестенсона показує можливості підвищення швидкодії за рахунок застосування компараторів з парафазними виходами та зменшення затримки сигналів у логічному елементі І у порівнянні з



логічним елементом Виключаюче АБО. Тобто підвищення швидкодії розробленого АЦП у порівнянні з відомим складає  $k=(9-12)/5= 1.8-2.4$  рази.

У структурі дешифратора відомого АЦП приведенного на рис. 2.11, який перетворює унітарний код в код Хаара-Крестеносна використовують два послідовно з'єднанні вентиля, що також знижує швидкодію даної реалізації АЦП.

Запропоноване виконання логічного елемента І-НЕ з інверсним виходом дозволяє виключити один із вентилів у структурі дешифратора, підвищити швидкодію АЦП та зменшити апаратну складність шляхом реалізації дешифратора унітарний код – код Хаара-Крестенсона застосуванням багатовходових логічних елементів І-НЕ, приклад якого показаний на рис. 2.12.

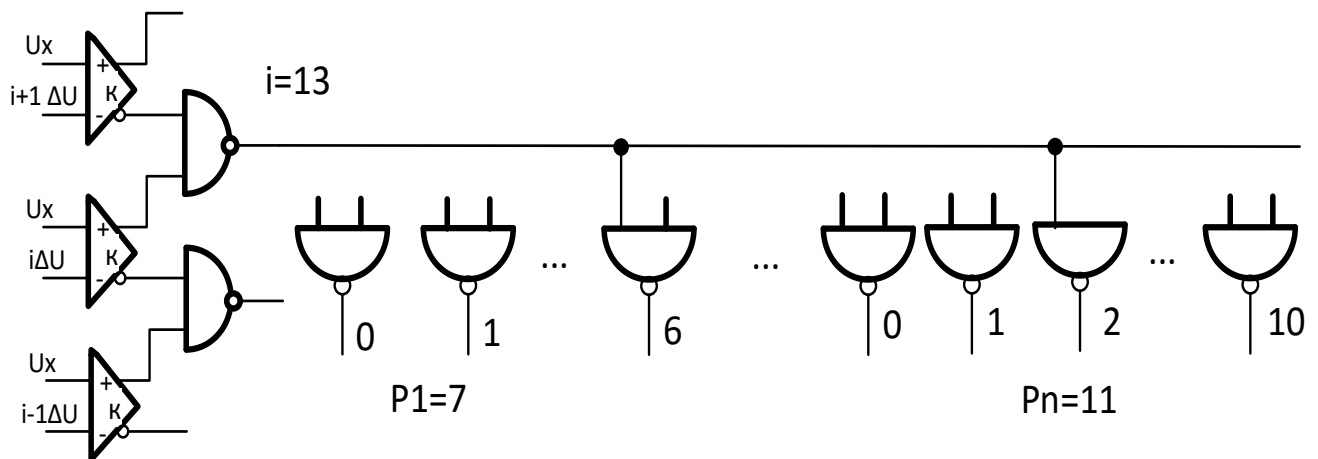


Рисунок 2.12 - Структура і-того рівня АЦП Хаара-Крестенсона з використанням елементів виключаюче АБО з інверсними виходами та багатовходових логічних елементів І-НЕ

Таким чином запропоноване вдосконалення структури і-тих компонентів АЦП Хаара-Крестенсона дозволяє підвищити його швидкодію та зменшити часову складність перетворень у 2,5-3 рази. При цьому апаратна складність і-того компонента АЦП відомого А1 та запропонованого А2 структурного рішення визначається згідно виразів:

$$A_i = A_k + A_{\text{ЛЕ}} + A_{\text{ш}}; \quad (2.2)$$

$$A_1 = 1 + 5 + 2 = 8;$$

$$A_2 = 2 + 1 + 1 = 4.$$

Тобто зменшення апаратної складності запропонованого схемотехнічного рішення і-того компонента АЦП Хаара-Крестенсона досягнуто у 2 рази.

Функціональна схема запропонованого схемотехнічного рішення швидкодіючого АЦП зображена на рисунку 2.13.

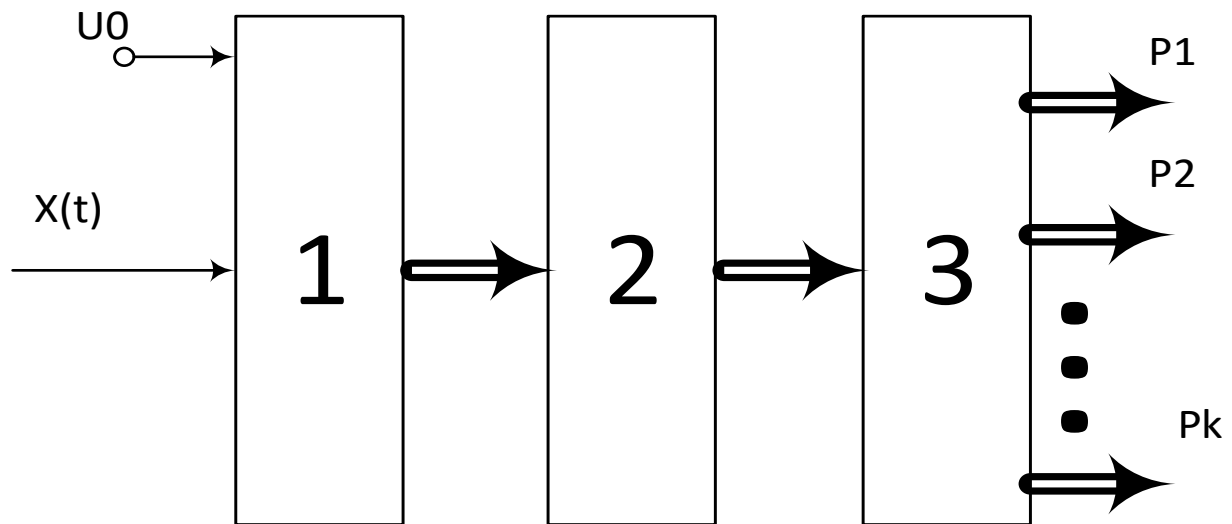


Рисунок 2.13 – Функціональна схема АЦП з вихідними кодами у базисі Хаара-Крестенсона

$x(t)$ - вхідний аналоговий сигнал,  $U_0$ - взірцевий потенціал, 1- лінійки взірцевих резисторів та парафазних компараторів, 2 – лінійка логічних елементів І-НЕ, 3 – група багатофазових логічних елементів І-НЕ, - 4.

При такій реалізації компонентів АЦП підвищується регулярність структури та швидкодія, що дозволяє його ефективно застосувати у якості базового компонента швидкодіючого спектрального косинусного претворення Фурє.

При цьому отримані вихідні цифрові коди у ТЧБ Хаара-Крестенсона у системі взаємопростих модулів  $P_1, P_2-P_k$  можуть безпосередньо використовуватися для реалізації швидкодіючих матрично модульних операцій множення та додавання у спектральному спецпроцесорі з часовою складністю 2 мікротакти.

На рис. 2.14 показана порівняльна характеристика часової складності реалізацій АЦП запропонованої структури АЦП Хаара-Крестенсона у залежності від розрядності. Розрахунки виконані на основі виразу:

$$L = \tau K + \tau L_E + \tau V . \quad (2.3)$$

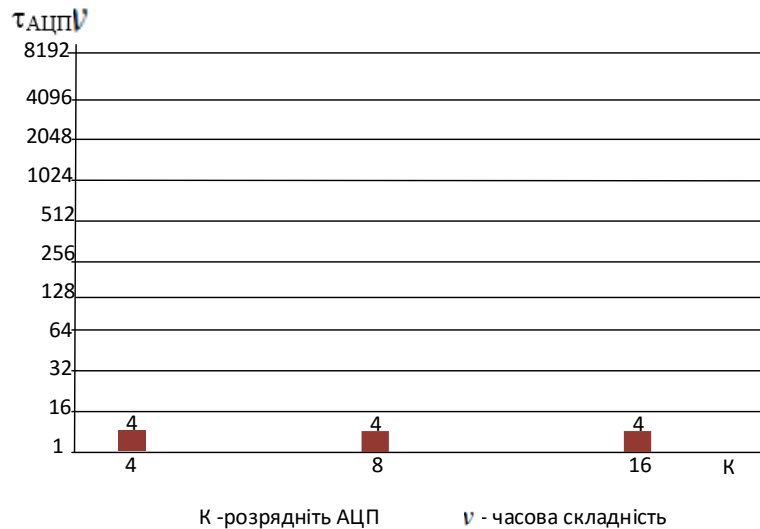


Рисунок 2.14 – Порівняльна характеристика часової складності АЦП Хаара Крестенсона у залежності від розрядності.

На рис. 2.15 показано порівняльну характеристику апаратної складності відомих реалізацій АЦП у базисі Радемахера та Хаара-Крестенсона на основі однофазних компараторів елементів виключаюче АБО у порівнянні з апаратною складністю запропонованої структури АЦП на основі парафазних компараторів та логічних елементів І-НЕ у залежності від їх розрядності. Розрахунки виконані на основі виразу:  $A=2k-1 (R+K+5LE+4V)$ .

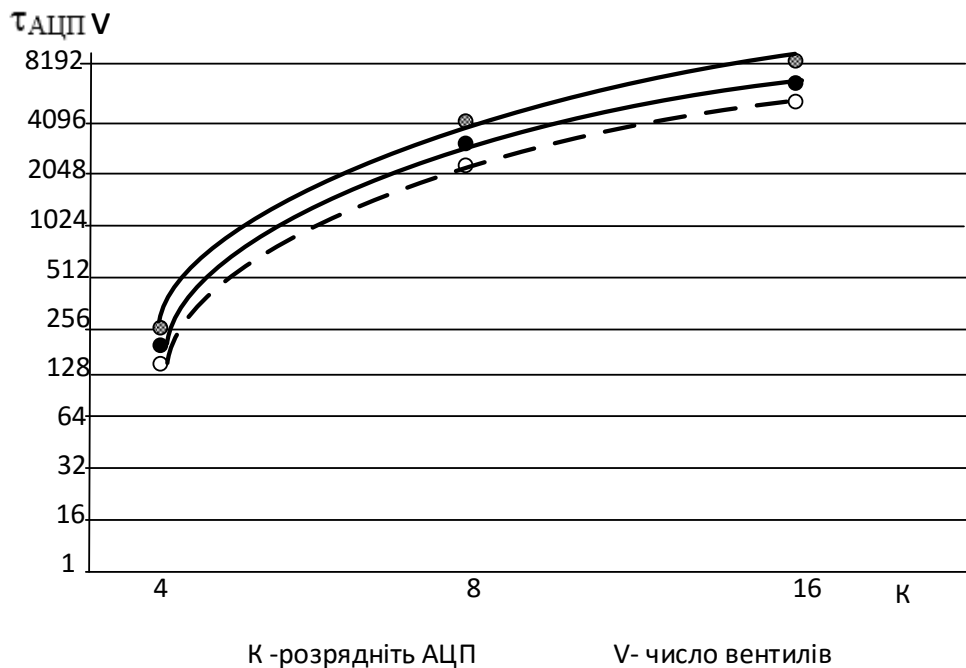


Рисунок 2.15 – Порівняльна характеристика апаратної складності АЦП Хаара Крестенсона.

Проведені дослідження обґрунтовують доцільність реалізації АЦП у базисі Хаара-Крестенсона на основі запропонованих компонентів для реалізації та його використання у якості базового компонента спектрального спецпроцесора косинусного перетворення Фур'є.

### 2.2.3. Дослідження часової та апаратної складності реалізації багаторозрядних регістрів зсуву у базисах Радемахера та Хаара-Крестенсона

У загальному випадку при побудови спецпроцесорів кореляційного опрацювання сигналів БРЗ виконує буферну функцію затримки цифрових відліків на певне число мікротактів, які поступають на входи відповідних перемножувачів де відбувається паралельне перемноження зсунутих у часі цифрових відліків з текучими відліками, які формуються на виході АЦП.

Функціональна структура формування вихідних кодів в БРЗ приведена на рис. 2.16 .

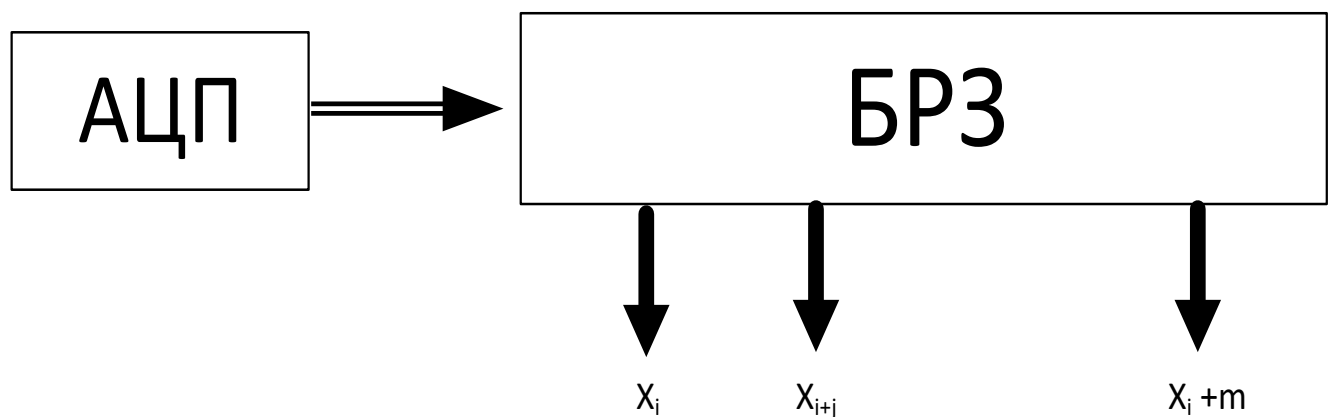


Рисунок 2.16 - Функціональна структура формування вихідних кодів БРЗ

Як було показано у попередньому розділі, на виході АЦП у реальному часі з кроком дискретизації  $\Delta T$  можуть формуватися паралельні коди цифрових відліків у різних ТЧБ, які забезпечують високу швидкодію опрацювання даних. Аналіз табл. 2.6 показує, що в залежності від використовуваного ТЧБ і заданому числі запам'ятованих цифрових відліків  $x_i$  об'єм необхідної пам'яті розраховується згідно аналітичних виразів представлених табл. 2.7.

Таблиця 2.6 – паралельний та послідовний код базису Радемахера на виході АЦП у реальному часі

$x_i \Rightarrow$	$a_{k-1} \ a_{k-2} \ \dots \ a_1 \ a_0$	Послідовний код базису Радемахера
$x_i \Rightarrow$	$a_{k-1}$ $\dots$ $a_j$ $\dots$ $a_0$	Паралельний код базису Радемахера

У випадку застосування знакового корелятора в якому число рівнів квантування = 1 і додатково реєструється біт знаку + - об'єм пам'яті такого БРЗ складає  $2N$ . Структурні схеми БРЗ у різних ТЧБ реалізовані на D-тригерах показані на рис. 2.17, 2.18, 2.19.

Таблиця 2.7 – Структура вихідних кодів БРЗ у різних ТЧБ

Паралельні коди ТЧБ		Вихідний код АЦП	Характеристика пам'яті БРЗ $V_{БРЗ}$
Унітарний	У	$b_{N-1}$ $\dots$ $b_j$ ; де $a_1 = 1; b_j = 0$ $\dots$ $a_1$ $a_2$	$N \cdot m$
Хаара	Н	$b_{N-1}$ $\dots$ $a_j$ ; де $a_1 = 1; b_j = 0$ $\dots$ $b_0$	$N \cdot m$
Радемахера	Р	$\begin{pmatrix} a_{k-1} \\ \dots \\ a_j \\ \dots \\ a_0 \end{pmatrix}$ ; $a_j \in \bar{0,1}$	$\log_2 N \cdot m$

## Продовження таблиці 2.7

Радемахера-Крестенсона	R - C	$a_k$ $\dots$ $a_j$ ; $0 \leq a_j \leq p_j$ $\dots$ $a_1$	$\sum_{j=1}^n \log_2 p_j$
Хаара-Крестенсона	H - C	$a_k$ $b_{p-1}$ $\dots$ $\dots$ $a_j$ $a_j = 1; b_j = 0$ $\dots$ $\dots$ $a_1$ $b_0$	$\sum_{j=1}^n p_j$

Розрахунок системних характеристик БРЗ, які можуть бути реалізовані у різних ТЧБ приведені в табл. 2.8

Таблиця 2.8 Розрахунок системних характеристик БРЗ у різних ТЧБ

характеристики БРЗ	Число розрядів	Швидкодія	Число каналів вихідних шин
паралельними кодами у базисі Радемахера	K	V=v	N <sub>K</sub> =K·m
з кодами у базисі Хаара	2 <sup>K</sup>	V=2ЛЕ	N <sub>K</sub> =2 <sup>K</sup> ·m
знакового	K=2	V=2ЛЕ	N <sub>K</sub> =2·m
релейного	K=K+1	V=2ЛЕ	N <sub>K</sub> =(K+1)·m
цифрових кореляторів у базисі Крестенсона	$K = 2K + \hat{E}[\log_2 n]$	V=2ЛЕ	N <sub>K</sub> =K·m

На рис. 2.17 показано структуру багатоканального регістру зсуву у базисі Радемахера.

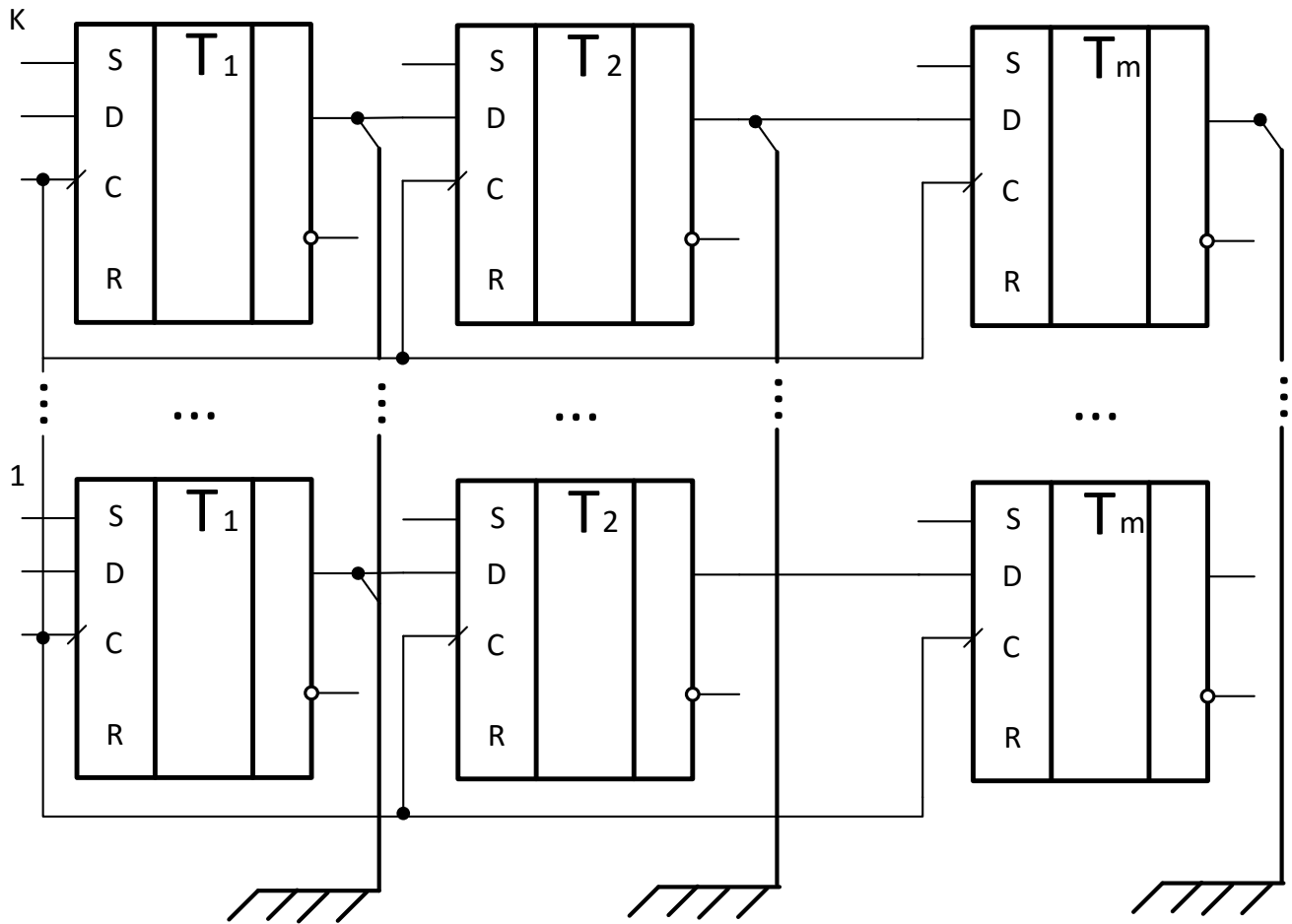


Рисунок 2.17 - Багатоканальний регістр зсуву у базисі Радемахера

Системні характеристики такого БРЗ розраховується згідно виразів:

1) апаратна складність:

$$A = m \cdot K, \quad (2.2)$$

де  $m$  – число точок затриманих цифрових відліків,  $K$  – розрядність АЦП у базисі Радемахера.

2) Часова складність  $\tau_{БРЗ} = \tau_{ТД} = 2\tau$ , тобто незалежить від розрядності АЦП  $k$  та числа затриманих цифрових відліків  $m$ .

При застосуванні на вході спектрального спецпроцесора швидкодіючого АЦП в кодах Радемахера-Крестенсона структура БРЗ представлена на рис. 2.18

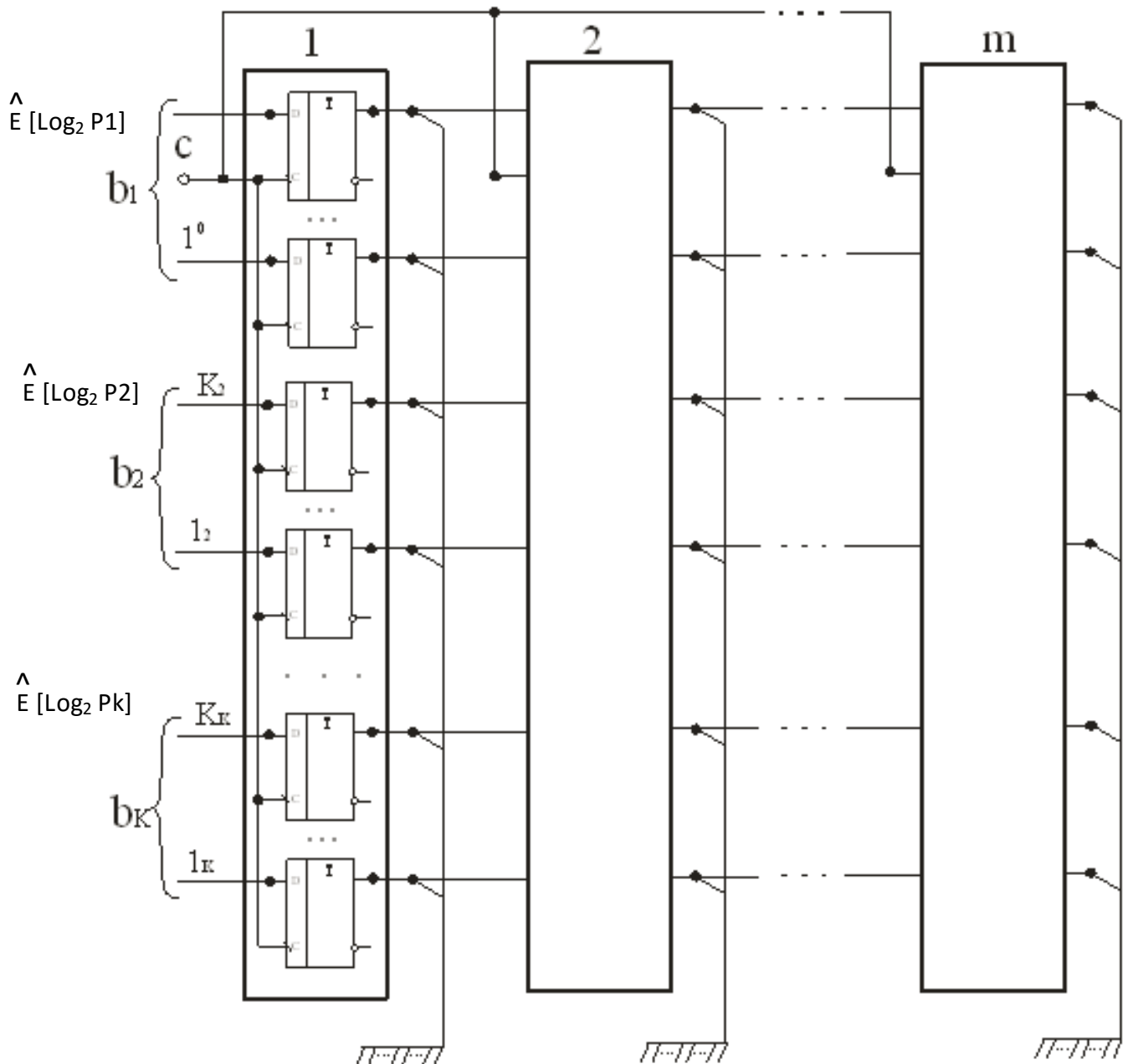


Рисунок 2.18 - Структура БРЗ спектрального спецпроцесора при використанні АЦП Радемахера- Крестенсона

Часова складність такого БРЗ також складає  $2\nu$ , а апаратна складність розраховується згідно виразу:

$$A = m \cdot \sum_{i=1}^k \hat{E}[\log_2 P_i]. \quad (2.3)$$

Враховуючи, що розрядність добутку модулів  $P_1 P_2 \dots P_k$  перевищує відповідну розрядність двійкового коду базису Радемахера на одиницю то значення  $A$  буде рівне  $m \cdot (K+1)$ .



При застосуванні запропонованого швидкодіючого АЦП у базисі Хаара-Крестенсона структура БРЗ має вид представлений на рис. 2.19.

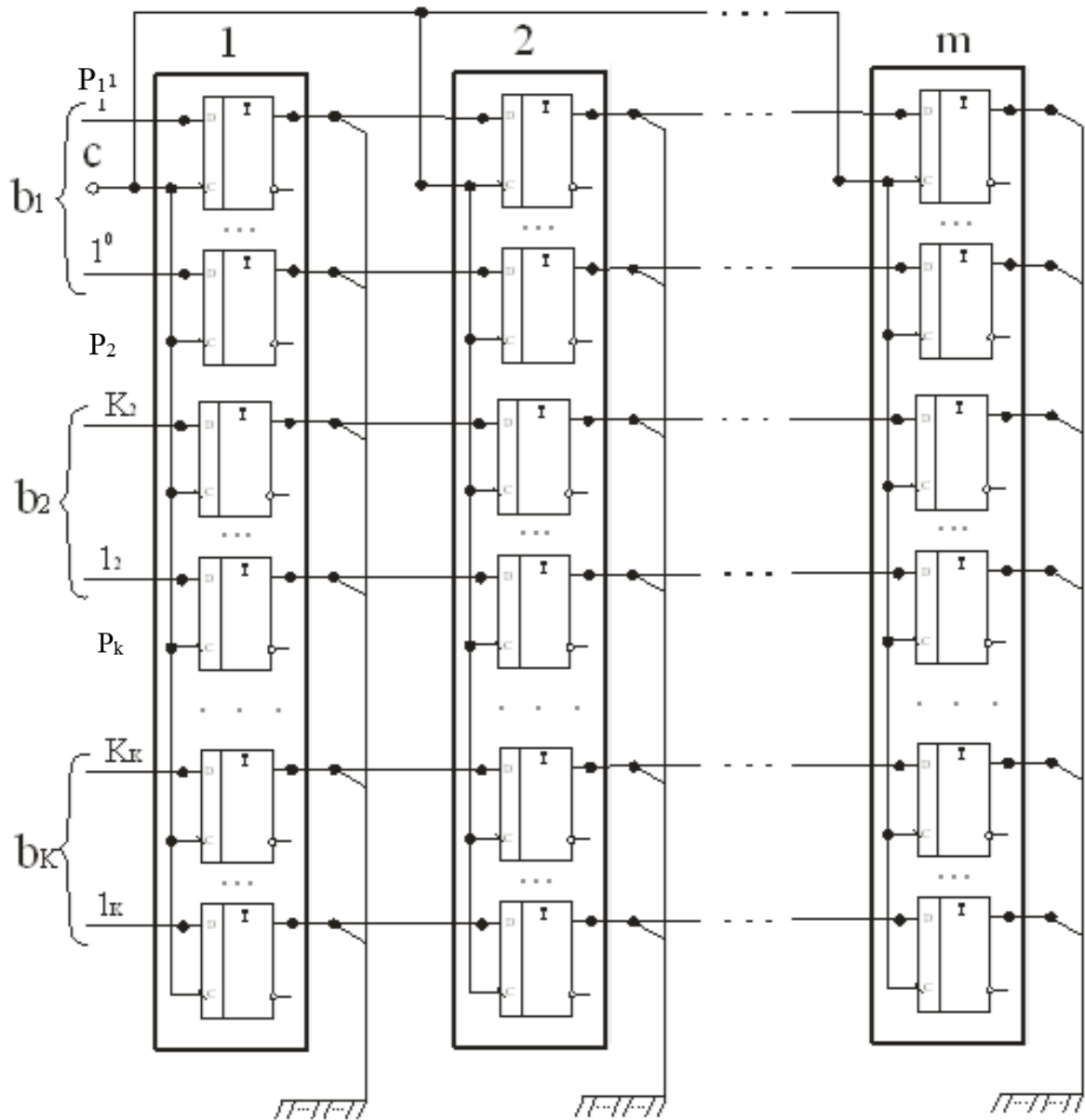


Рисунок 2.19 - структура БРЗ спектрального спецпроцесора при застосуванні АЦП Хаара-Крестенсона

Аналогічно з характеристиками попередньої структури часова складність складає  $2\nu$ , а апаратна складність розраховується згідно виразу:

$$A = m \cdot \sum_{i=1}^k P_i. \quad (2.4)$$

Таблиця 2.8 – Апаратна та часова складність БРЗ у різних ТЧБ

розрядність	Радемахера		Радемахера-Крестенсона		Хаара- Крестенсона	
	Апаратна складність (число вентелів)	Часова складність ( $\nu$ )	Апаратна складність (число вентелів)	Часова складність ( $\nu$ )	Апаратна складність (число вентелів)	Часова складність ( $\nu$ )
2	320	2	480	2	800	2
4	640	2	800	2	1600	2
8	1280	2	1440	2	3040	2
16	2560	2	2720	2	3840	2

При розрахунку характеристик БРЗ прийнято:

Часова складність D- тригера  $5(\nu)$ , часова складність БРЗ у різних ТЧБ  $= 2(\nu)$ .  
 Апаратна складність БРЗ для різних ТЧБ у залежності від розрядності АЦП  $K=2, 4, 8$  та числа зсувів  $m=32$  розраховується згідно виразів (2.2), (2.3), (2.4).

На рис. 2.20 приведені діаграми апаратної складності БРЗ при застосуванні у якості компонентів різних структур мультибазисних АЦП.

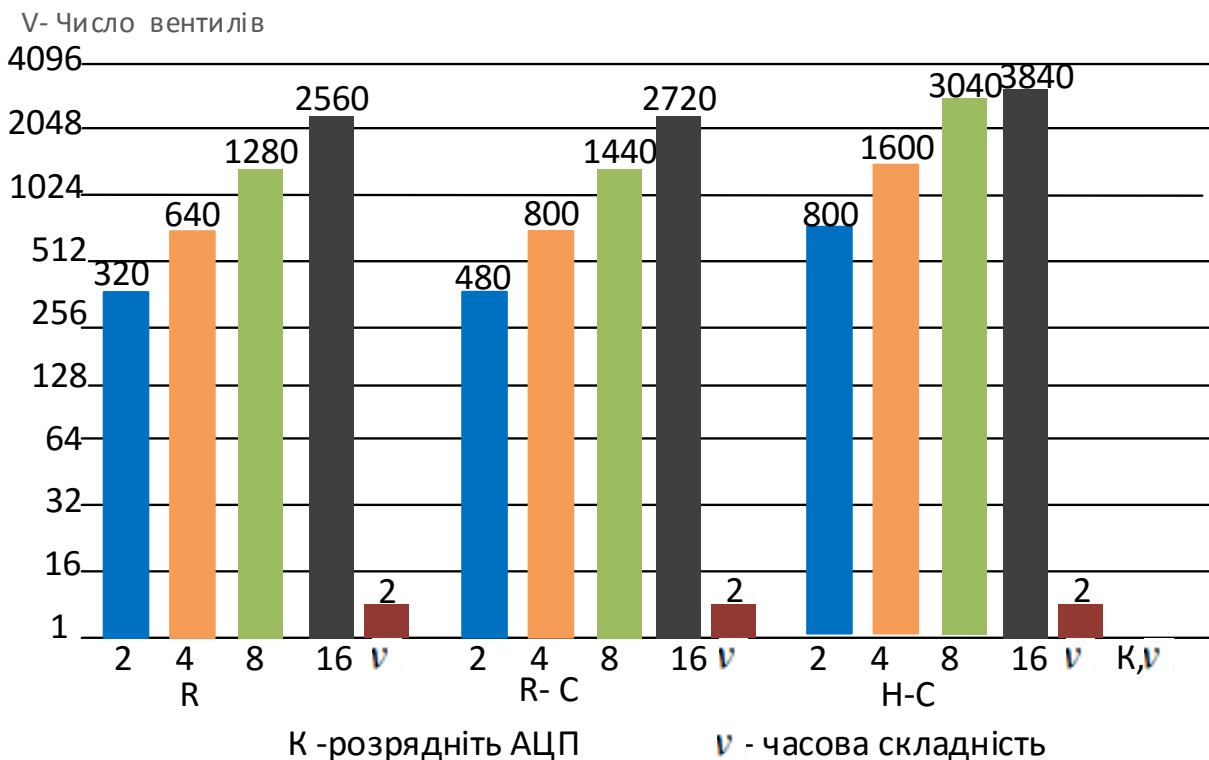


Рисунок 2.20 - Порівняльні діаграми апаратної складності БРЗ у різних ТЧБ

### 2.3 Структура та системні характеристики операційних пристроїв модульного матричного перемноження кодів у різних теоретико-числових базисах

Як показано у першому розділі алгоритм виконання спектрального косинусного перетворення Фур'є включає операції перемноження цифрованих даних, які в розрядах автокореляційних обчислень є базовими компонентами спектрального процесора. Тому метод виконання операцій множення та реалізація відповідних операційних пристроїв є важливою задачею, успішне рішення якої суттєво впливає на швидкодію реалізації спектрального перетворення Фур'є. Таким чином доцільним є дослідження різних методів реалізації перемножувачів цифрових даних.

Аналіз літературних даних показує, що операція множення може виконуватися на основі різних алгоритмів. В роботах [118] показано, що операція множення двох однакових цифрових кодів, тобто піднесення до квадрату, реалізується шляхом сумування непарних чисел згідно виразу:

$$x \cdot x = \sum_{i=1}^x (2i - 1).$$

Наприклад:  $x=9$ ;  $x \cdot x = 1+3+5+7+9+11+13+15+17=81$ .

У випадку коли перемножуються різні числа обчислення добутку виконується згідно аналітичного виразу [116-117]:

$$x \cdot y = \frac{(x + y)^2 - (x - y)^2}{4}. \quad (2.5).$$

Особливістю реалізації множення згідно виразу (2.5) є виконання тільки операцій піднесення до квадрату та різниці отриманих квадратів. При цьому якщо виконання операцій множення здійснюється у двійковій системі числення базису Радемахера то операція ділення на число у знаменнику 4 виконується простим відкидання трьох молодших розрядів отриманого двійкового числа.

На рис.2.21 показана структура пристрою піднесення до квадрату, яка містить двійковий лічильник  $L$  та блок накопичуючого сумування  $\Sigma$ .

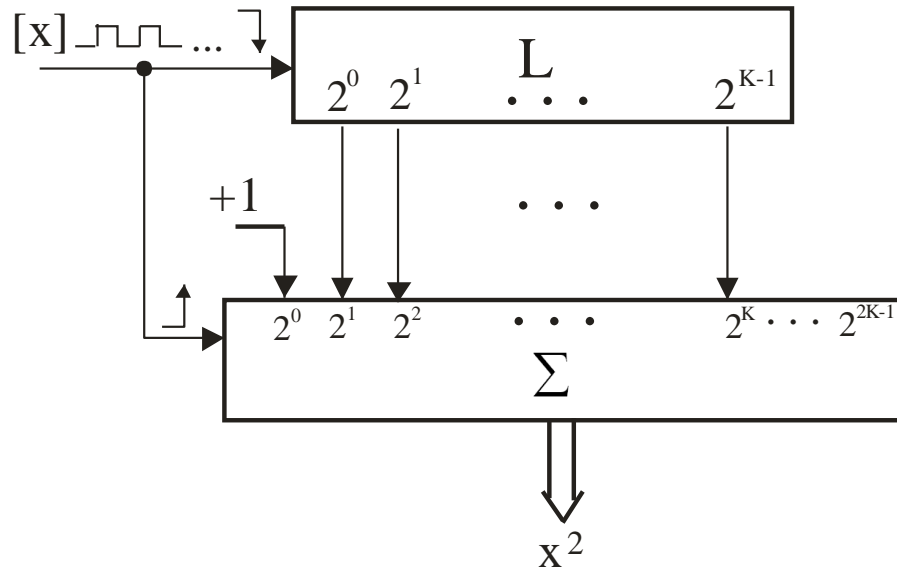


Рисунок 2.21 - Структура перемножувача шляхом піднесення до квадрату унітарних кодів ( $L, \Sigma$  – відповідно лічильник та суматор у базисі Радемахера)

Недоліком такого компонента цифрового перемножувача є низька швидкодія обумовлена представленням вхідних даних в унітарному ТЧБ. Тобто швидкодія операції піднесення до квадрату  $n$ -розрядного двійкового числа складає  $2^{n-1}$  операцій додавання в лічильнику  $L$  та накопичуючому суматорі  $\Sigma$ . Враховуючи наявність наскрізних переносів у суматорі  $\Sigma$  швидкодія такого перемножувача не перевищує часову складність  $\tau = 2^{n-1} \cdot \log_2 2n$ , а апаратна складність розраховується згідно виразу  $A = L + \Sigma = 3n + 5 \cdot 2n + 6n$ . Таким чином встановлено, що використання перемножувачів на основі квадраторів унітарного ТЧБ є неефективним та недоцільними при реалізації перемножувачів у спектральних спецпроцесорах.

В роботі [5] запропонований алгоритм множення згідно виразу:

$$[x] \cdot [y] = [x^2] + [|y - x| \cdot x], \quad x \cdot y = \sum_{i=1}^{\check{Z}} (2i - 1) + \sum_{i=\check{Z}+1}^{\hat{Z}} \check{Z}, \quad x \leq y;$$

$$\check{Z} = \begin{cases} x, & x \leq y \\ y, & x > y \end{cases}; \quad \hat{Z} = \begin{cases} x, & x > y \\ y, & y \geq x \end{cases}, \quad (2.6)$$

в основу якого покладено піднесення до квадрату меншого з двох чисел, які перемножуються та додавання до нього добутку меншого числа на модульну різницю між числами що перемножуються. Наприклад:  $x=5, y=9$  тоді

$$x \cdot y = 5^2 + 5 |9 - 5| = 25 + 5 \cdot 4 = 45,$$

або згідно виразу (2.6) отримаємо

$$x \cdot y = 1 + 3 + 5 + 7 + 9 + 5 + 5 + 5 = 45.$$

на рис. 2.22. показано структуру яка містить однорозрядний неповний суматор  $\overset{\vee}{\Sigma}$ , n-розрядний двійковий лічильник L, мультиплексор Mp та 2n-розрядний накопичуючий суматор  $\Sigma$ .

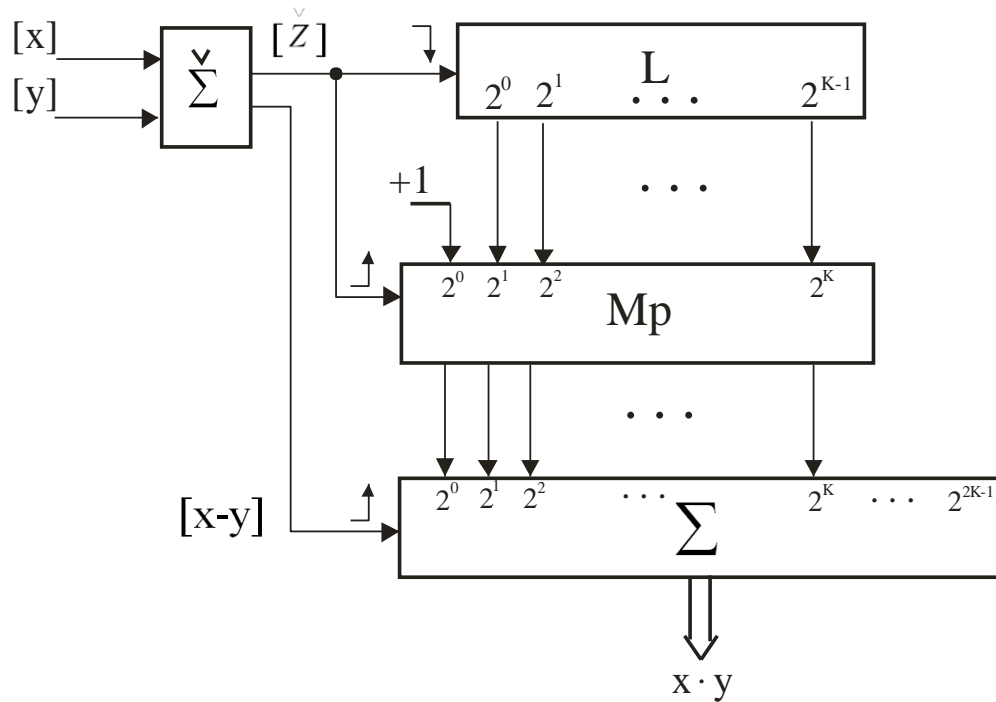


Рисунок 2.22 - Структура перемножувального пристрою унітарних кодів

Принцип роботи такого суматора полягає у тому, що на першому етапі виконується накопичення коду меншого числа в лічильнику L та додавання відповідного числа непарних чисел в накопичуючому суматорі  $\Sigma$ , на другому етапі виконується додавання двійкових кодів чисел у суматорі  $\Sigma$ , які сформовані в лічильнику L певне число разів, яке відповідає різниці між двома числами, які перемножуються.

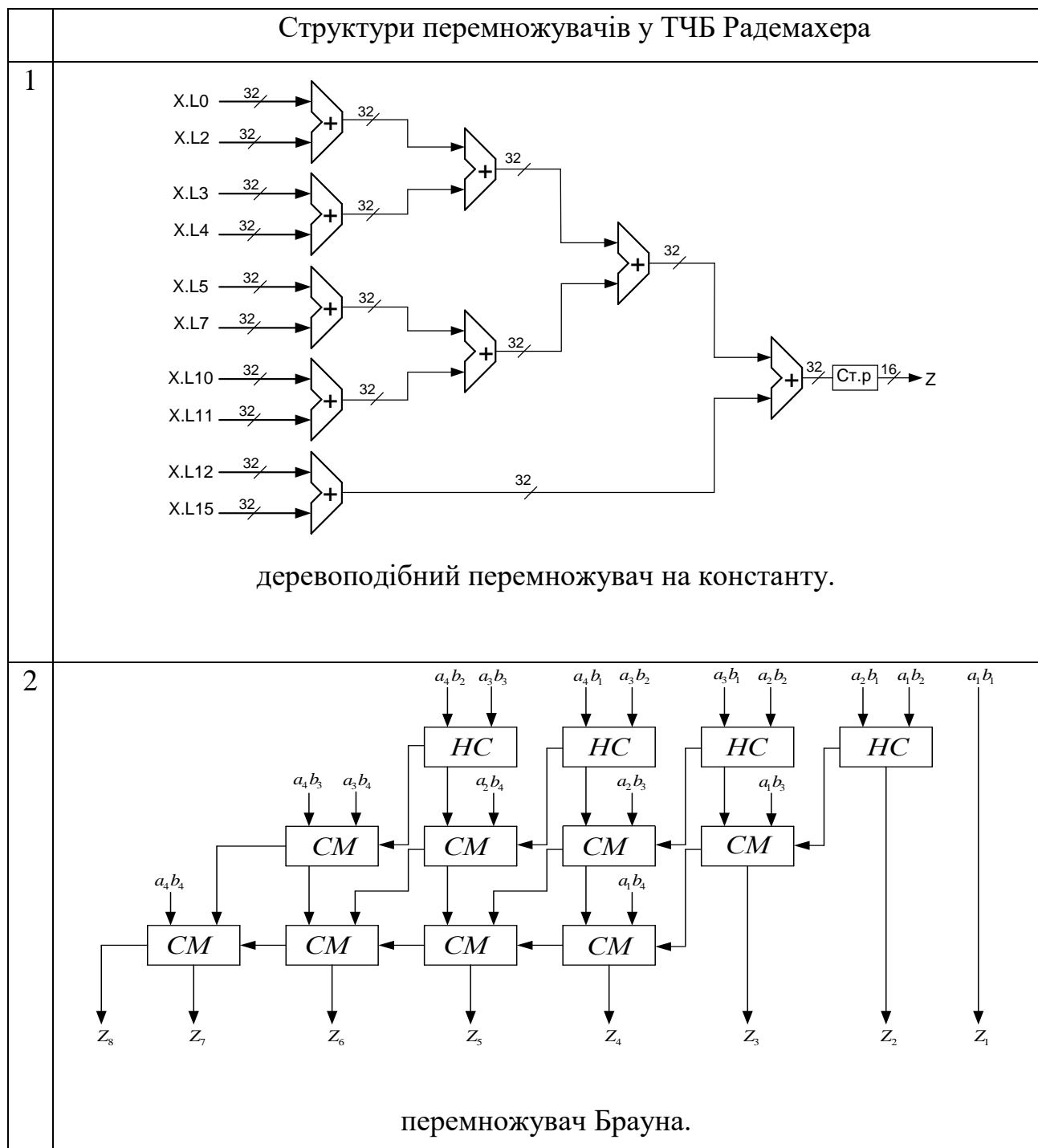
Часова складність такого перемножувача розраховується згідно виразу  $\tau_x = (\overset{\vee}{\Sigma} + L + Mp + \Sigma = 3 + 2 + 2 + 10n)$  апаратна складність  $A = (5 + 3^n + 3^n + 8^n)v$ .

Даний перемножувач також характеризується низькою швидкістю і не може бути застосований в якості компонента спектрального спецпроцесора.

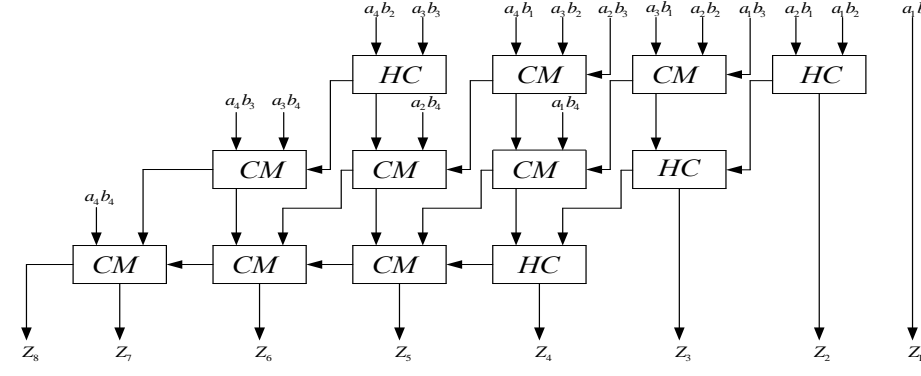
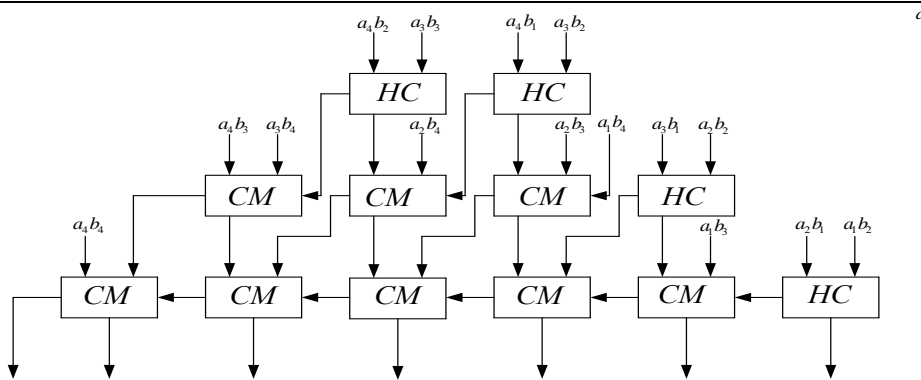
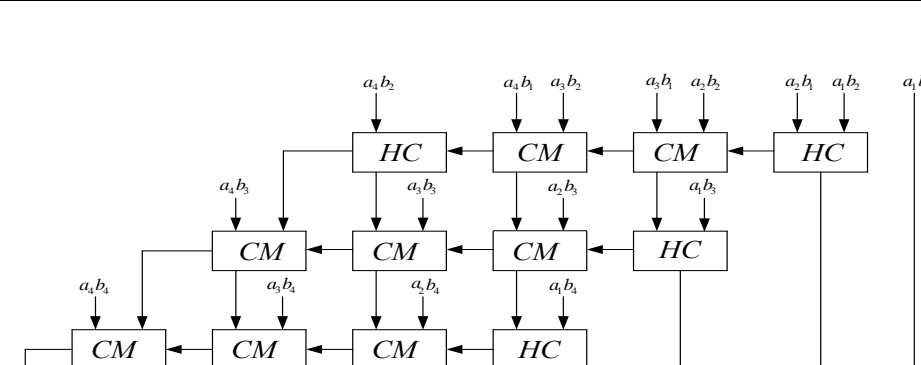
В роботах [156,157] запропоновано ряд структур деревоподібних перемножувачів на константу (рис. 2.4) та матричних перемножувачів двійкових чисел типу:

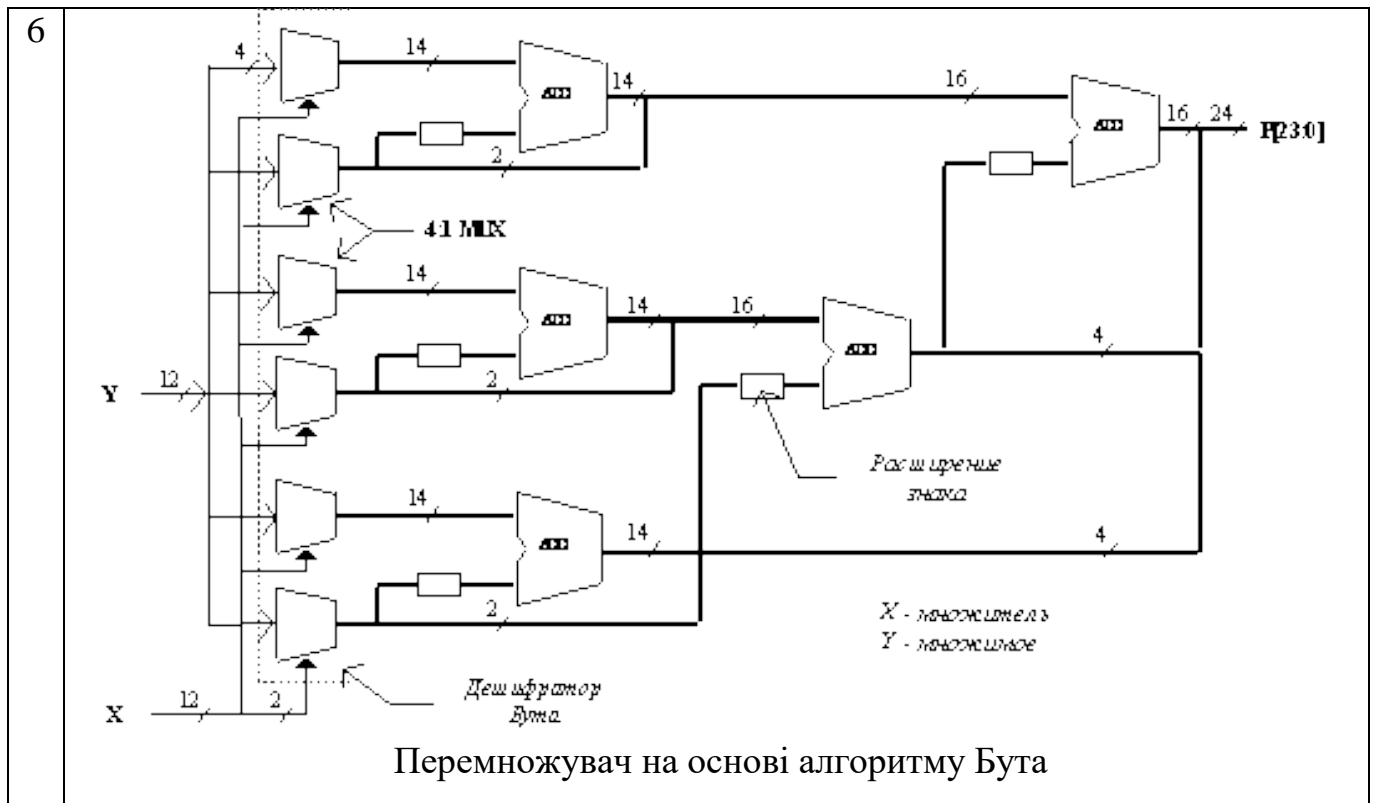
Брауна, Уоллеса, Дадда, перемножувача з горизонтальним розповсюдженням переносів а також швидкодіючий перемножувач Бута, структури яких представлені в табл. 2.9.

Таблиця 2.9 структури перемножувачів у базисі ТЧБ Радемахера



Продовження таблиці 2.9

3	 <p style="text-align: center;">перемножувач Уоллеса.</p>
4	 <p style="text-align: center;">перемножувач Дадда.</p>
5	 <p style="text-align: center;">перемножувач з горизонтальним розповсюдженням переносу.</p>



В основу роботи таких перемножувачів деревовидного та матричного типу покладено аналітичний вираз [157]:

$$Z = A \times B = \left( \sum_{i=0}^{n-1} a_i \times 2^i \right) \times \left( \sum_{j=0}^{n-1} b_j \times 2^j \right) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_i b_j \times 2^{i+j}, \quad (2.4)$$

де  $a_i, b_j \in \{0,1\}$ ,  $i, j \in \{0, n-1\}$ .

Аналіз швидкодії такого класу перемножувачів виконується згідно оцінки часової складності  $\tau_x = (2n-2)v$  апаратна складність  $A = (n^2 + 5n + 9(n^2 - 2n)) = (10n^2 - 13n)v$ .

В алгоритмі перемноження Бута використовується базовий модуль так званий дешифратор Бута, який виконує шляхом мультиплексування три обчислення операцій над двійковими кодами  $x$  та  $y$  що перемножуються, а також формування чотирьох вихідних кодів:  $0$ ;  $x$ ;  $2x$ ;  $3x$ .

Швидкодія перемножувача Бута характеризується більш розпаралеленим алгоритмом виконання операцій додавання кодів, які формуються дешифраторами Бута і розраховуються згідно виразу оцінки часової складності  $\tau_x = 5(n+1)\log_2 nv$ , апаратна складність у порівнянні з алгоритмом множення з паралельними



переносами зростає  $A = (n^2 + 5n + 9(n^2 - 2n)) = (10n^2 - 13n + 5\log_2 n)v$ , за рахунок додаткових суматорів у структурі перемножувача.

Виконане дослідження структур, часової та апаратної складності відомих деревоподібних та матричних перемножувачів двійкових чисел у базисі Радемахера дозволяє встановити їх функціональні обмеження та недоліки:

- 1) швидкодія такого класу перемножувачів пропорційно зменшується при зростанні розрядності перемножуваних чисел;
- 2) апаратна складність таких перемножувачів квадратно зростає у залежності від розрядності перемножуваних чисел;
- 3) в якості базових компонентів таких перемножувачів використовується неповні та повні однорозрядні суматори на логічних елементах І, АБО-НЕ, які мають часову складність  $\tau_{\Sigma} 3-5v$ , та апаратну складність  $A_{\Sigma} = 5-12v$ , що приводить до значного зниження швидкодії багаторозрядних перемножувачів;
- 4) наявність наскрізних переносів послідовно з'єднаних однорозрядних суматорів пропорційно  $n$ -розрядних перемножуваних чисел в  $2n$ - разів знижує швидкодію;
- 5) в структурі таких перемножувачів відсутні операції по модулю, що виключає можливість їх застосування при реалізації модульної арифметики у базисах Радемахера-Крестенсона, Хаара та Хаара-Крестенсона.

Запропонована у розділі 2.2.2 структура швидкодіючого АЦП у базисі Хаара-Крестенсона дозволяє ефективно застосувати матрично модульні перемножувачі у цьому базисі. На рис. 2.23 показано приклад виконання перемноження двох трьохрозрядних двійкових чисел у базисі Хаара, де на пересіченні вертикальних та горизонтальних шин розміщені логічні елементи І-НЕ.

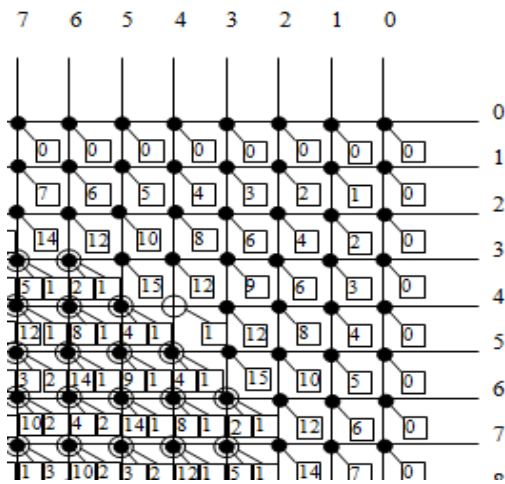


Рисунок 2.23 - Матричний перемножувач в базисі Хаара

Таким чином у базисі Хаара процес перемноження двох чисел певної розрядності виконується за один мікротакт. Розрахунок часової та апаратної складності такого класу перемножувачів виконується згідно виразів:  $\tau_x = 1\nu$ , апаратна складність  $A = 2^{2n}\nu$ . Фактично така структура представляє собою ПЗП з 3D пам'яттю. Наприклад при  $n=4,8,16$  відповідно апаратна складність буде рівна 256,  $2^{16}$ ,  $2^{32}$  що суттєво обмежує можливість практичного застосування перемножувачів такого класу у процесорах розрядність, яких перевищує 16 біт.

На рис. 2.24 показано приклад реалізації перемножувача довільного числа на константу у ТЧБ Хаара.

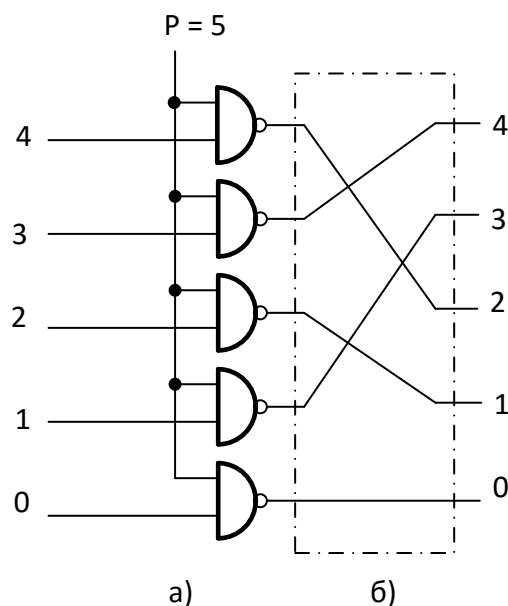


Рисунок 2.24 - Структура перемножувача на константу у ТЧБ Хаара.

а) лінійка логічних елементів І-НЕ б) рандомізатор

Аналіз структур матричного перемножувача на константу у ТЧБ Хаара дозволяє зробити наступні висновки:

- 1) перевагою таких структур є висока швидкодія а отже низька часова складність операцій множення, яка при перемноженні числа на константу складає  $\tau_x = 1v$ , апаратна складність рівна  $A = 2^n v$ ;
- 2) недоліком таких структур є швидке зростання апаратної складності у залежності від зростання розрядності перемножуваних чисел;
- 3) недоліком є формування вихідних кодів в інверсному вигляді без мультиплексування, що приводить до швидкого зростання розрядностей вихідних кодів Хаара;
- 4) недоліком також є наявність багатовходових логічних елементів АБО.

Названі недоліки швидкого зростання апаратної складності структур перемножувачів та перемножувачів на константу у базисі Хаара обмежує можливість застосування такого класу структур для побудови компонентів для реалізації швидкодіючих спектральних спецпроцесорів.

Таким чином проведений аналіз часової та апаратної складності перемножувачів у базисі Хаара визначає перспективу застосування мультибазисного ТЧБ Хаара-Крестенсона, оскільки даний базис суттєво зменшує апаратну складність перемножувачів без зниження високої швидкодії структур перемножувачів.

Оскільки на пересіченні шин матриць розміщений один елемент «І-НЕ», апаратна складність зменшується за рахунок переходу до базису Крестенсона, де число вентилів елементів замість  $N^2$ , буде рівне сумі  $P_i$ ,  $P_i$ - взаємопрості модулі відповідно до розрядності процесора.

На рис. 2.25 показана структура багатомодульного перемножувача у базисі Хаара-Крестенсона шляхом синтезу матрично-модульної структури на основі рис. 2.2.

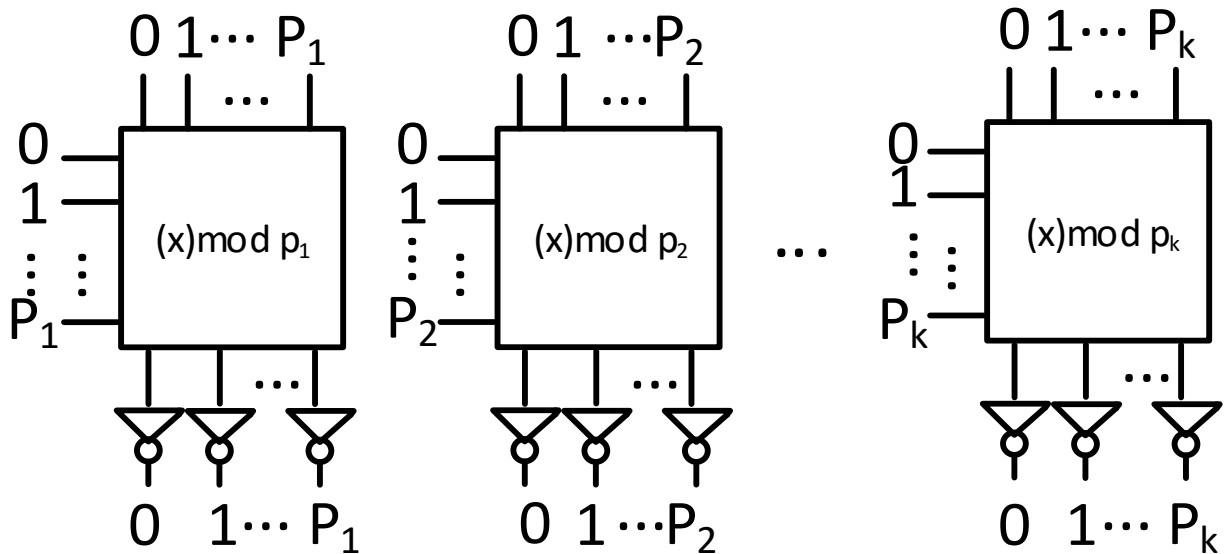


Рисунок 2.25 - Структура багатомодульного перемножувача у базисі Хаара-Крестенсона

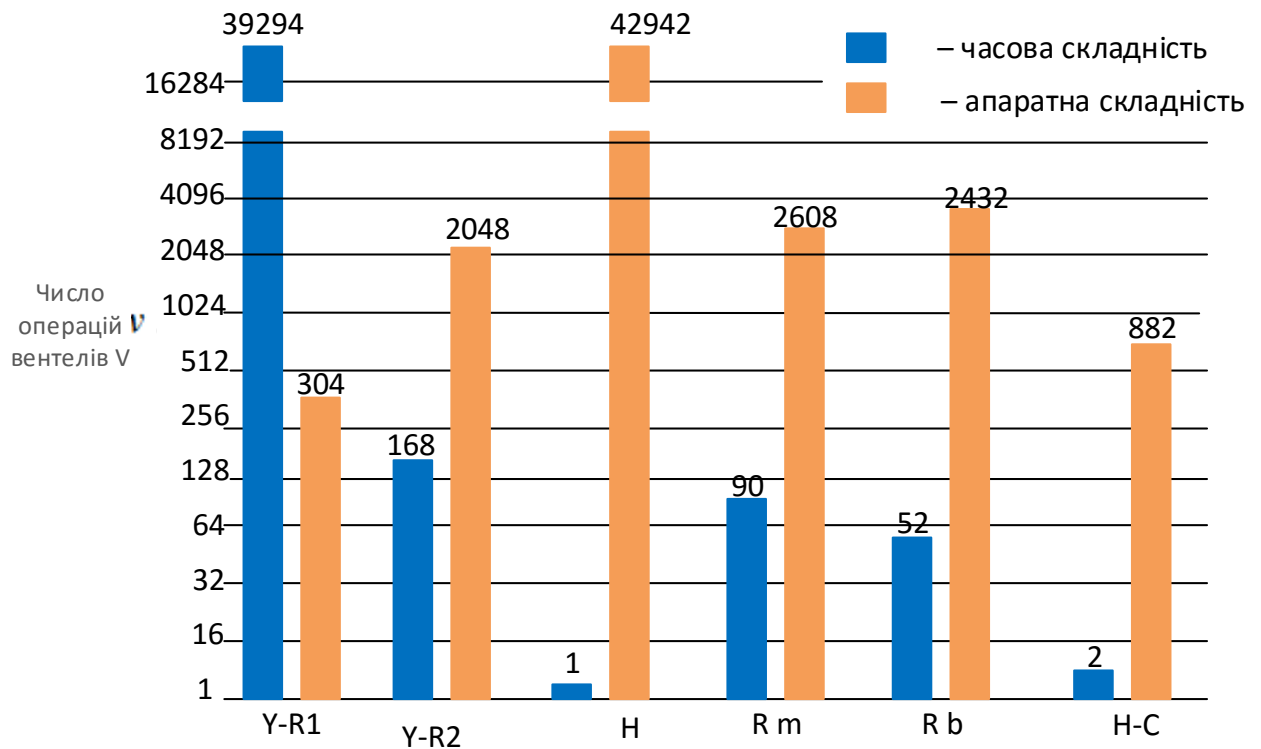
У результаті розрахунку часової та апаратної складності утримуємо  $\tau_x = 2\nu$ , апаратна складність рівна  $A = (P_1^2 + P_2^2 + \dots + P_k^2) + P_1 + P_2 + \dots + P_k$ .

Наприклад для процесорів розрядністю 4, 8, 16 біт апаратна складність такого перемножувача представлена у табл. 2.10.

Таблиця 2.11 - Апаратна складність перемножувача у ГЧБ Хаара Крестенсона

№	Число розрядів	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	Апаратна складність
1	4	4	5	-	-	-	-	-	$4^2 + 5^2 + 9 = 50$
2	8	5	7	8	-	-	-	-	$5^2 + 7^2 + 8^2 + 20 = 158$
3	16	7	11	13	15	16	-	-	$7^2 + 11^2 + 13^2 + 15^2 + 16^2 + 62 = 882$
4	32	13	19	23	27	29	31	32	$13^2 + 19^2 + 23^2 + 27^2 + 29^2 + 31^2 + 32^2 + 172 = 4786$

Порівняльна характеристика часової та апаратної складності досліджених перемножувачів представлена на рис. 2.26.



Y-R1 – Квадратор у ТЧБ (унітарно-Радемахера)  
 Y-R2 – Матричний перемножувач у ТЧБ (унітарно-Радемахера)  
 H – Перемножувач у ТЧБ (Хаара)  
 R m – Матричні перемножувачі у ТЧБ (Радемахера)  
 R b – Матричний перемножувач бута у ТЧБ (Радемахера)  
 H-C – Матричні перемножувачі у ТЧБ (Хаара-Крестенсона)

Рисунок 2.26 - Порівняльна характеристика часової та апаратної складності перемножувачів у різних ТЧБ

Проведені дослідження часової та апаратної складності перемножувачів реалізованих в унітарному, Хаара, Радемахера та Хаара-Крестенсона ТЧБ показує, що високу швидкодію наближеної до максимальною забезпечують перемножувачі Хаара-Крестенсона при відносно невисокій апаратній складності. Наприклад, при реалізації перемножувачів 16-розрядних двійкових чисел їх швидкодія у базисі Хаара-Крестенсона перевищує швидкодію еквівалентних перемножувачів у базисі Радемахера у 45 разів, а швидкодію перемножувачів в унітарному ТЧБ перевищує на 3 порядки. Таким чином обґрунтовано застосування структур матрично-модульних перемножувачів у базисі Хаара-Крестенсона для реалізації відповідних компонентів множення швидкодіючих спектральних спецпроцесорів косинусного перетворення Фур'є.

## 2.4. Реалізація та системні характеристики накопичуючих суматорів

Для побудови спектрального спецпроцесора проведено дослідження характеристик структурних схем, часової та апаратної складності накопичуючих суматорів, які є компонентами цифрового корелятора та входять у склад спектрального процесора при формуванні кінцевого результату оцінки спектру. На рис. 2.27 приведено функціональну схему накопичуючого суматора у базисі Радемахера.

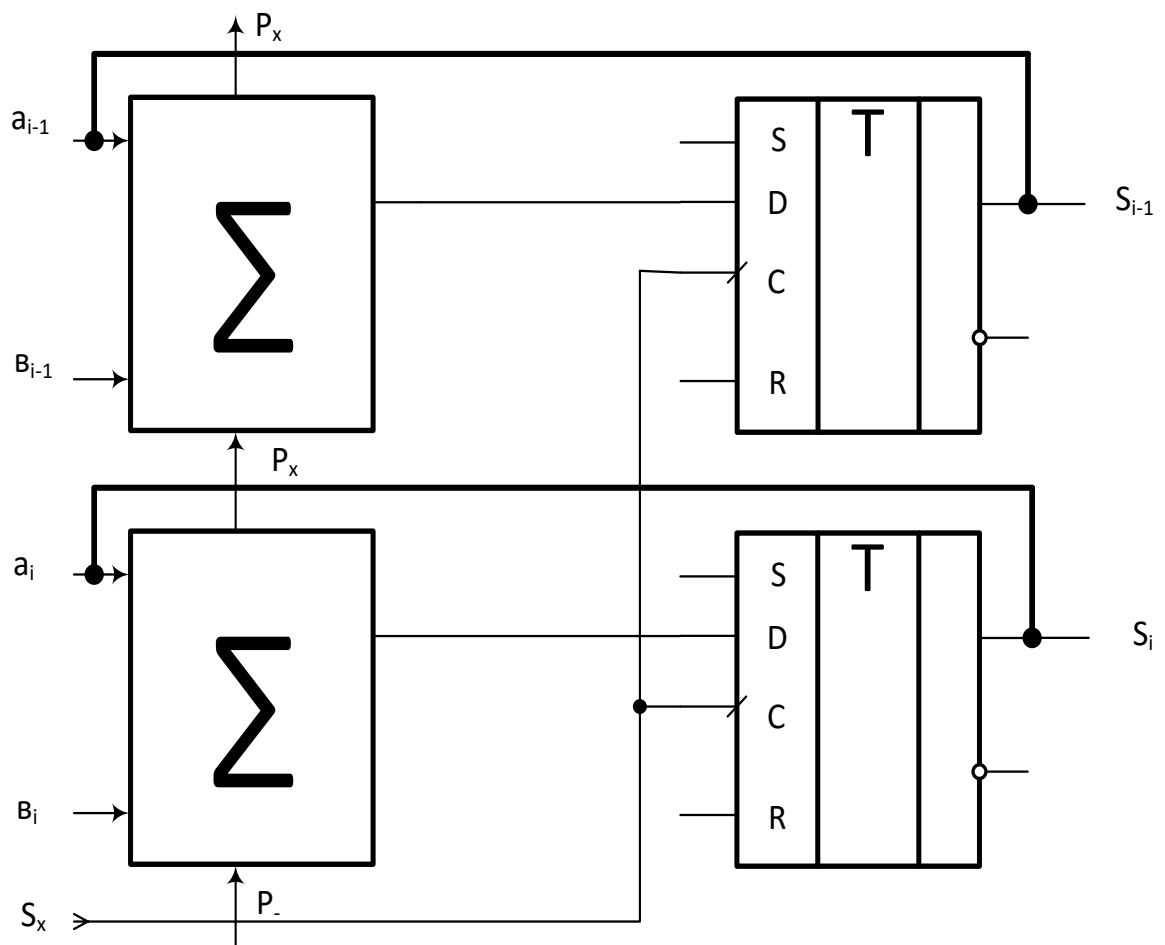


Рисунок 2.27 – Функціональна схема накопичуючого суматора у базисі Радемахера

Структура повного однорозрядного суматора, який є компонентом накопичуючого суматора приведена на рис. 2.28.

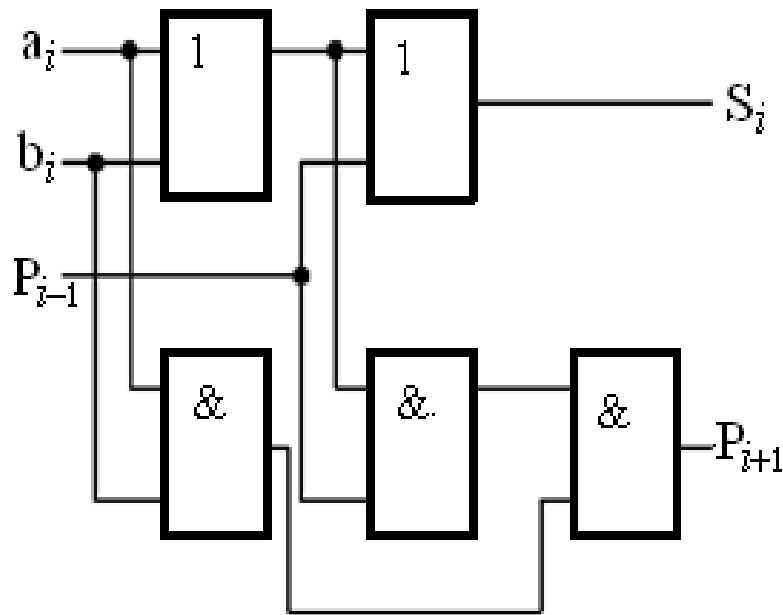


Рисунок 2.28 - Структурна схема однорозрядного повного суматора у базисі Радемахера

Часова та апаратна складності повного однорозрядного суматора розраховується згідно виразів:  $\tau = (\tau_{\Sigma} + \tau_T)n$ ,  $A = [A_{\Sigma} + A_T]n$ ,

де  $\tau_{\Sigma}$  та  $\tau_T$  - відповідно часова та апаратна складність повного однорозрядного суматора та D – тригера регістра пам'яті, аналогічно  $A_{\Sigma}$  та  $A_T$  - їх апаратні складності,  $n$  - розрядність накопичуючого суматора,  $m$  - число зсувів вхідних даних у БРЗ.

Аналіз часової та апаратної складності даного суматора при 16-ти розрядах показує, що часова складність становить  $\tau = 48v$  а апаратна складність становить  $A = 704v$ .

Недоліком відомої структури багаторозрядного суматора у базисі Радемахера є низька швидкодія його однорозрядних компонентів (рис. 2.28), тривалість переключення яких становить для різних реалізацій 3-5v.

На рис. 2.29 зображена структурна схема накопичуючого суматора з парафазними виходами та переносами.

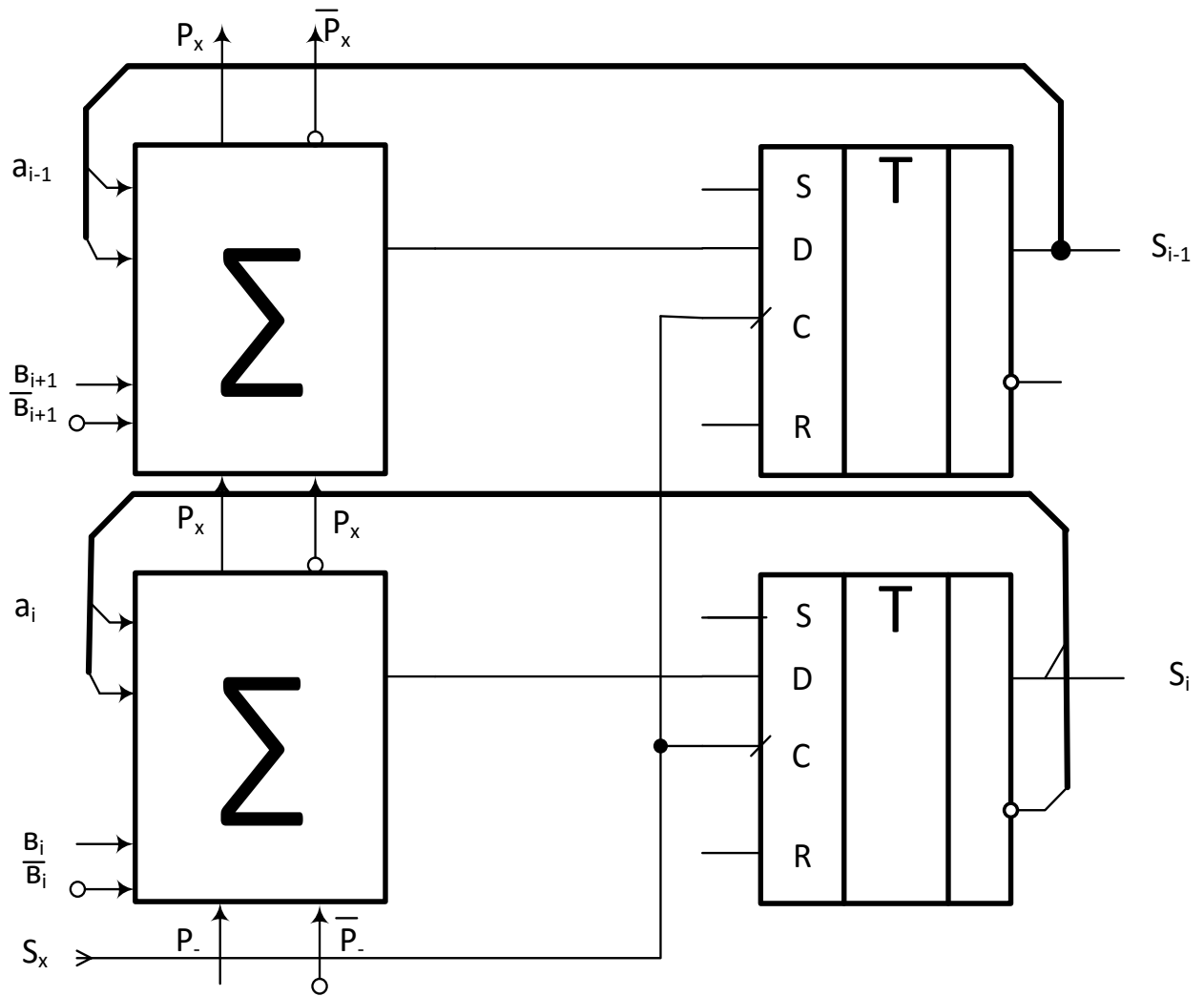


Рисунок 2.28 - Функціональна схема багаторозрядного суматора з парафазними виходами

Така реалізація накопичуючого суматора дозволяє застосувати структурні схеми однорозрядних суматорів з парафазними входами та виходами (рис.2.29), які реалізуються на основі логічних елементів І-НЕ з парафазними входами, виходи яких з'єднанні між собою елементом «провідне АБО». Така реалізація однорозрядного компонента накопичуючого суматора забезпечує формування переносів старших розрядів та коду суми за один мікротакт, що підвищує швидкодію накопичуючого суматора у 3 рази.



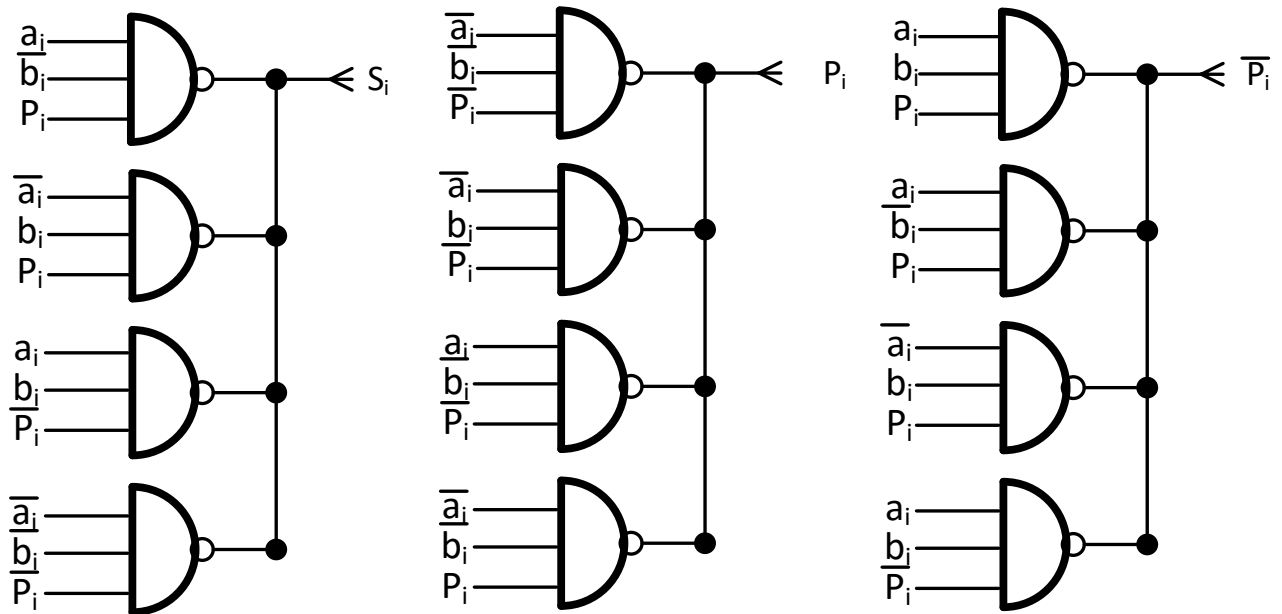


Рисунок 2.29 - Структурна схема однорозрядного повного суматора з парафазними виходами та виходами.

Аналіз часової та апаратної складності такого класу суматорів при 16 розрядах показує, що часова складність становить  $\tau = 136v$ , а апаратна складність при 16-ти розрядах становить  $A=544v$ .

З метою підвищення швидкодії накопичуючого суматора у структурі спектрального спецпроцесора запропоноване виконання його однорозрядних компонентів у базисі Хаара-Крестенсона.

На рис. 2.30 приведено структурну схему накопичуючого суматора у базисі Хаара-Крестенсона, де  $\sum p_i$  - однорозрядний суматор Хаара по модулю,  $V_i$  - залишок вхідного цифрового коду  $x_i$ , який формується на виході відповідного модульного розряду АЦП Хаара-Крестенсона.

Даниа структура суматора показує, що часова складність запропонованого рішення є найменшою і складає 3 мікротакти, незалежно від розрядності спектрального спецпроцесора, а апаратна складність зростає в 1,5 рази в залежності від розрядності процесора.

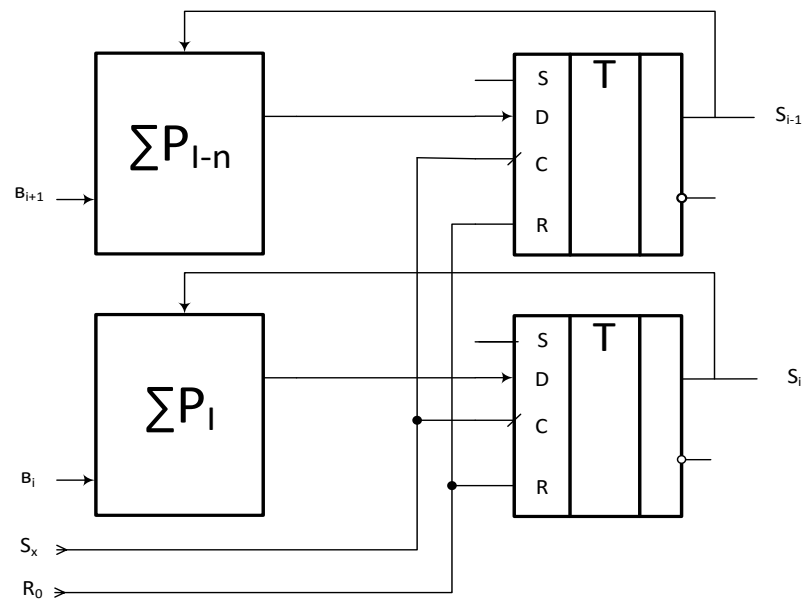


Рисунок 2.30 - Структурна схема накопичуючого суматора у базисі Хаара-Крестенсона

$S_x$  - вхід синхронізації,  $R_0$  - вхід скиду D- тригерів у нульовий стан.

Приклад структури компонента модульно-матричного накопичуючого суматора по модулю 11 приведена на рис 2.31. На пересіченні шин такого суматора розміщені двохвходові елементи І-НЕ.

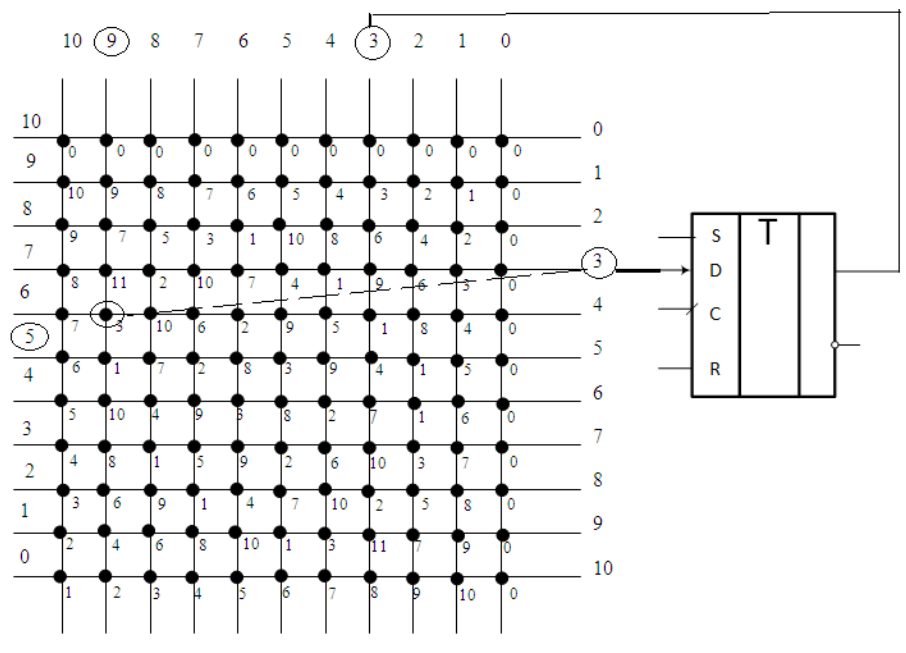


Рисунок 2.31 - Структура однорозрядного модульно-матричного накоплюючого суматора по модулю  $P=11$

Аналіз часової складності такого компонента накопичуючого суматора

розраховується згідно виразу:  $\tau = (\tau_{\Sigma} + \tau_T)$ , складає  $\tau = 3\nu$  і не залежить від розрядності спектрального спецпроцесора.

Апаратна складність такого класу суматорів розраховується на основі виразу:  $A = \sum P_i [(A_{\Sigma} + A_T)]$ .

Апаратна та часова складності матрично-модульного накопичуючого суматора у базисі Хаара-Крестенсона розраховуються згідно виразів:

$$A_{nc} = A_1 + A_2 + A_3, \quad L = 4LE,$$

де  $A_1 = \sum_{i=1}^{\nu} P_j^2 \cdot LE$  - сумарне число логічних елементів І-НЕ в матрицях суматора з набором модулів  $P_j$ ,  $A_2 = \sum_{i=1}^{\nu} P_j \cdot 2LE$  - сумарне число D – тригерів у регістрі пам'яті накопичуючого суматора,  $A_3 = \sum_{i=1}^{\nu} P \cdot LE$  - сумарне число інверторів на виходах матричних суматорів по модулю.

При 16 - розрядах спектрального спецпроцесора апаратна складність рівна  $A=5484$ .

На рис. 2.32 показано порівняльні характеристики часової та апаратної складності багаторозрядних накопичуючих суматорів реалізованих згідно різних структур та ГЧБ.

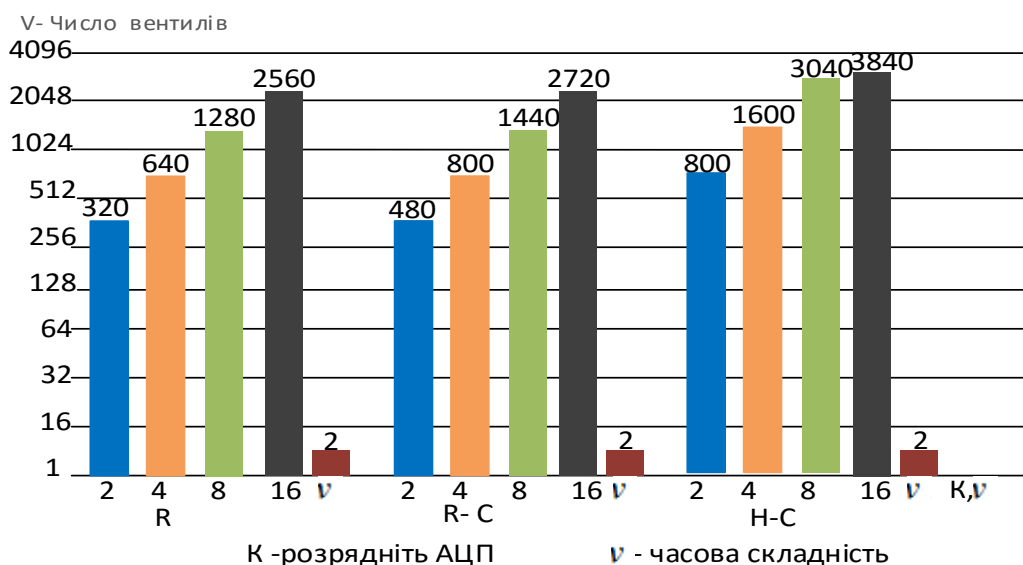


Рисунок 2.32 – Порівняльна характеристика часової та апаратної складності багаторозрядних накопичуючих суматорів

Аналіз характеристик часової та апаратної складностей відомих багаторозрядних накопичуючих суматорів у базисі Радемахера та запропонованої реалізації накопичуючого суматора у базисі Хаара-Крестенсона показує, що часова складність запропонованого рішення є мінімальною і складає 3 мікротакти незалежно від розрядності спектрального спецпроцесора, а апаратна складність зростає в залежності від розрядності процесора та перевищує у 3 рази, при цьому характеризується більш високою регулярністю структури компонентів, що спрощує мікроелектронну реалізацію на ПЛІС.

Шифратор, який використовується в структурі спектрального спецпроцесора реалізується на основі матричного ПЗП і виконує функції міжбазисного перетворення кодів Хаара-Крестенсона у двійкові коди базису Радемахера. Апаратна та часова складність, шифратора розраховується згідно виразів:

$$A_{ш} = 2K \sum p_i \cdot v, \tau_{ш} = 4v.$$

При  $K=16$  отримуємо  $A_{ш}=768$  для ПЗП.

Виконані дослідження апаратної та часової складності компонентів спектрального спецпроцесора, які реалізовані в базисі Хаара-Крестенсона включаючи: швидкодіючий АЦП, матрично-модульні перемножувачі та матрично-модульні накопичуючі суматори демонструють їх суттєві переваги по швидкодії та регулярності архітектури у порівнянні з існуючими рішень таких компонентів у двійковій арифметиці базису Радемахера.

## ВИСНОВКИ ПО ДРУГОМУ РОЗДІЛІ

1. Встановлено, що асимптотика автоковаріаційної функції  $K_{xx}(j)$  дозволяє застосувати отримані цифрові дані у результаті обчислення в нульовій точці  $K_{xx}(0)$  та при  $j = \infty$  значення дисперсії та квадрату математичного сподівання що спрощує алгоритм обчислення спектру досліджуваних сигналів.
2. Формалізовані і досліджені методи аналого-цифрового перетворення сигналів у різних ТЧБ та запропонований новий метод побудови швидкодіючого АЦП у базисі Хаара-Крестенсона, який характеризується у порівнянні з АЦП паралельного типу у базисі Радемахера зменшеною у 3 рази апаратною складністю, а представлення цифрових даних у кодовій системі Хаара-Крестенсона забезпечує максимальне зменшення часової складності базових компонентів спектрального спецпроцесора цифрових перемножувачів та накопичуючих суматорів при зростанні їх апаратної складності не більше ніж у 3 рази, а також підвищеною регулярністю архітектури, що спрощує їхню мікроелектронною реалізацію на ПЛІС.
3. На основі систематизації характеристик та критеріїв складності спектрального спецпроцесора, досліджені наступні базові компоненти: АЦП, БРЗ, квадратори, пристрої перемноження, накопичуючі суматори та шифратори, які можуть реалізовуватися в різних ТЧБ, що дозволило диференціювати оцінки часової та апаратної складності названих компонентів, а також виконати їх порівняльний аналіз в залежності від діапазону квантування вхідних цифрових даних.
4. У результаті виконаного дослідження апаратної та часової складності структурних схем мікроелектронних компонентів спектрального спецпроцесора встановлено, що максимальною швидкодією характеризуються цифрові матрично-модульні перемножувачі та накопичуючі суматори у базисі Хаара-Крестенсона, які обґрунтовано застосувати у якості базових компонентів швидкодіючих спектральних спецпроцесорів.

## РОЗДІЛ 3

РОЗРОБКА ТА ДОСЛІДЖЕННЯ АЛГОРИТМІВ РЕАЛІЗАЦІЇ ДИСКРЕТНОГО  
КОСИНУСНОГО ПЕРЕТВОРЕННЯ ФУР'Є НА ОСНОВІ РІЗНИХ  
АВТОКОРЕЛЯЦІЙНИХ ФУНКЦІЙ

3.1. Дослідження формалізованого алгоритму обчислення спектрів на основі центрованої нормованої автокореляції.

Алгоритм ДКПФ виконується шляхом обчислення АКФ на основі вхідного масиву даних  $\{x_i\}$  та адаптації її характеристик до нормалізованих і центрованих характеристик базисних функцій ТЧБ згідно алгоритму представленого функціоналом  $F[\circ]$ :

$$F[\text{СКПФ}] = F_1 \Rightarrow F_2 \Rightarrow F_3 \Rightarrow F_4 \Rightarrow F_5 \Rightarrow F_6 \Rightarrow F_7 \Rightarrow F_8 \Rightarrow F_9,$$

де:  $F_1[x(t)]$ -реєстрація аналогового вхідного сигналу;

$F_1[x_i]$ - аналого-цифрове перетворення  $x(t)$  і формування вхідної вибірки масиву даних  $\{x_i\}$ ;  $i \in \overline{1, n}$ ;  $0 \leq x_i \leq A$ ;  $n$ - об'єм вибірки;  $A$  – діапазон квантування  $A = 2^p$ ;  $p$ - розрядність двійкового представлення цифрових значень  $x_i$ .

$$F_2[M_x] - \text{обчислення математичного сподівання } M_x = \frac{1}{n} \sum_{i=1}^n x_i;$$

$$F_3\left[\left\{x_i^\circ\right\}\right] - \text{обчислення масиву центрованих цифрових відліків } \{x_i^\circ\} = \{x_i - M_x\};$$

$$F_4[D_x] - \text{обчислення дисперсії } D_x = \frac{1}{n} \sum_{i=1}^n (x_i^\circ)^2;$$

$$F_5[R_{xx}(j)] - \text{обчислення центрованої АКФ } R_{xx}(j) = \frac{1}{n} \sum_{i=1}^n x_i^\circ \cdot x_{i+j}^\circ;$$

$$F_6[\rho_{xx}(j)] - \text{обчислення нормованої АКФ } \rho_{xx}(j) = \frac{R_{xx}(j)}{D_x}; j = 1, m; m - \text{число точок АКФ};$$

$$F_7[|\rho_{xx}(j-1) - \rho_{xx}(j)| \leq 0,01; j = m] - \text{визначення інтервалу кореляції } m;$$

$$F_8[e^{-\gamma j}] - \text{визначення коефіцієнта } \gamma \text{ затухання енергії АКФ } \rho_{xx}(j) \text{ на інтервалі } j \in \overline{0, m};$$

$F_9[S(\omega_i) i \in \overline{0, k}, e^{-\gamma i}]$  - обчислення спектру у базисі Фур'є для  $\omega_i, i \in \overline{0, k}$ ,  $k$  число гармонік спектрального перетворення Фур'є згідно виразу:

$$S(\omega_i) = \frac{1}{m} \sum_{j=i}^m \rho_{xx}(j) \cdot \cos \omega_{ij} \cdot e^{-\gamma j}$$

Проведені дослідження обчислювальної складності даного алгоритму демонструє значний об'єм обчислень для розрахунку спектру що принципово обмежує можливості застосування такого формалізованого алгоритму обчислення спектру при створенні швидкодіючих спектральних спецпроцесорів.

3.2 Розробка та дослідження формалізованого алгоритму обчислення спектрів на основі кореляційної функції автоковаріації з адаптацією характеристик базисних функцій різних теоретико-числових базисів.

З метою зменшення числа обчислювальних операцій при реалізації ДКПФ запропонований метод спектрального перетворення шляхом адаптації параметрів базисних функцій ТЧБ (рис.3.1) до характеристик реалізації нецентрованої і ненормованої коваріаційної АКФ, обчисленої на основі масиву даних  $\{x_i\}$  згідно

виразу  $K_{xx}(j) = \frac{1}{N} \sum_{i=1}^N x_i \cdot x_{i+j}$ ,  $j \in \overline{0, m_\tau}$ . Дана АКФ має асимптотику  $K_{xx}(0) = D_x + M_x^2$ ;

$K_{xx}(\infty) = M_x^2$  до якої адаптується ортогональна гармонічна функції ТЧБ Фур'є, як показано на рис 3.1.

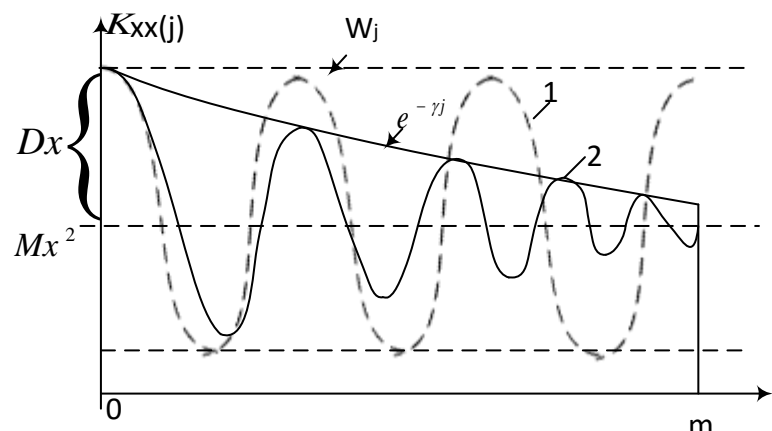


Рисунок 3.1 – Асимптотика коваріаційної АКФ та базисних функцій косинусного перетворення Фур'є без врахування затухання енергії (1) та затухання енергії (2).

Як видно з рис 3.1 функція спектру визначається шляхом обчислення коефіцієнта взаємкореляції між автоковаріаційною функцією та базисною функцією косинусного перетворення Фур'є, в якій враховано затухання енергії згідно  $e^{-\lambda j}$ .

Приклади ортогональних функцій  $W_{ij}$  відомих ТЧБ адаптованих до асимптотики коваріаційної АКФ показані на рис 3.2.

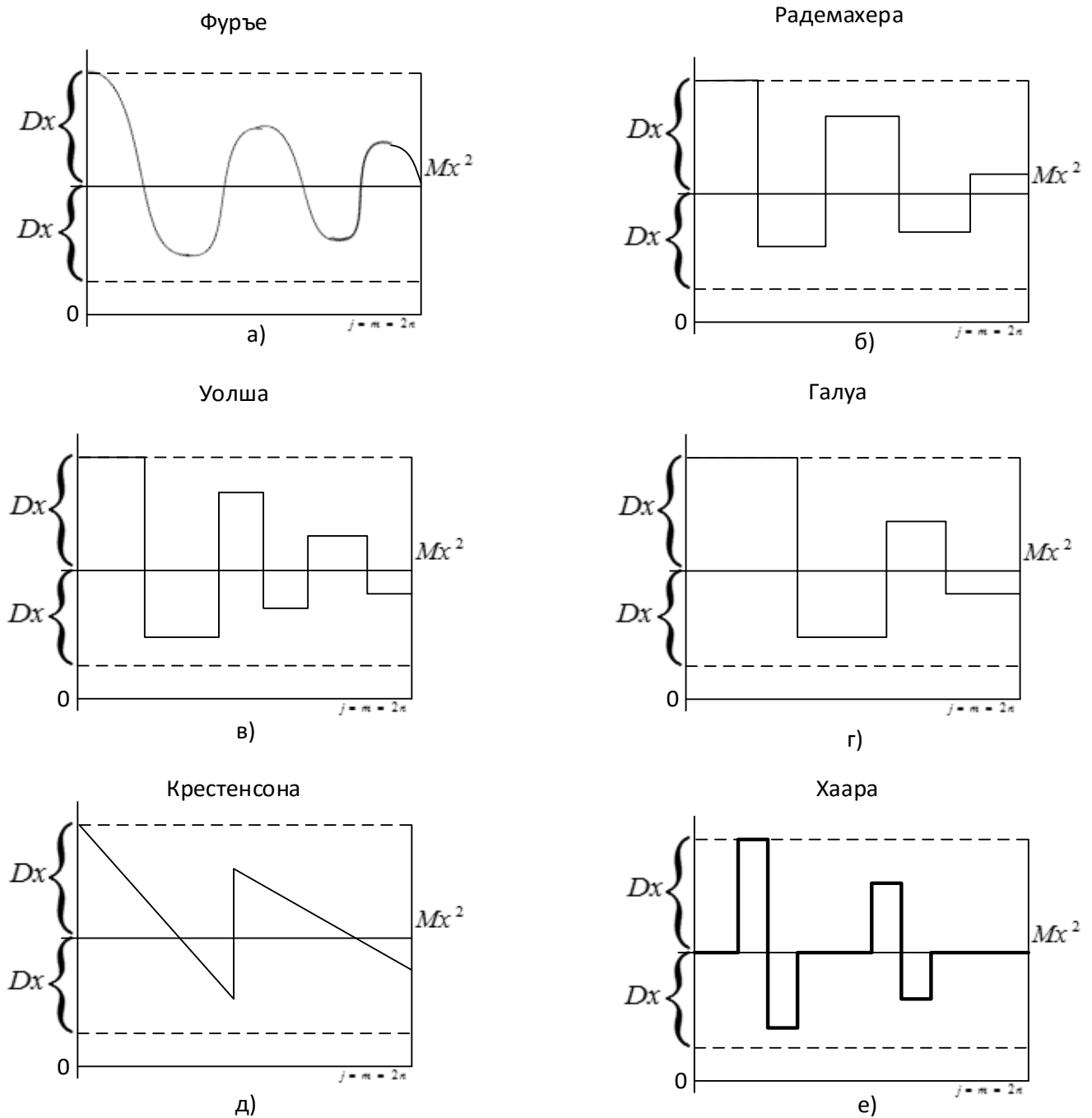


Рисунок 3.2 – Реалізація ортогональних функцій ТЧБ адаптованих до параметрів асимптотики коваріаційної АКФ.



Таким чином у запропонованому алгоритмі ДКПФ необхідно адаптувати характеристики базисної функції  $W(j)$  до реалізованої  $K_{xx}(j)$  з врахуванням її асимптотики. Процес адаптації характеристик базисної функції ДКПФ

зводиться до обчислення або вибірки з пам'яті процесора функції  $W(j) = (Mx^2 + Dx \cdot \cos(w_j)) \cdot e^{-Vj}$ . Перевага реалізації даного методу ДКПФ у ТЧБ (W) полягає в тому, що при виконанні такого алгоритму відсутня необхідність додаткового обчислення математичного сподівання  $Mx^2$ , центрованих значень  $\hat{X}_i$ , дисперсії  $Dx$  та операцій нормування АКФ

$$\rho_{xx(j)} = \frac{R_{xx}(j)}{Dx}.$$

Таким чином запропонований алгоритм визначення спектру виконується шляхом обчислення коефіцієнта взаємкореляції згідно виразу:

$$S(w) = \frac{1}{m} \sum_{j=1}^m K_{xx}(j) \cdot \hat{w}(j) - M_x^2. \quad (3.1)$$

Тому запропонований метод ДКПФ на основі автоковаріаційного корелятора в базисі Хаара-Крестенсона [11], який реалізується в системі залишкових класів шляхом представлення вхідних даних  $[x_i]$  набором залишків взаємнопростих модулів [12] виконується згідно алгоритму:

1. Вибирається система взаємно простих модулів, добуток яких перевищує діапазон квантування  $X_i$  ( $0 \leq X_i \leq A$ ) і відповідає умові:  $\prod_{i=1}^k P_i \geq A \cdot m \cdot n$ ;

Формалізація математичних операцій запропонованої структури спецпроцесора згідно виразу (3.1) представлена на рис.3.3

$$\begin{array}{l}
 \rightarrow x_i(\text{mod } p_1) = b_{1i} \Rightarrow C_{01} = b_{1i} \cdot b_{1i-0}(\text{mod } p_1) C_{11} = b_{1i} \cdot b_{1,i-1}(\text{mod } p_1) C_{m1} = b_{1i} \cdot b_{1i-m}(\text{mod } p_1) \\
 \rightarrow x_i(\text{mod } p_2) = b_{2i} \Rightarrow C_{02} = b_{2i} \cdot b_{2i-0}(\text{mod } p_2) C_{12} = b_{2i} \cdot b_{2,i-1}(\text{mod } p_2) C_{m2} = b_{2i} \cdot b_{2i-m}(\text{mod } p_2) \\
 \dots \\
 \rightarrow x_i(\text{mod } p_k) = b_{ki} \Rightarrow C_{0k} = b_{ki} \cdot b_{ki-0}(\text{mod } p_k) C_{1k} = b_{ki} \cdot b_{k,i-1}(\text{mod } p_k) C_{mk} = b_{ki} \cdot b_{ki-m}(\text{mod } p_k)
 \end{array} \left. \vphantom{\begin{array}{l} \dots \\ \dots \end{array}} \right\} S_0$$

$$\begin{array}{l}
 [C_{01} \cdot W_{00}(\text{mod } p_1) + C_{11} \cdot W_{01}(\text{mod } p_1) \dots C_{m1} \cdot W_{0m}(\text{mod } p_1) = S_{01}] \\
 [C_{02} \cdot W_{02}(\text{mod } p_2) + C_{12} \cdot W_{02}(\text{mod } p_2) \dots C_{m2} \cdot W_{0m}(\text{mod } p_2) = S_{02}] \\
 \dots \\
 [C_{0k} \cdot W_{0k}(\text{mod } p_k) + C_{1k} \cdot W_{0k}(\text{mod } p_k) \dots C_{mk} \cdot W_{0m}(\text{mod } p_k) = S_{0k}]
 \end{array} \left. \vphantom{\begin{array}{l} \dots \\ \dots \end{array}} \right\} S_1$$

$$\begin{array}{l}
 [C_{01} \cdot W_{l0}(\text{mod } p_1) + C_{11} \cdot W_{l1}(\text{mod } p_1) \dots C_{m1} \cdot W_{lm}(\text{mod } p_1) = S_{l1}] \\
 [C_{02} \cdot W_{l2}(\text{mod } p_2) + C_{12} \cdot W_{l2}(\text{mod } p_2) \dots C_{m2} \cdot W_{lm}(\text{mod } p_2) = S_{l2}] \\
 \dots \\
 [C_{0k} \cdot W_{lk}(\text{mod } p_k) + C_{1k} \cdot W_{lk}(\text{mod } p_k) \dots C_{mk} \cdot W_{lm}(\text{mod } p_k) = S_{lk}]
 \end{array} \left. \vphantom{\begin{array}{l} \dots \\ \dots \end{array}} \right\} S_l$$

Рисунок 3.3 - Формалізація алгоритму обчислення спектра косинусного перетворення Фур'є на основі різних ортогональних функцій  $w_{ij}$  різних ТЧБ.

S0- залишки  $b_i \cdot b_{i-j}$ ;  $i \in \overline{1, k}$  ;  $j \in \overline{0, m}$  ; k- число модулів  $p_i$ ; m – число значень  $b_j$ ;  
 S1-Sl – число обчислених значень точок спектру ; l-число базисних функцій.

Даний алгоритм використовується для реалізації структури спектрального спецпроцесора косинусного перетворення у базисі Хаара-Крестенсона..

### 3.3. Розрахунок системи взаємопростих модулів перетворення системи залишкових класів в базисі Крестенсона.

Розрядність спектрального спецпроцесора визначається згідно відповідної послідовності модульних обчислень, які представлені на рис. 3.4 та табл. 3.1.

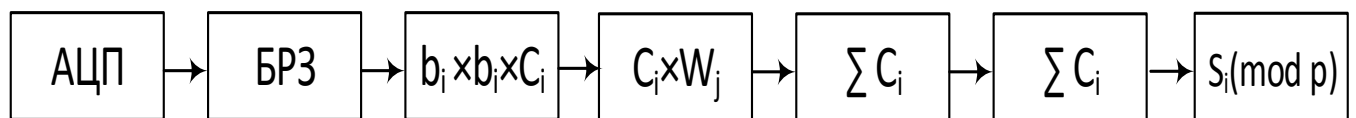


Рисунок 3.4 - Послідовності модульних обчислень для базисної функції  $W_j$ .

На рис. 3.4 показано приклад виконання послідовності виконання модульних операцій при заданій розрядності АЦП k, числі точок коваріаційної функції m, та об'єму вибірки n. Нехай k=4, m= 32, n = 256,  $b_i = res_{x_i}(\text{mod } P_i)$ . Тоді при виконанні операцій множення діючих та зсунутих у часі часових відліків розрядність добутоків

становить 8 біт. У результаті сумування отриманих добутків в накопичуючих суматорах кожного розряду обчислювача АКФ розрядність кодів становить 16 біт, а їх усереднених значень, які визначаються шляхом відкидання молодших 8 біт розрядність вихідних кодів становить 8 біт. Після перемноження отриманих кодів на коди базисних функцій  $W$  розрядність збільшується до 16 біт, які підраховуються в суматорах з числом  $m=32$ , вихідні коди яких також усереднюються, що відповідає розрядності спектрального процесора у базисі Радемахера 16 біт.

Таким чином розрядність спектрального спецпроцесора у базисі Хаара Крестенсона повинна відповідати 16-ти бітам для даного прикладу, що наведено у табл. 3.1.

Таблиця 3.1 - Залежність розрядності та системи модулів спектрального спецпроцесора від розрядності АЦП.

АЦП біт	БРЗ	$b_i \times b_j \times C_i$	$C_i \times W_j$	$\sum C_i$	$\sum C_i$	$S_j(\text{mod } p)$	$P_1, P_2, \dots, P_k$
k=2	2	4	6	6	6	7	5,7,8
k=4	4	8	12	12	12	13	7,11,13,17,19
K=8	8	16	24	24	24	25	13,19,23,27,29,31,32

У процесі обчислення коваріаційної функції згідно модульної арифметики СЗК виконується матрично-модульне перемноження діючих та зміщених в регістрі цифрових відліків, а також накопичуюче матрично-модульне додавання кодів Хаара-Крестенсона. У результаті, у кожному  $j$ -му каналі по  $P_i$  модулю виконується операція модульного множення і на виходах формуються коди залишків згідно виразу:

$$b_m(H) = \text{res}[b_i(H) + b_{i-j}(H)] \text{ mod } P_i, \quad (3.2)$$

де  $\text{res}$  - визначення залишку по модулю  $P_i$ .

Функціональна структура компонента спектрального спецпроцесора в якому обчислюється значення функції коваріації показана на рис 3.5.

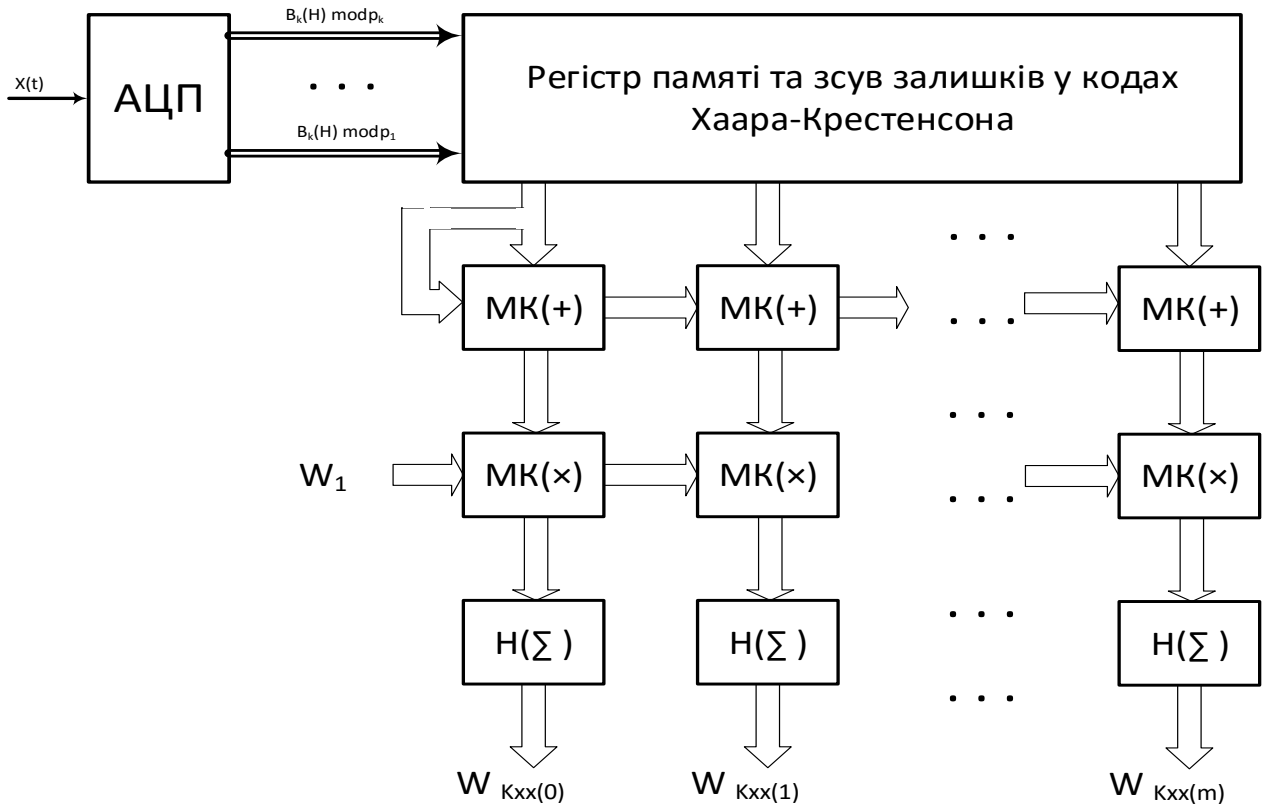


Рисунок 3.5 - Функціональна структура обчислення коваріаційної та компонентів спектральної функції косинусного перетворення Фур'є.

Структура матрично-модульних перемножувачів у базисі Хаара показана на рис. 3.6.

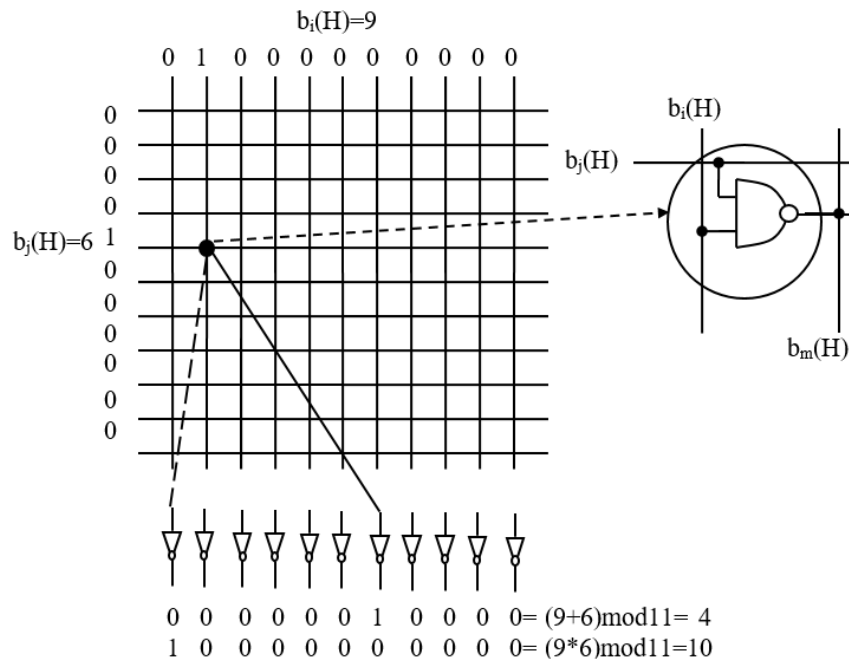


Рисунок 3.4 - Структура модульної матриці додавання та множення кодів Хаара при  $P_i=11$

Особливістю такого класу матрично-модульних суматорів та перемножувачів є висока регулярність структури та швидкодія, яка відповідає 2-м мікротактам.

При створенні спектральних спецпроцесорів у базисі Радемахера-Крестенсона необхідно розраховувати набір взаємопростих модулів ( $P_1, P_2, \dots, P_k$ ). При цьому окремі модулі можуть бути простими числами, а інші добутками простих чисел. Окремі модулі можуть бути представлені квадратами та більш високими степеннями простих чисел, а один з модулів може бути представленим числом  $2^k$ , де  $k$ - ціле число більше 1.

В таблиці 3.2 приведені розраховані програмних шляхом набори взаємопростих модулів для процесорів базису Крестенсона різної розрядності та взаємопростих модулів різної розрядності.

Таблиця 3.2 - Набори взаємопростих модулів для процесорів базису Крестенсона

Розрядність біт	Прості числа	Складні числа
3	3,5	7,8
4	7,11,13	15,16.
5	7,11,13,17,19,23	25,27,29,31,32.
6	13,17,19,23,29,31,37,41,43,47,53,59,61	55,63,64
7	13,19,23,29,31,37,43,47,53,59,61,67,71,73,79,83,89,97,101,103,107,109,113.	119,121,123,125,127,128.
8	29,37,41,43,47,53,59,61,67,71,73,79,83,89,97,101,103,107,109,113,127,131,137,139,149,151,157,163,167,173,179,181,191,193,197,199,211,223,227,229,233,239,241,251.	217,247,253,255,256.
9	31,41,67,71,79,83,89,97,103,107,109,113,127,131,137,139,149,151,157,163,173,179,181,191,193,197,199,211,223,227,229,233,239,241,251,257,263,269,271,277,281,283,293,307,311,313,317,331,337,347,353,359,367,373,379,383,389,397,401,409,419,421,431,433,439,443,449,457,461,463,467,473,479,487,491,499,503,509,	437,473,481,493,501,505,511,512.

## Продовження таблиці 3.2

10	37,41,47,61,67,71,79,83,89,97,101,103,107,109,113,127,131,137,149,151,157,163,167,173,179,181,191,193,197,211,223,227,229,233,239,241,251,257,263,269,271,277,281,283,293,307,311,313,317,331,337,347,349,353,359,367,373,379,383,389,397,401,409,419,421,431,433,439,443,449,457,461,463,467,479,487,491,499,503,509,521,523,541,547,557,563,569,571,577,587,593,599,601,607,613,617,619,631,641,643,647,653,659,661,673,677,683,691,701,709,719,727,733,733,739,743,751,757,761,769,773,787,797,8009,811,821,823,827,829,839,841,853,857,859,863,877,881,883,887,907,911,919,929,937,941,947,953,967,971,977,983,991,1009,1013,1019,1021.	949,973,989,995,1003,1007.
----	---	----------------------------

В таблиці 3.3 розраховані на основі розробленої програми набори модулів для спектральних процесорів різної розрядності.

Таблиця 3.3 – Набори взаємопростих модулів та їх характеристики для процесорів різної розрядності

	Набори взаємопростих модулів	Обернені елементи	Ортогональні базиси СЗК	Діапазон кодування чисел	Розрядність процесора
		$m_i$	$B_i$	$P$	$k$
4 бітні:	16	7	105105	240240	16
	15	11	176176		
	13	2	36960		
	11	9	196560		
	7	6	205920		
5 бітні:					
1)	29	28	63308	65569	24
	19	8	27608		
	17	8	30856		
	7	1	9367		
2)	27	1	2431	65637	18
	17	9	34749		
	13	8	40392		

## Продовження таблиці 3.3

	11	9	53703		
3)	29	9	20475	65975	18
	25	9	23751		
	13	8	40600		
	7	5	47125		
4)	32	31	64449	66528	18
	27	4	9856		
	11	5	30240		
	7	3	28512		
5)	32	5	10465	66976	18
	23	5	14560		
	13	10	51520		
	7	6	57408		
6 бітні:					
1)	63	46	48070	65835	18
	55	38	45486		
	19	11	38115		
2)	61	43	46483	65941	18
	47	20	28060		
	23	20	57340		
3)	53	36	44892	66091	18
	43	39	59943		
	29	12	27348		
4)	53	37	64343	92167	18
	47	18	35298		
	37	34	84694		
5 бітні:					
1)	32	1	137894913	4412637216	34
	31	16	2277490176		

## Продовження таблиці 3.3

	29	25	3803997600		
	27	17	2778327136		
	23	14	2685953088		
	19	14	3251416896		
	13	8	2715469056		
2)	31	1	145422675	4508102925	34
	29	4	621807300		
	27	8	1335734200		
	25	3	540972351		
	23	13	2548058175		
	19	15	3559028625		
	17	1	265182525		
3)	32	21	5568833025	8485840800	34
	31	17	4653525600		
	29	13	3803997600		
	27	11	3457194400		
	25	18	6109805376		
	23	22	8116891200		
	19	5	2233116000		
6 бітні:					
1)	64	15	1006745025	4295445440	34
	59	16	1164866560		
	55	27	2108673216		
	43	36	3596186880		
	37	9	1044838080		
	13	12	3965026560		
2)	59	44	3277649716	4395030301	34
	47	17	1589691811		
	43	32	3270720224		



## Продовження таблиці 3.3

	41	36	3859050996		
	31	18	2551953078		
	29	20	3031055380		
3)	59	6	436916418	4296344777	34
	53	45	3647839905		
	47	15	1371173865		
	41	4	419155588		
	31	25	3464794175		
	23	19	3549154381		
7 бітні:					34
1)	125	59	2027274751	4295073625	
	107	11	441549625		
	83	75	3881090625		
	73	67	3942053875		
	53				
2)	119	54	1951010118	4299448223	34
	89	78	3768055746		
	83	13	673407553		
	73	44	2591448244		
	67	61	3914423009		
3)	101	57	2428247253	4302683729	34
	89	73	3529167553		
	83	21	1088630823		
	79	73	3975897623		
	73	32	1886107936		
4)	128	83	2785025409	4294978944	34
	123	70	2444296960		
	101	45	1913604480		
	73	68	4000802304		
	37	15	1741207680		

Розрядність розрахованих спецпроцесорів спектрального перетворення Фур'є у базисі Радемахера-Крестенсона перевищує відповідну розрядність спецпроцесорів у базисі Радемахера на 2 біти, оскільки при розрахунку добутків взаємопростих модулів їх розрядність зменшується на 1-2 біти по відношенню до суми розрядностей наборів модулів.

У табл. 3.4 розраховані відповідні розрядності процесорів спектрального косинусного перетворення Фур'є у базисі Хаара-Крестенсона які, розраховуються згідно виразу:

$$K_{HC} = \sum_{i=1}^k P_i .$$

Таблиця 3.4 - Розрядності спектрального косинусного перетворення Фур'є у базисі Хаара-Крестенсона

Набори взаємопростих модулів							Розрядність спецпроцесорів		
P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	ТЧБ Радемахера	ТЧБ Радемахера-Крестенсона	ТЧБ Хаара-Крестенсона
7	11	13	15	16			16	18	62
7	17	19	29				16	18	72
11	13	17	27				16	18	68
7	13	25	29				16	18	74
7	11	27	32				16	18	77
7	13	23	32				16	18	75
19	55	63					16	18	137
23	47	61					16	18	131
29	43	53					16	18	125
37	47	53					16	18	137
13	19	23	27	29	31	32	32	34	174

Продовження таблиці 3.4

17	19	23	25	27	29	31	32	34	171
19	23	25	27	29	31	32	32	34	186
13	37	43	55	59	64		32	34	271
29	31	41	43	47	59		32	34	250
23	31	41	47	53	59		32	34	254
53	73	83	107	125			32	34	441
67	73	83	89	119			32	34	443
73	79	83	89	101			32	34	425
37	73	101	123	128			32	34	462

З таб. 3.2 та 3.4 видно, що при реалізації спектральних спецпроцесорів косинусного перетворення Фур'є їх розрядність у базисах ТЧБ Радемахера-Крестенсона та Хаара-Крестенсона у порівнянні з розрядністю спецпроцесорів у базисі Радемахера відповідно зростає на 2 біти а апаратна складність зростає у середньому у 2-3 рази. У той же час швидкодія спецпроцесорів у базисі Хаара-Крестенсона у порівнянні з швидкодією спецпроцесорів у базисі Радемахера зростає у 2-3 порядки, а також характеризується більш регулярною архітектурою з глибоким розпаралеленням обчислювальних процесів та відсутністю наскрізних міжрозрядних переносів.

При цьому об'єм пам'яті базисних функцій адаптованих до коваріаційної АКФ розраховується згідно виразу:

$$I = z2^k \cdot m_j.$$

де  $z$ - число ТЧБ,  $2^k$ -число рівнів квантування АКФ,  $m_j$  – число точок АКФ на інтервалі кореляції. При  $z = 6$ , числі рівнів квантування АКФ 16-256 та затуванні

її енергії на інтервалі  $m_j = 8, 12, 16, 24, 32$  об'єм пам'яті для зберігання кодів базисних функцій становить відповідно 1-32 Мбіт.

У випадку коли спектральний спецпроцесор реалізується для однієї базисної функції Фур'є, Радемахера, Крестенсона або Хаара відповідно спрощується апаратна складність та зменшується об'єм пам'яті а також перемножувач базисної функції на добуток залишків реалізується на базі схеми модульного множення з константою, яка має часову складність  $1^v$  а апаратну складність рівну сумі логічних елементів модуля  $P_i$ . Наприклад для 18 бітних спецпроцесорів у базисі Хаара-Крестенсона та базисних функцій певного числа ТЧБ апаратна складність матрично-модульних перемножувачів розраховується згідно виразу

$A = \sum_{i=1}^k P_i^2 + \sum_{i=1}^k P_i$ , а для базисних функції одного ТЧБ  $A = \sum_{i=1}^k P_i$ . Тобто апаратна

складність зменшується згідно відношення  $\frac{\sum_{i=1}^k P_i^2 + \sum_{i=1}^k P_i}{\sum_{i=1}^k P_i} = \sum_{i=1}^k P_i + 1$ . Наприклад для

18-ти розрядного спецпроцесора у базисі Хаара-Крестенсона з набором модулів 7,11,13,15,16 апаратна складність зменшується на 240240 вентилів, що приводить до відповідного зменшення апаратної складності перемножувачів кодів добутоків  $x_i x_j W_j$  спецпроцесорів Хаара-Крестенсона на 3 порядки та зменшення часової складності у 2 рази.

### 3.4 Розробка функціональної схеми спецпроцесора спектрального перетворення сигналів у базисі Хаара-Крестенсона

На основі розробленого методу спектрального перетворення Фур'є згідно модульної арифметики системи залишкових класів ТЧБ Хаара-Крестенсона запропонована функціональна-схема обчислювальних перетворень, показана на рис 3.5, і містить:

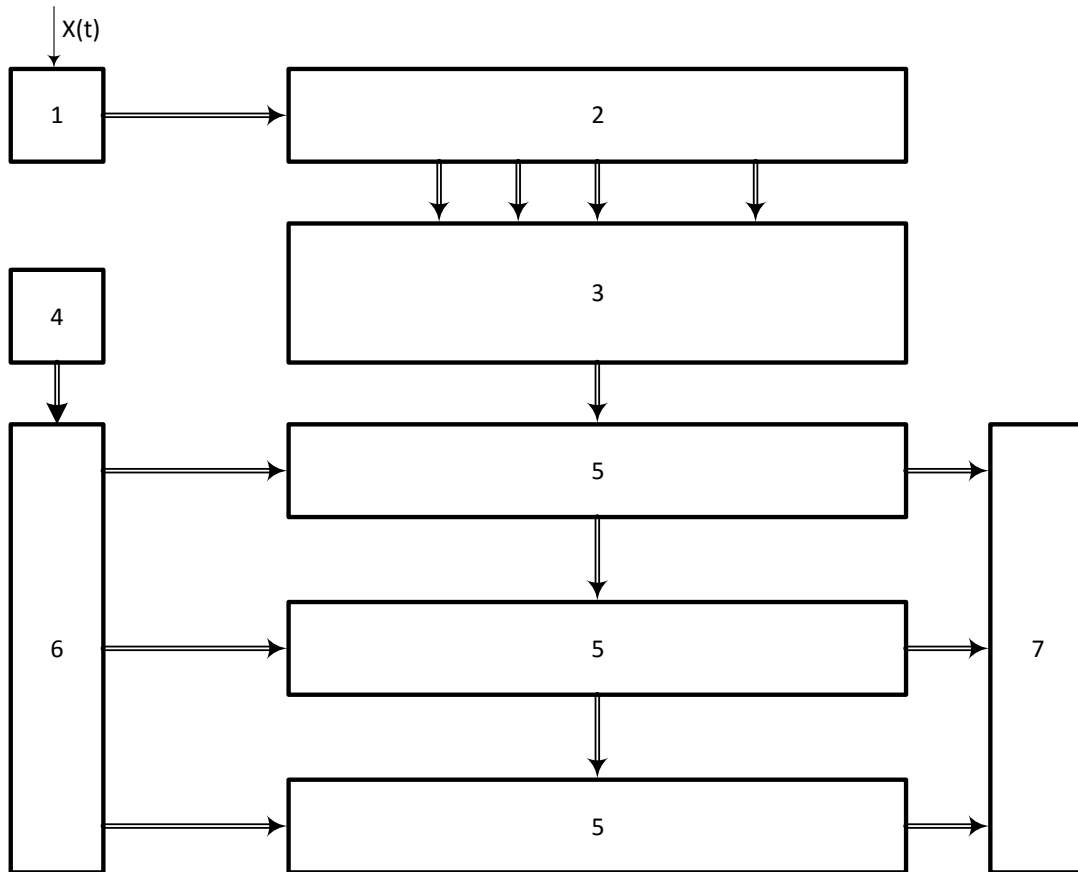


Рисунок 3.5 - Функціональна схема реалізації методу спектрального перетворення Фур'є у базисі Хаара-Крестенсона

$x(t)$  - вхідний аналоговий випадковий процес;

1- аналого-цифровий перетворювач, який формує цифрові коди  $X_i$  ( $H - C$ ) у базисі Хаара-Крестенсона;

2 – багаторозрядний регістр зсуву, у якому запам'ятовуються зсунуті у часі цифрові відліки;

3- обчислювач дискретної оцінки автоковаріаційної функції

$$K_{xx}(j) = \frac{1}{N} \sum_{i=0}^{N-1} x_i \cdot x_{i-j};$$

4- шина задання коду системи ортогональних функцій  $w$  відповідного ТЧБ;

5- пам'ять цифрових значень ортогональних функцій  $w$ , представлених у кодах Хаара-Крестенсона;

6 – обчислювач коефіцієнта взаємоковаріації між автоковаріаційною функцією  $K_{xx}(j)$  і ортогональною функцією  $w(j)$  заданого в ТЧБ;

7- вихідні шини кодів спектру косинусного (парного) перетворення у різних ТЧБ.

Розроблена функціональна структура яка реалізує запропонований метод спектрального перетворення Фур'є з адаптацією базисних функцій ТЧБ до асимптотики коваріаційної АКФ трансформується у структурну схему спектрального спецпроцесора у базисі Хаара-Крестенсона згідно запропонованих структур компонентів: АЦП, БРЗ, матрично-модульних суматорів, перемножувачів та накопичуючих суматорів у базисі Хаара-Крестенсона.

## ВИСНОВКИ ПО ТРЕТЬОМУ РОЗДІЛІ

1. Досліджений формалізований алгоритм обчислення спектру на основі центрованої та нормованої автоковаріації. Показано, що такий алгоритм характеризується значною обчислювальною складністю і не може бути ефективно застосований для побудови швидкодіючих спектральних спецпроцесорів.
2. Вперше розроблений та досліджений формалізований алгоритм обчислення спектрів на основі кореляційної функції автоковаріації з адаптацією характеристик базисних функцій різних ТЧБ до їх асимптотики.
3. Виконаний розрахунок систем взаємопростих модулів перетворення залишкових класів базису Крестенсона на основі розробленого програмного забезпечення розрахунку системи взаємопростих модулів для спектральних спецпроцесорів розрядністю 4-32 біта. Встановлено, що розрядність досліджуваного класу спецпроцесорів у базисі Радемахера-Крестенсона у порівнянні з розрядністю спецпроцесорів у базисі Радемахера зростає на 2 біти, а у Хаара-Крестенсона не більше 3-х разів.
4. На основі розробленого методу спектрального перетворення Фур'є згідно модульної арифметики системи залишкових класів ТЧБ Хаара-Крестенсона запропонована функціональна-схема обчислювальних перетворень, що трансформуються у структурну схему спектрального спецпроцесора у базисі Хаара-Крестенсона згідно розроблених структурних рішень компонентів: АЦП, БРЗ, матрично модульних суматорів, перемножувачів та накопичуючих суматорів у базисі Хаара-Крестенсона.

## РОЗДІЛ 4

РОЗРОБКА СХЕМОТЕХНІЧНИХ РІШЕНЬ ТА РЕАЛІЗАЦІЯ СПЕКТРАЛЬНИХ  
СПЕЦПРОЦЕСОРІВ4.1. Розробка структури та мікроелектронних компонентів швидкодіючого  
аналого-цифрового перетворювача у базисі Хаара-Крестенсона

На основі запропонованої реалізації компонентів швидкодіючого АЦП у базисі Хаара Крестенсона розроблена структурна схема аналого-цифрового перетворювача, з парафазними компараторами який використовується у якості базового вхідного модуля спектрального спецпроцесора перетворення Фур'є, яка показана на рис. 4.1.

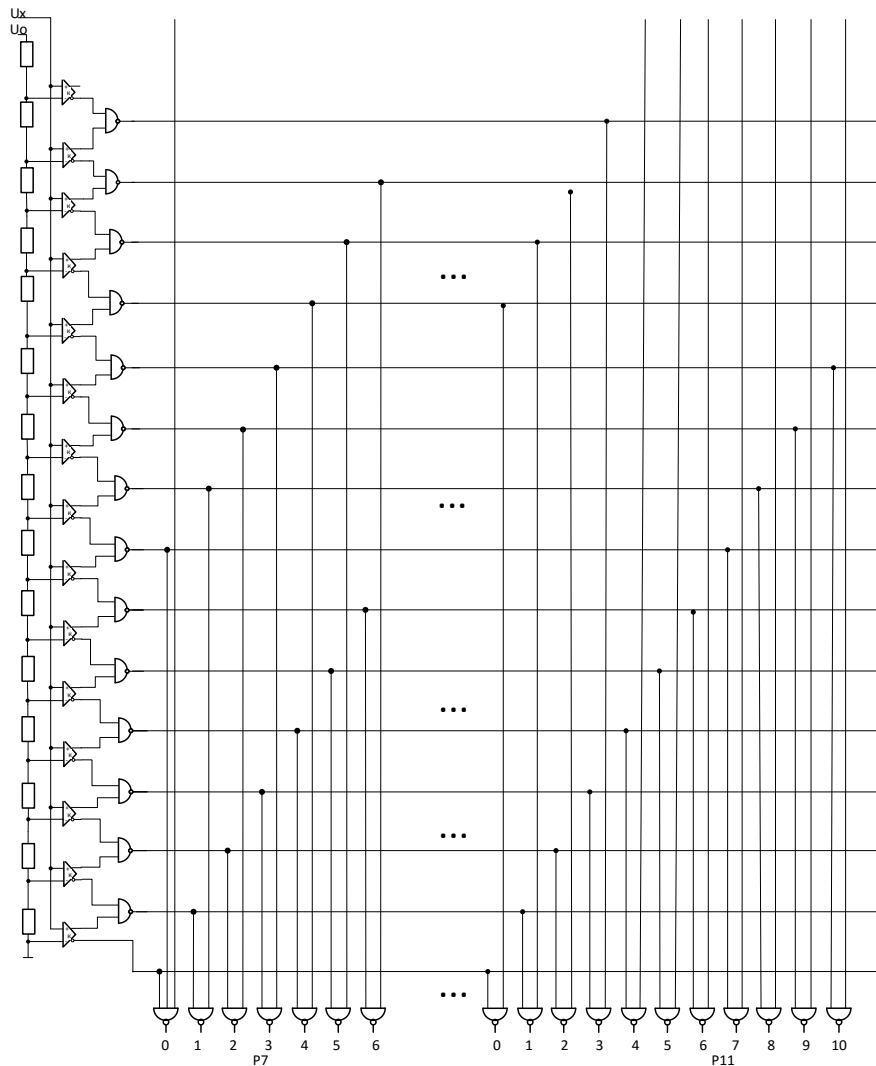


Рисунок 4.1 - Структура швидкодіючого АЦП Хаара-Крестенсона з парафазними компараторами



АЦП містить вхідну шину взірцевої потенціалу  $U_0$ , яка підключення до лінійки взірцевих резисторів  $R_i$ , і-ті з'єднання яких підключені до перших входів і-тих компараторів  $K$  з парафазними виходами, другі входи яких з'єднані з вхідною шиною аналогового сигналу  $x_i$  представленого відповідним потенціалом  $U_x$ . Інверсні виходи компараторів  $K_i$  з'єднані з першими входами і-тих логічних елементів І-НЕ, другі входи яких з'єднані з прямими виходами  $K_{i-1}$  компараторів. Виходи і-тих логічних елементів І-НЕ з'єднані з відповідними входами лінійки логічних елементів І-НЕ, які відповідають залишкам системи взаємопростих модулів  $P_i$  згідно виразу  $b_i = res_{x_i}(\text{mod } P_i)$ . Таким чином згідно запропонованої структури АЦП виконується перетворення згідно рис 2.10 аналогового сигналу  $U_x$  у цифровий парафазний унітарний код Хаара та паралельний код Хаара-Крестенсона у системі взаємопростих модулів  $P_1 \dots P_k$ .

Розрахунок часової складності такого АЦП виконується на основі аналітичного виразу  $A_{АЦП} = A_K + A_{ЛЕ} + A_{ЛЕ}$ , яка рівна  $4 \nu i$  не залежить від числа рівнів квантування та розрядності АЦП.

Розрахунок апаратної складності виконується на основі наступного аналітичного виразу  $A_{АЦП} = (A_R + A_k + A_{ЛЕ})2^k + \sum_{i=1}^k P_{i(ЛЕ)}$ ,

Де апаратна складність прецизійного резтора  $A_R = 1$ ;

компаратора  $A_K = 5 \nu$ ;

логічного елемента І-НЕ  $A_{ЛЕ} = 1 \nu$ ;

$k$ -розрядність АЦП;

Апаратна складність розробленого АЦП залежить від вибраного набору взаємопростих модулів, які показані у таблиці 4.1.

Таблиця 4.1 - Модульні набори для побудови АЦП Хаара-Крестенсона різної розрядності

Набір модулів	Кількість біт	Розроблений АЦП	Відомий АЦП
4,5	4	121	304
4,5,7	8	1808	6556

Продовження таблиці 4.1

4,5,7,9	12	28697	143360
4,5,7,9,11	16	458788	3044656

Діаграми залежності апаратної складності АЦП від їх розрядності показані на рис.4.2.

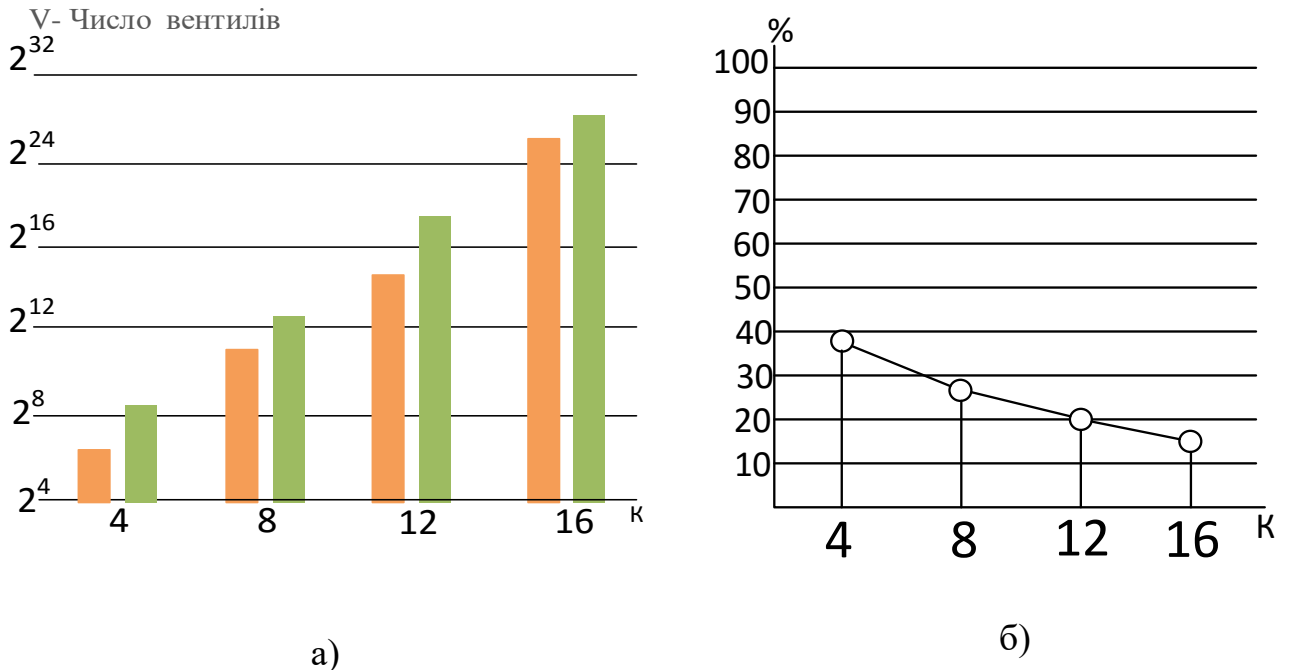


Рисунок 4.2 – Діаграма (а) та характеристика (б) апаратної складності запропонованої та відомої структури АЦП у базисі Хаара-Крестенсона при різних розрядності

З рис. 4.2. б) видно, що при зростанні розрядності запропонованої структури АЦП у базисі Хаара-Крестенсона його апаратна складність зменшується у 2-5 разів при розрядності 4-16 біт.

#### 4.2. Розробка та оцінка системних характеристик швидкодіючого спецпроцесора обчислення спектру у різних теоретико-числових базисах

Алгоритм обчислення спектру на основі косинусного перетворення Фур'є у загальному вигляді для будь-якої ортогональної функції різних ТЧБ у цифровій формі виконується згідно виразу (3.1) та алгоритму представленого на рис. 3.3.

Розроблений швидкодіючий спецпроцесор для обчислення спектрального

косинусного перетворення Фур'є у системі числення залишкових класів у базисі Хаара-Крестенсона та різних ТЧБ, в якому реалізована адаптація характеристик ортогональних базисних функцій до асимптотики автокореляційної функції кореляції. Структурна схема такого спецпроцесора показана на рис.4.3.

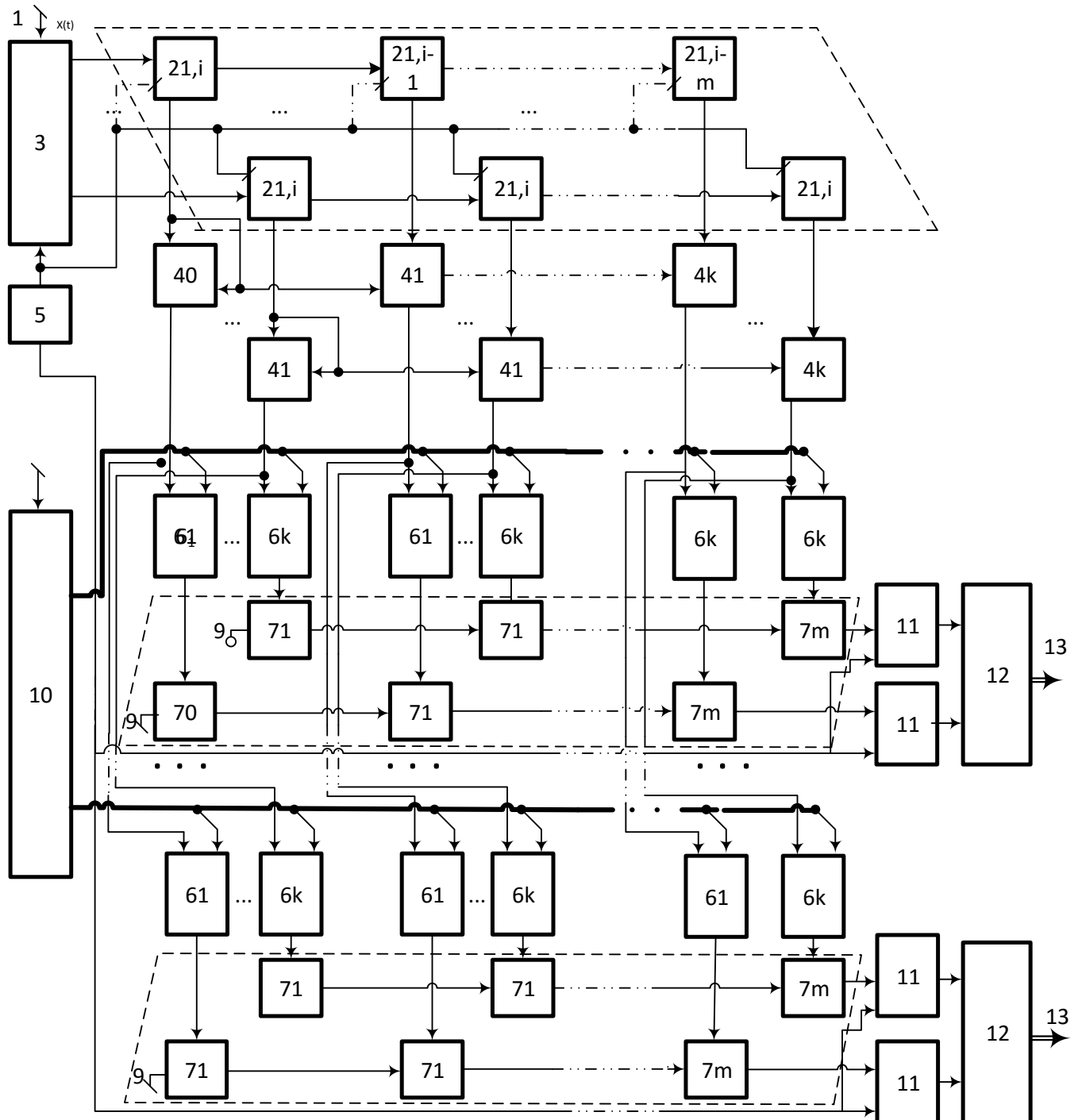


Рисунок 4.3. Структурна схема спецпроцесора обчислення спектру косинусного перетворення Фур'є

На рис 4.3 показана структурна схема спецпроцесора, який реалізує обчислення спектрального косинусного перетворення, де: 1- вхід аналогового сигналу, 2 – блок пам'яті, 3- АЦП паралельного типу з вихідним кодом базису Хаара-Крестенсона, 4 – модульні перемножувачі, 5 - генератор імпульсів, 6 – вентильні матриці перемноження по модулю, 7 –модульні матричні суматори, 8 – вхід вибору базисної функції ТЧБ, 9 – вхід матричного суматора нульового каналу, 10 – пам'ять кодів базисних функцій, 11 – накопичувальний модульний суматор; 12- дешифратор; 13 - вихідні шини кодів спектру.

Процесор працює таким чином.

Вхідний аналоговий сигнал  $x(t)$  в АЦП 3 дискретизується в часі, квантується по амплітуді і перетворюється у цифрові розрядно-позиційні коди Хаара-Крестенсона по кожному взаємопростому модулю системи залишкових класів базису Крестенсона. Отримувані коди записуються в блок пам'яті, який виконує функції багаторозрядного регістру зсуву 2, де формується масив зсунутих даних  $(x_i, x_{i-1}, \dots, x_{i-m})$ , де  $m$  - число точок АКФ, а  $k$  – число модулів СЗК, на основі яких в режимі on-line в матричних модульних перемножувачах 4, обчислюються добутки  $C_j = x_i \cdot x_{i-j} \pmod{p}$ , які додатково перемножуються в модульних перемножувачах 6  $C_j \cdot W_j \pmod{p}$  з цифровими кодами відповідних точок базисних ортогональних функцій  $W_j$ , які зберігаються в пам'яті 10. Отримані добутки послідовно додаються в матрично-модульних суматорах 7 та накопичуючому суматорі 11, вихідні коди яких представляють цифрові значення спектру у вигляді залишків по кожному модулю  $P_i$ , які перетворюються у двійковий коди базису Радемахера у дешифраторах 12 і подаються на відповідні вихідні шини 13.

Запропонований спецпроцесор характеризується розширеними функціональними можливостями за рахунок того, що в якості базисних функцій  $W_j$ , які представлені в залишках по модулю  $P_i$ , можуть бути відповідно обчислені спектри у базисах Фур'є, Радемахера, Крестенсона та інших. Матричні перемножувачі 4, 6 та суматори 7, 11 реалізуються на вентильних матрицях (рис. 2.25, 2.31, 3,4).

Швидкодія запропонованого спецпроцесора визначається сумарною затримкою сигналів у послідовно з'єднаних компонентах згідно виразу.

$$\tau_w = (\tau_{АЦП} + 2\tau_x + m\tau_\Sigma + \tau_{ш})\nu,$$

де:

$\tau_w$  - швидкодія спецпроцесора спектрального косинусного перетворення Фур'є;

$\nu$  - тривалість переключення мікроелектронного вентиля ПЛМ або ПЛІС;

$\tau_{АЦП} = 5\nu$  - часова складність розробленого АЦП у базисі Хаара-Крестенсона;

$\tau_x = 2\nu$  - часова складність матрично-модульного перемножувача;

$m\tau_\Sigma = (m + 2) \cdot 2\nu$  - часова складність лінійки модульно-матричних суматорів з врахуванням нульового та накопичувальних суматорів 11;

$\tau_{ш} = 2\nu$  - часова складність шифратора 12.

При  $m = 32$  - часова складність спектрального спецпроцесора складає  $24 + 64\nu = 88\nu$ , тобто не перевищує 100 мікротактів.

При швидкодії мікроелектронних вентилів ПЛІС не менше 100 МГц – обчислення спектру для одної базисної функції складає не менше  $10^6$  обчислень в секунду.

При реалізації 64 точок одної базисної функції швидкодія обчислень спектру не менша  $0,5 \cdot 10^6$  обчислень за секунду.

При обчисленні спектру в 6-ти теоретико-числових базисах не буде менша  $10^5$  обчислень в секунду.

Таким чином швидкодія розроблено спецпроцесора спектрального косинусного перетворення Фур'є у різних ТЧБ на 1-2 порядки перевищує швидкодію відомих, тиражованих спектральних аналізаторів, які мають обмежені функціональні можливості, оскільки можуть обчислювати спектри тільки в базисі гармонічних функцій Фур'є. Обчислення спектрів в інших ТЧБ Радемахера, Уолша Хаара, Галуа та Крестенсона дозволяє значно спростити генерацію цифрових

значень базисних функцій спектрального перетворення, зменшити об'єм пам'яті для зберігання їх адаптованих цифрових відліків, які враховують ступінь затухання енергії автоковаріаційних функцій, що дозволяє виключити одну з трьох операцій множення у процесорі обчислення спектру. Відомий недолік рядів Фур'є, які характеризуються низькою сходимістю потребує обчислення на порядок більше спектральних ліній у порівнянні з числом ортогональних функцій Радемахера та Крестенсона, що також забезпечує відповідне прискорення спектрального діагностування характеристик промислових об'єктів та маніпульованих сигналів, які потребують спектрального налізу сигналів у діапазоні частот стандартних протоколів комп'ютерних мереж 10-100МГц та спеціалізованих системах передавання даних.

#### 4.3 Розробка програмного продукту для розрахунку наборів взаємопростих модулів спектрального спецпроцесора різної розрядності

Розроблені оболонки і програми розрахунку наборів взаємопростих модулів для процесорів різної розрядності виконані на мові програмування C#.

На рис. 4.4 показано створення проекту програми розрахунку взаємопростих модулів, а також що даний продукт працюватиме на Framework 4.5.1.

Для розробки програмного продукту обрано Framework 4.5.1, так як основні нововведення у даній версії торкнулись інструментарію налагоджування та покращення продуктивності застосунків, а саме:

- Налагоджування (debugging):
  - Підтримка Edit & Continue для 64-бітної архітектури;
  - Підтримка обриву та збою зв'язку в ADO.NET;
  - Покращення асинхронного налагоджування;
- Оптимізація продуктивності застосунків:
  - Механізм стиснення великих об'єктів у пам'яті по запиту;

- Вдосконалення системи побудови програм Just-In-Time для підтримки багатьох платформ;
- Новий функціонал ASP.NET App Suspend, який дозволяє автоматично вимикати неактивні сайти та завантажувати їх по запиту.

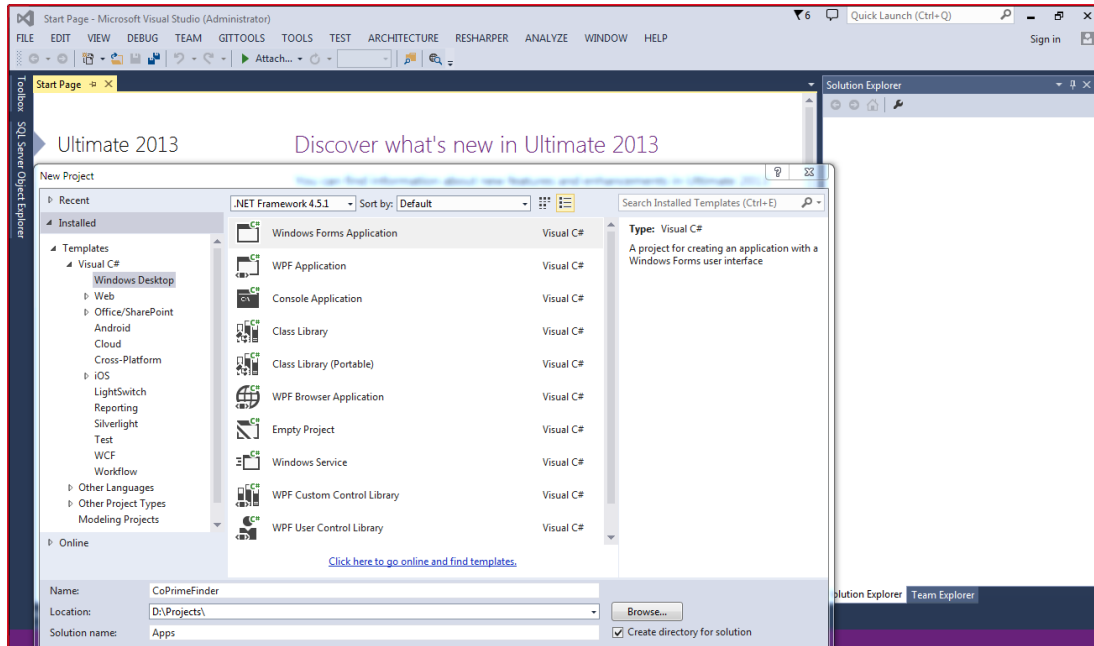


Рисунок 4.4 – Проект програми розрахунку взаємопростих модулів

На рис. 4.5 зображено створення основного фрейму програми.

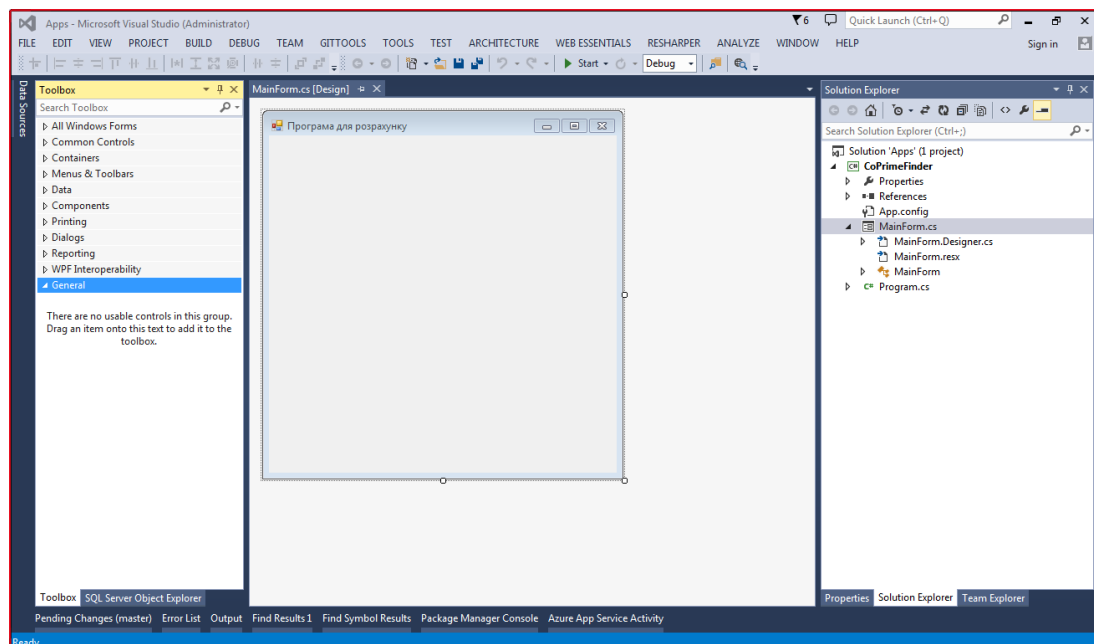


Рисунок 4.5 – створення головного вікна програми

Фрагмент коду проектування головного меню програми, який складається із фреймів:

namespace KrestensonMath.CoPrimesFinderApp

```
{
    partial class MainForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.tabControll = new
System.Windows.Forms.TabControl();
            this.tabPage1 = new System.Windows.Forms.TabPage();
            this.chkOnlyCoprime = new
System.Windows.Forms.CheckBox();
            this.chkFullInfo = new System.Windows.Forms.CheckBox();
            this.nudAdv = new System.Windows.Forms.NumericUpDown();
            this.label1 = new System.Windows.Forms.Label();
            this.nudMin = new System.Windows.Forms.NumericUpDown();
            this.nudMax = new System.Windows.Forms.NumericUpDown();
            this.txtResult = new System.Windows.Forms.TextBox();
            this.lblMin = new System.Windows.Forms.Label();
            this.lblMax = new System.Windows.Forms.Label();
            this.btnFind = new System.Windows.Forms.Button();
            this.tabPage2 = new System.Windows.Forms.TabPage();
            this.chkFullDataFilter = new
System.Windows.Forms.CheckBox();

```



```

        this.chkGreaterCpu = new
System.Windows.Forms.CheckBox();
        this.txtInNumbersFilter = new
System.Windows.Forms.TextBox();
        this.txtOutNumbersFilter = new
System.Windows.Forms.TextBox();
        this.btnFindMin = new System.Windows.Forms.Button();
        this.tabControll.SuspendLayout();
        this.tabPage1.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.nudAdv)).BeginInit(
);

((System.ComponentModel.ISupportInitialize)(this.nudMin)).BeginInit(
);

        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
13F);
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(497, 473);
        this.Controls.Add(this.tabControll);
        this.Name = "MainForm";
        this.Text = "Розрахунок наборів взаємопростих модулів";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.tabControll.ResumeLayout(false);
        this.tabPage1.ResumeLayout(false);
        this.tabPage1.PerformLayout();
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(497, 473);
        this.Controls.Add(this.tabControll);
        this.Name = "MainForm";
        this.Text = "Розрахунок наборів взаємопростих модулів";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.tabControll.ResumeLayout(false);
        this.tabPage1.ResumeLayout(false);
        this.tabPage1.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.nudAdv)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.nudMin)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.nudMax)).EndInit();
        this.tabPage2.ResumeLayout(false);
        this.tabPage2.PerformLayout();
        this.ResumeLayout(false);
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(497, 473);

```

```

        this.Controls.Add(this.tabControl1);
        this.Name = "MainForm";
        this.Text = "Розрахунок наборів взаємопростих модулів";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.tabControl1.ResumeLayout(false);
        this.tabPage1.ResumeLayout(false);
        this.tabPage1.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.nudAdv)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.nudMin)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.nudMax)).EndInit();
        this.tabPage2.ResumeLayout(false);
        this.tabPage2.PerformLayout();
        this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.TabControl tabControl1;
private System.Windows.Forms.TabPage tabPage1;
private System.Windows.Forms.CheckBox chkOnlyCoprime;
private System.Windows.Forms.CheckBox chkFullInfo;
private System.Windows.Forms.NumericUpDown nudAdv;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.NumericUpDown nudMin;
private System.Windows.Forms.NumericUpDown nudMax;
private System.Windows.Forms.TextBox txtResult;
private System.Windows.Forms.Label lblMin;
private System.Windows.Forms.Label lblMax;
private System.Windows.Forms.Button btnFind;
private System.Windows.Forms.TabPage tabPage2;
private System.Windows.Forms.TextBox txtInNumbersFilter;
private System.Windows.Forms.TextBox txtOutNumbersFilter;
private System.Windows.Forms.Button btnFindMin;
private System.Windows.Forms.CheckBox chkGreaterCpu;
private System.Windows.Forms.CheckBox chkFullDataFilter;

    }
}

```

Інтерфейс програми складається з декількох фреймів. Перший з них — це головне вікно програми (рис.4.6). З нього відбувається вхід у режим розрахунку взаємопростих модулів.

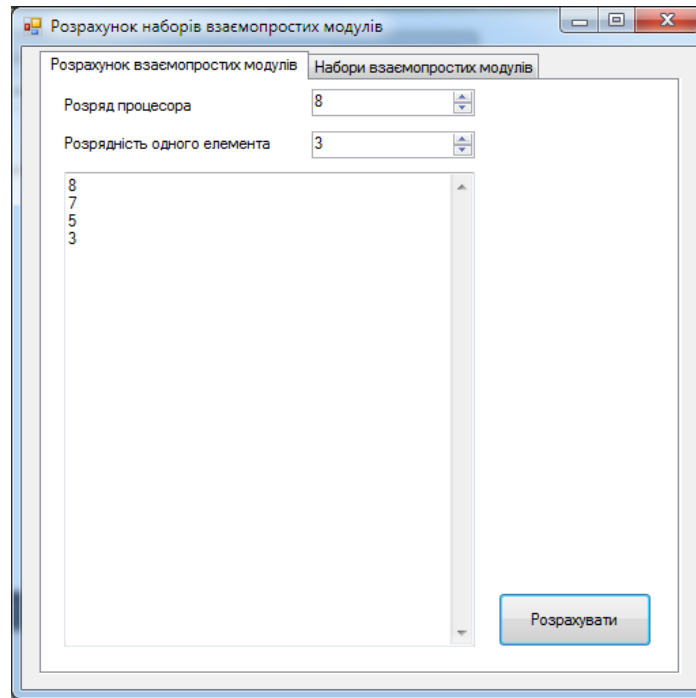


Рисунок 4.6 - Розрахунок взаємопростих модулів

На рис 4.6 зображено головне вікно програми де потрібно вказати розрядність процесора та кількість біт у одному модулі, після чого потрібно натиснути клавішу розрахувати.

Рис. 4.7 показує розрахунок взаємопростих 4-х бітних модулів для процесорів розрядністю 16 .

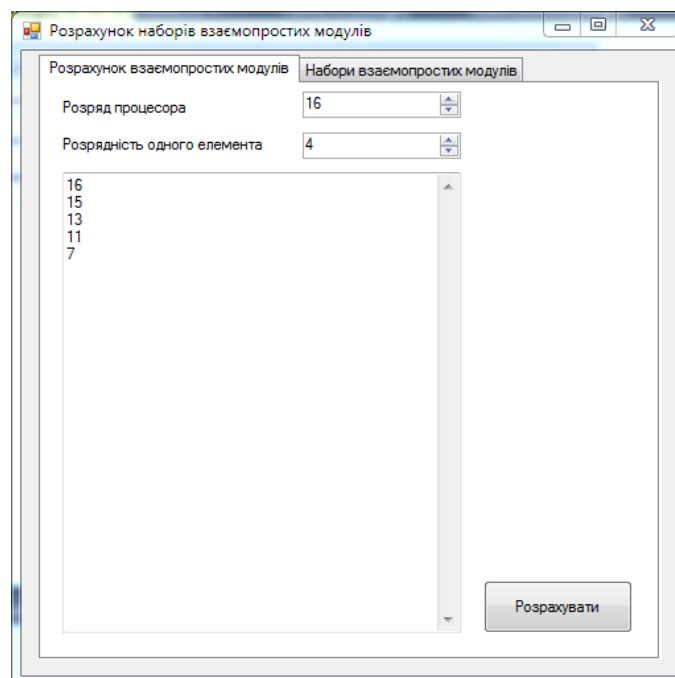


Рисунок 4.7 - Розрахунок взаємопростих модулів для процесора розрядністю  
16 біт

Рисунок 4.8 показує розрахунок взаємопростих модулів для процесорів розрядністю 32 біт з 5-ти бітними модулями.

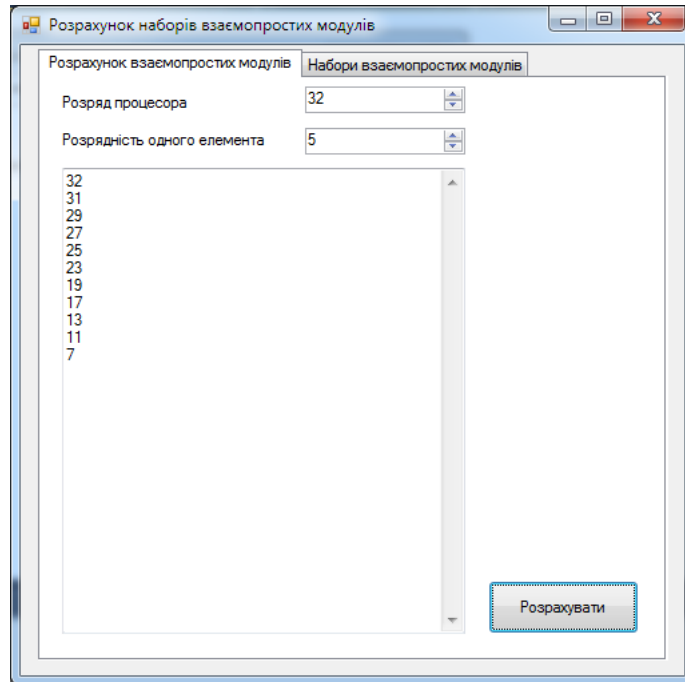


Рисунок 4.8 - Розрахунок взаємопростих модулів для процесора розрядністю 32 біт та 5-ти бітними модулями

Лістинг коду для розрахунку модулів:

```
ExtendedNumberMultiplier(Utils.GetMultipliers(maxValue));
    foreach (var pair in primesList)
    {
        if (pair.Key.Length < itemMaxCapacity)
        {
            searchList.Add(pair.Key, pair.Value);
        }
        else if (pair.Key.Length == itemMaxCapacity)
        {
            if (res.Contains(pair.Value))
                continue;
            res.Add(pair.Value);
            res2.Add(new ExtendedNumberPrime(pair.Value));
        }
    }
else
```

```

        {
            break;
        }
    }

    var needContinue = res.Count < returnPrimesCount;
    if (needContinue)
    {
        var tempRes = new List<uint>();

        foreach (var pairOuter in searchList)
        {
            uint temp;
            foreach (var pairInner in searchList)
            {
                if (pairOuter.Key.Length +
pairInner.Key.Length - 1 > itemMaxCapacity)
                {
                    continue;
                }

                temp = pairInner.Value * pairOuter.Value;

                if (temp >= maxValue ||
tempRes.Contains(temp) || res.Contains(temp))
                    continue;
                tempRes.Add(temp);
                res2.Add(new
ExtendedNumberMultiplier(new[] { pairInner.Value, pairOuter.Value
}));
            }

            if (!tempRes.Contains(pairOuter.Value))
            {
                tempRes.Add(pairOuter.Value);
                res2.Add(new
ExtendedNumberPrime(pairOuter.Value));
            }
        }
    }
}

```

```

        for (var i = 2; i <= itemMaxCapacity; i++)
        {
            temp = (uint)Math.Pow(pairOuter.Value, i);
            if (temp > maxValue)

                {
                    break;
                }

            if (tempRes.Contains(temp))
                continue;
            tempRes.Add(temp);
            res2.Add(new
ExtendedNumberPowerer(pairOuter.Value, (uint)i));
        }

        res.AddRange(tempRes);
    }

    res2 = (from entry in res2 orderby entry.GetRealNumber()
descending select entry).ToList();
    return res2.Take(returnPrimesCount).ToList();

using System.Globalization;
using System.Linq;

namespace KrestensonFindPrimes
{
    public abstract class ExtendedNumber
    {
        protected uint RealNumber;
        protected uint[] Multipliers;

        public uint GetRealNumber()
        {
            return RealNumber;
        }

        public uint[] GetMultipliers()
        {
            return Multipliers;
        }
    }
}

```

}

}

Даний розрахунок відбувається по принципу, який зображений на рис 4.9.

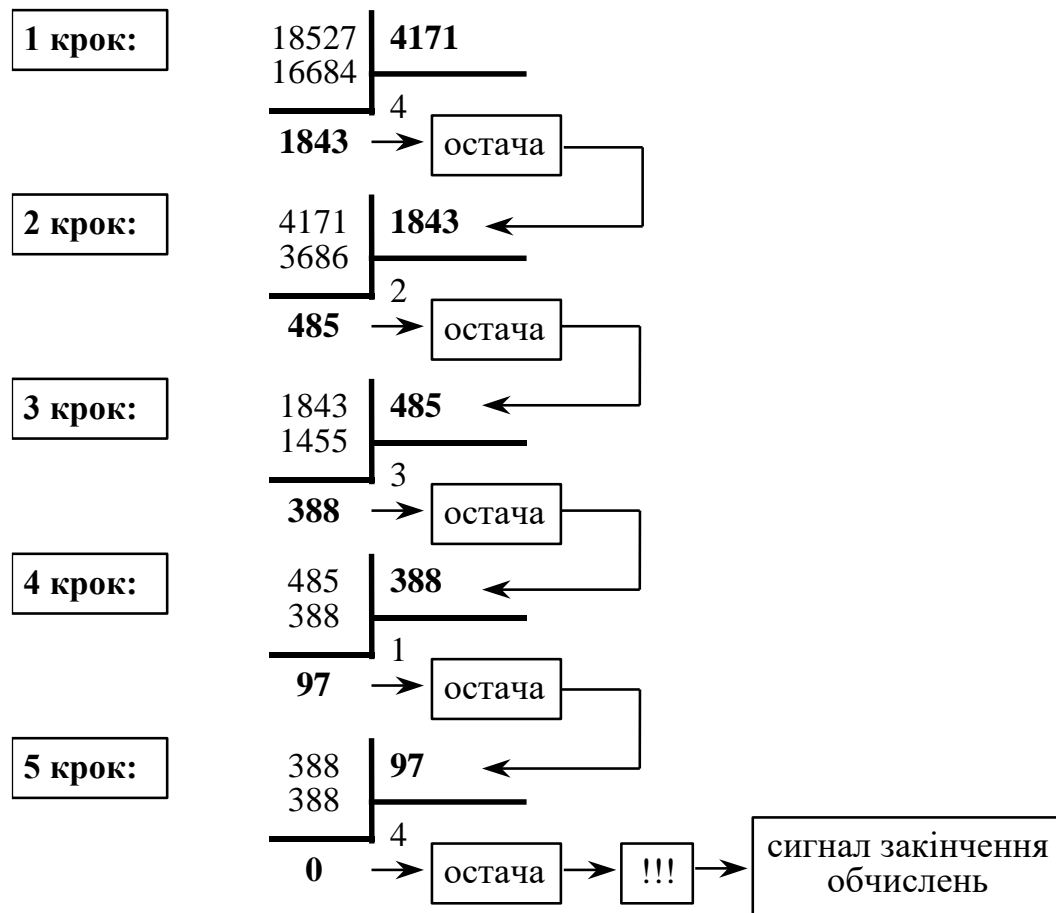


Рисунок 4.9 – Принцип обчислення взаємпростих модулів для процесора різної розрядності

Для розрахунку кодів спектральних ліній на виході спецпроцесора потрібно набори модулів  $P_i$ , тому в програмному продукті розрахунку системи взаємпростих модулів розроблена можливість розрахунку шляхом задання максимально можливого набору модулів.

Для того щоб розрахувати набори взаємпростих модулів потрібно перейти у вкладку «набори взаємпростих» модулів, де потрібно вказати з яких модулів формуватимуться набори рис 4.10. Модулі вказуємо з першої вкладки даного програмного продукту.

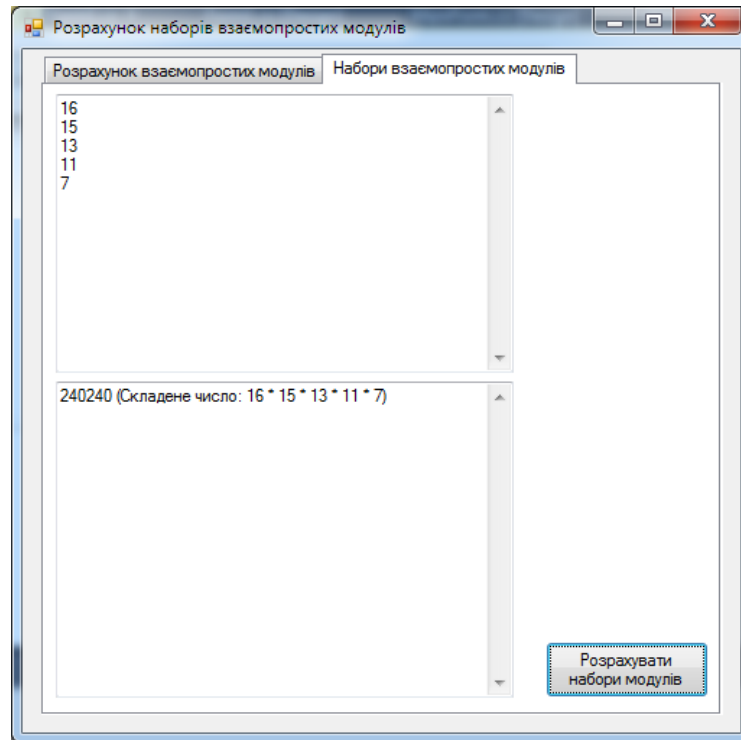


Рисунок 4.10 - Розрахунок наборів взаємопростих модулів для процесора розрядністю 16 біт та 4-х бітними модулями

На рис 4,11 показано розрахунок набору модулів для процесора з розрядністю 16 біт та 4- бітними модулями.

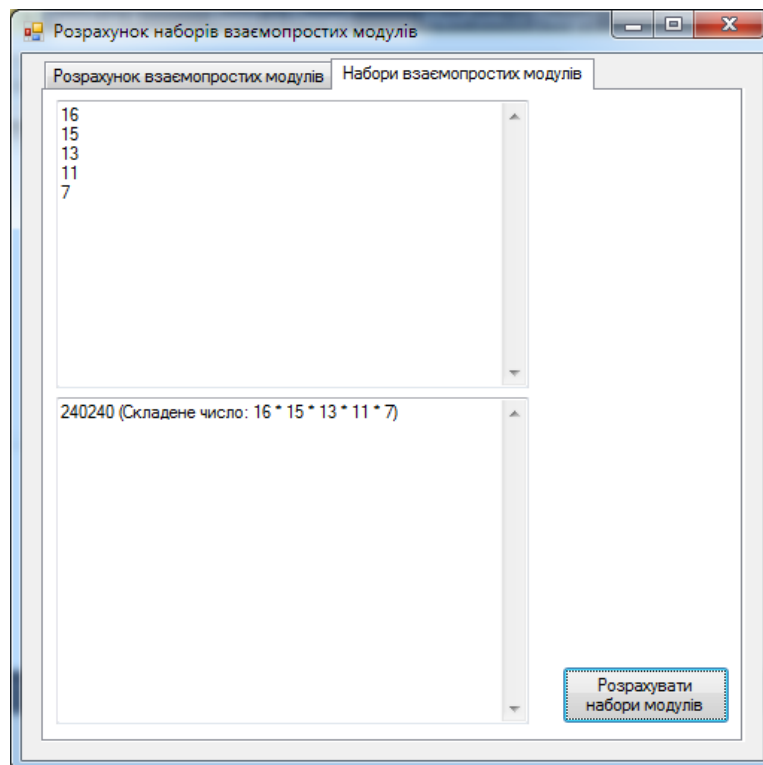


Рисунок 4.11 - Розрахунок наборів взаємопростих модулів для процесора розрядністю 32 біта та 4-ти бітними модулями



На рис 4.12 показано розрахунок набору модулів для процесора з розрядність 32 біт, відповідно у першому вікні задавши 5-ти бітні модулі та натиснувши клавішу «розрахувати набори модулів», де у наступному вікні програми відображається процес розрахунку а також всі можливі комбінації наборів модулів.

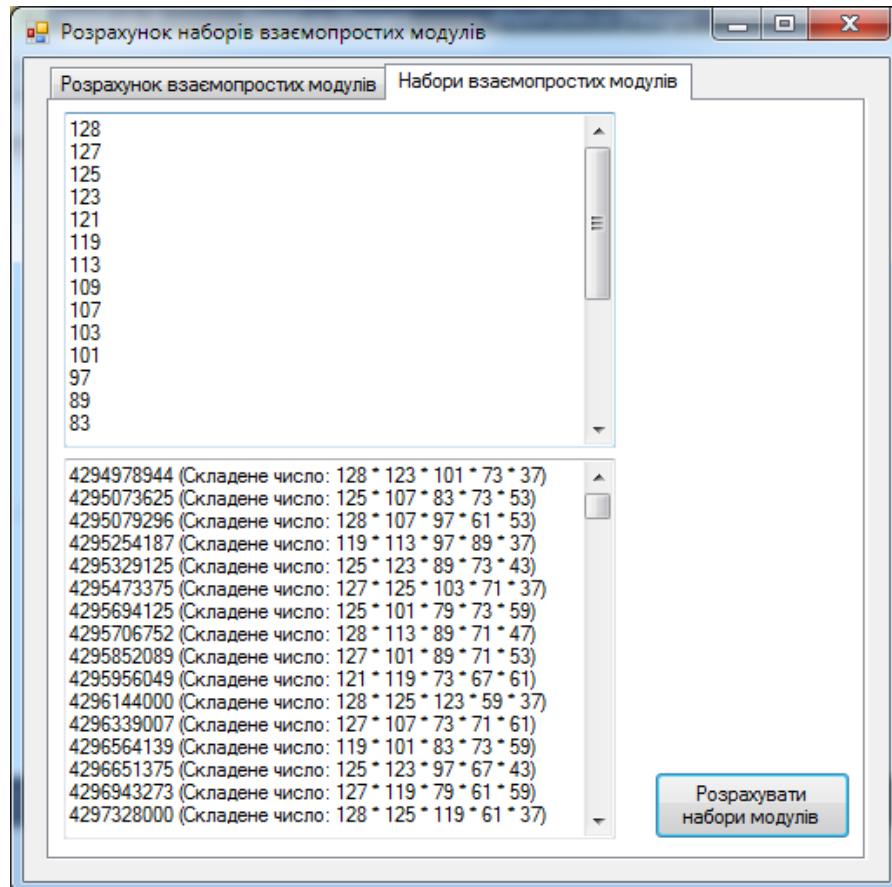


Рисунок 4.12 - Розрахунок наборів взаємопростих модулів для процесора розрядністю 32 біт та 5-ти бітними модулями

Фрагмент коду рзрахунку набору взаємопростих модулів :

```
public class ExtendedNumberPrime : ExtendedNumber
{
    public ExtendedNumberPrime(uint number)
    {
        RealNumber = number;
        Multipliers = new uint[] { number, 1 };
    }
}
```

```

public override string ToString()
{
    return RealNumber + " (Просте число)";
}

public ExtendedNumberPrime(uint number)
{
    RealNumber = number;
    Multipliers = new uint[] { number, 1 };
}

public override string ToString()
{
    return RealNumber + " (Просте число)";
}
}

public class ExtendedNumberMultiplicator : ExtendedNumber
{

    public ExtendedNumberMultiplicator(uint[] multipliers)
    {
        var number = multipliers.Aggregate<uint, uint>(1,
(current, multiplier) => current*multiplier);
        RealNumber = number;

        Multipliers = multipliers;
    }
    public ExtendedNumberMultiplicator(uint[] multipliers)
    {
        var number = multipliers.Aggregate<uint, uint>(1,
(current, multiplier) => current*multiplier);
        RealNumber = number;
    }

    public override string ToString()
    {

```

```

        public ExtendedNumberMultiplier(uint[]
multipliers)
    {
        var number = multipliers.Aggregate<uint, uint>(1,
(current, multiplier) => current*multiplier);
        RealNumber = number;

        Multipliers = multipliers;
    }
    public ExtendedNumberMultiplier(uint[] multipliers)
    {
        var number = multipliers.Aggregate<uint, uint>(1,
(current, multiplier) => current*multiplier);
        RealNumber = number;

        Multipliers = multipliers;
    }

    var res =
RealNumber.ToString(CultureInfo.InvariantCulture);
    res += " (Складене число: ";
    res += string.Join(" * ", Multipliers);
    res+=")";

return res;

    }

}

public class ExtendedNumberPowerer : ExtendedNumberMultiplier
{
    private readonly uint _powerKoefficient;
    public ExtendedNumberPowerer(uint multiplier, uint
powerKoefficient)
        : base(Enumerable.Repeat(multiplier,
(int)powerKoefficient).ToArray())
    {
        _powerKoefficient = powerKoefficient;
    }

    public override string ToString()

```

```

        {
            return RealNumber + " (Складене число: " +
Multipliers[0]
+
            " ^ " + _powerKoefficient + ")";
        }
    }
}

```

Таким чином дане, програмне забезпечення для розрахунку модулів та їх розраховувані набори модулів можна використовувати для розрахунку кодів спектральних ліній з використанням спектрального спецпроцесора у базисі Хаара-Крестенсона різної розрядності. Код програми на мові С# наведено у Додатку А.

#### 4.4 Розробка програмного забезпечення синтезу мікроелектронної реалізації компонентів спектрального спецпроцесора на програмованих логічних інтегральних схемах

Система автоматизованого проектування Xilinx Ise 10.3. являє собою інтегроване середовище для розробки цифрових пристроїв на базі програмованих логічних інтегральних схем фірми Xilinx, яка забезпечує виконання всіх етапів, необхідних для випуску готових виробів:

- 1) створення логічних пристроїв;
- 2) створення синтезу структур і прокладка внутрішніх зв'язків в програмованих логічних та інтегральних схемах;
- 3) програмованих логічних інтегральних схем (компіляцію);
- 4) моделювання функціональності та часового аналізу логічних інтегральних схем.

Проектування дослідного взірця спектрального спецпроцесора в ТЧБ Хаара-Крестенсона включає наступні етапи:

- розробка загальної логічної схеми спектрального спецпроцесора (рис. 4.13);

- розробка логічних блоків компонентів спектрального спецпроцесора, що дублюються  $m+1$  – разів у багаторозрядному регістрі зсуву (рис. 4.14);
- розробка блока логічних елементів “I-HE” (рис. 4.15);
- розробка блоку накопичуючого суматора (рис. 4.16);

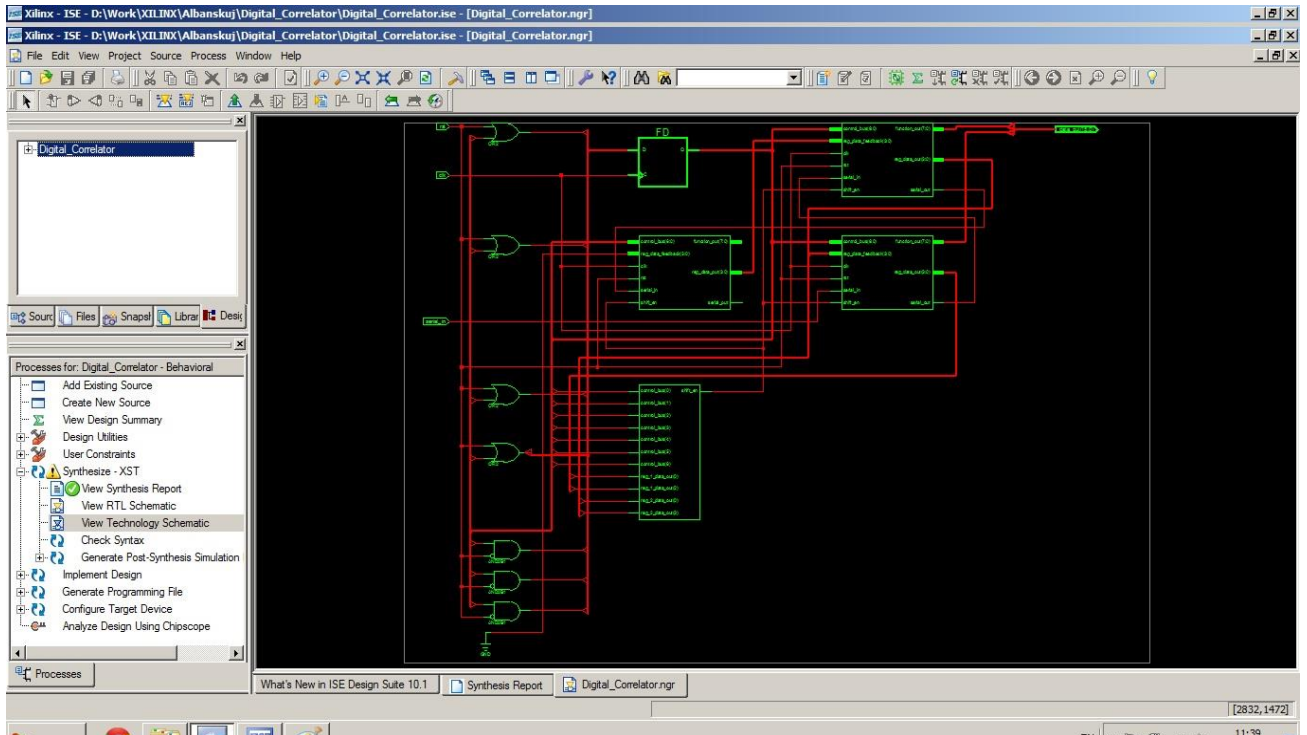


Рисунок 4.13 – Загальна логічна схема спецпроцесора

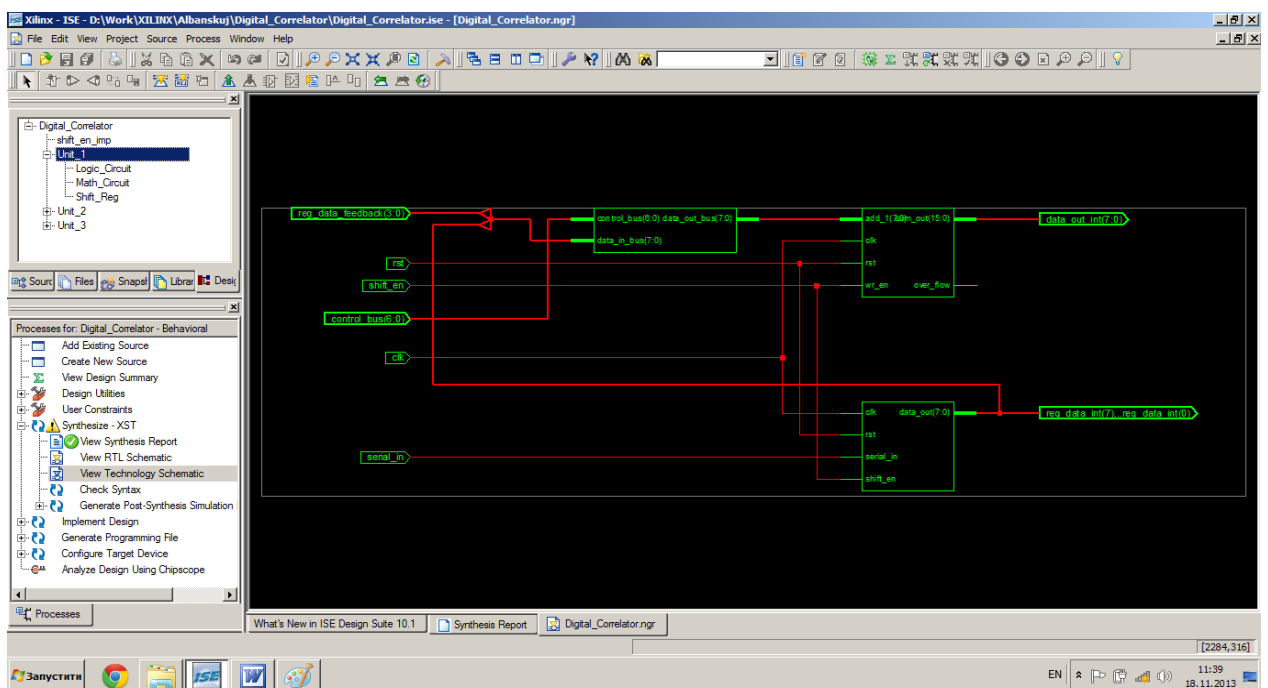


Рисунок 4.14 - Логічний блок багаторозрядного регістра зсуву спектрального спецпроцесора

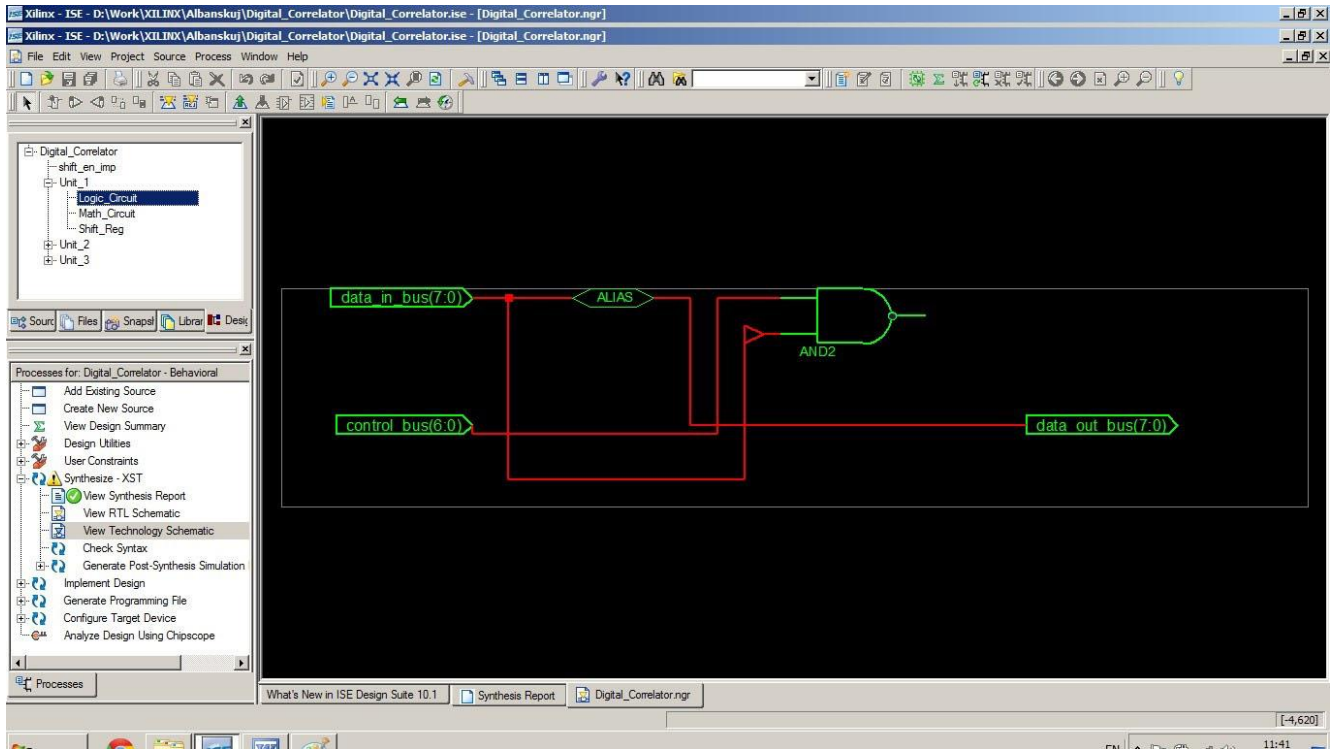


Рисунок 4.15 – Блок логічних елементів “І-НЕ”

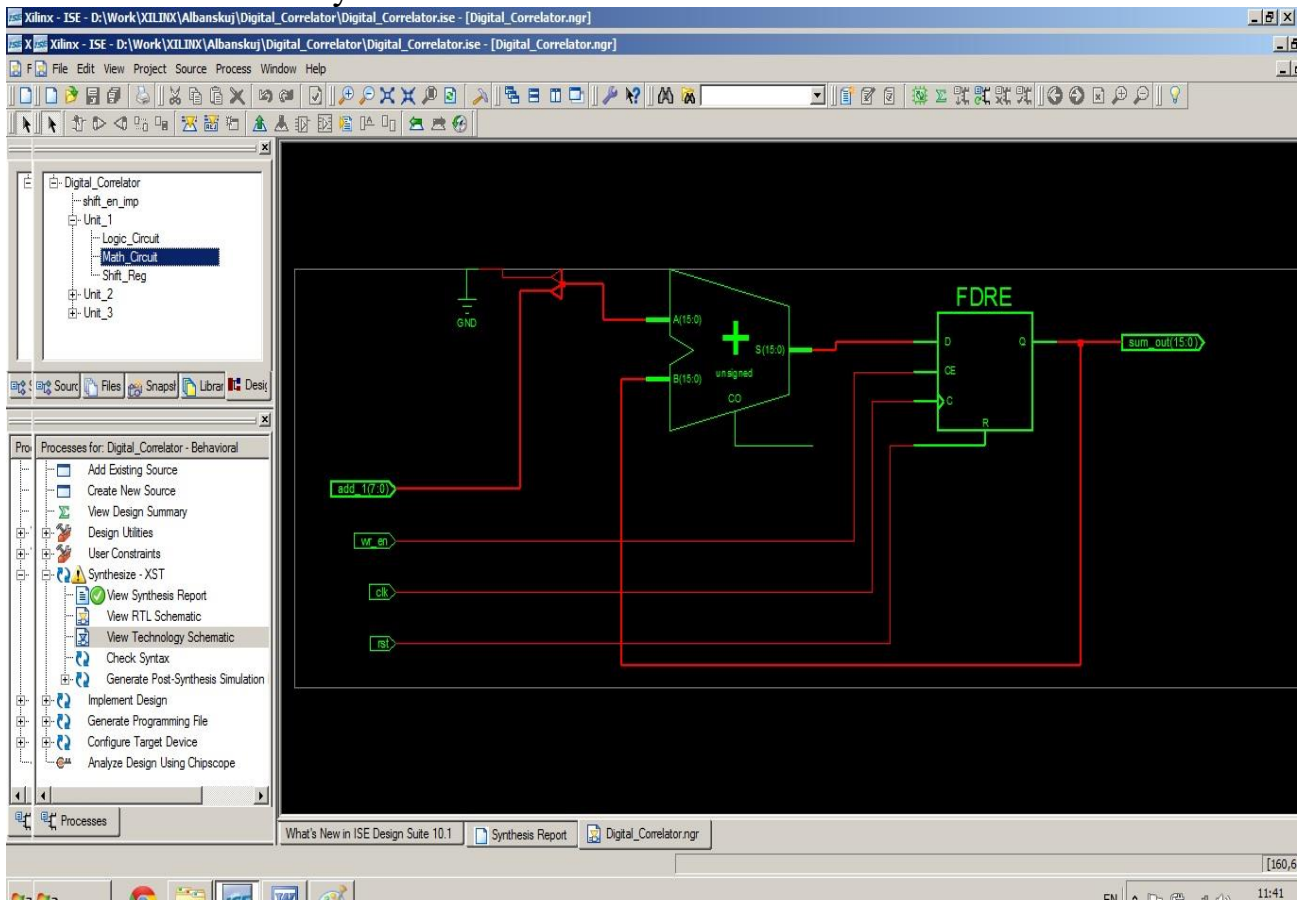


Рисунок 4.16 – Блок накопичуючого суматора

Для коректної роботи спектрального спецпроцесора відбувається дублювання його основних блоків компонент  $m$  та  $m+1$  разів: накопичуючого суматора, групи (блоку) логічних елементів “I-HE”, розрядів багаторозрядного регістра зсуву.

Таким чином в процесі проектування пристрою використано ряд компонентів з бібліотек програмного комплексу Xilinx Ise 10.3. Статистика використаних компонентів Ise 10.3. приводиться в наступному вигляді:

### 1. Macro Statistics (макро статистика) -

# Adders/Subtractors	: 3
16-bit adder carry out	: 3
# Registers	: 28
1-bit register	: 24
16-bit register	: 3
7-bit register	: 1

### 2. Design Statistics (статистика проектування) -

# IOs	: 19
Cell Usage :	
# BELS	: 51
# GND	: 1
# LUT2	: 3
# LUT3	: 16
# LUT4	: 1
# LUT4_L	: 1
# MUXCY	: 14
# VCC	: 1
# XORCY	: 14
# FlipFlops/Latches	: 34
# FDE	: 1
# FDR	: 3
# FDRE	: 26
# FDS	: 4
# Shift Registers	: 1
# SRL16E	: 1
# Clock Buffers	: 1
# BUFGP	: 1
# IO Buffers	: 18
# IBUF	: 2
# OBUF	: 16
# IOs	: 19
Cell Usage :	
# BELS	: 51
# IOs	: 19
Cell Usage :	
# BELS	: 51
# IOs	: 19
Cell Usage :	

```

# BELS : 51
# GND : 1
# LUT2 : 3
# LUT3 : 16
# LUT4 : 1
# LUT4_L : 1
# MUXCY : 14
# VCC : 1
# XORCY : 14
# FlipFlops/Latches : 34
# FDE : 1
# FDR : 3
# FDRE : 26
# FDS : 4
# Shift Registers : 1
# SRL16E : 1
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 18
# IBUF : 2
# OBUF : 16
# IOs : 19
Cell Usage :
# BELS : 51
# IOs : 19
Cell Usage :
# BELS : 51

# IOs : 19
Cell Usage :
# BELS : 51
# GND : 1
# LUT2 : 3
# LUT3 : 16
# LUT4 : 1
# LUT4_L : 1
# MUXCY : 14
# VCC : 1
# XORCY : 14
# FlipFlops/Latches : 34
# FDE : 1
FDS : 4
# Shift Registers : 1
# SRL16E : 1
# Clock Buffers : 1
# BUFGP : 1

```



```

# IOs : 19
Cell Usage :
# BELS : 51
# GND : 1
# LUT2 : 3
# LUT3 : 16
# LUT4 : 1
# LUT4_L : 1
# MUXCY : 14
# VCC : 1
# XORCY : 14
# FlipFlops/Latches : 34
# FDE : 1
# FDR : 3
# FDRE : 26
# FDS : 4
# Shift Registers : 1
# SRL16E : 1
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 18
# IBUF : 2
# OBUF : 16

```

### 3. Device utilization summary (резюме використаних пристроїв):

```

Selected Device : 3s200ft256-4
Number of Slices: 19 out of 1920 0%
Number of Slice Flip Flops: 34 out of 3840 0%
Number of 4 input LUTs: 22 out of 3840 0%
  Number used as logic: 21
  Number used as Shift registers: 1
Number of IOs: 19
Number of bonded IOBs: 19 out of 173 10%
Number of GCLKs: 1 out of 8 12%

```

### 4. Timing constraint: Default period analysis for Clock 'clk' (часові рамки виконання)-

Clock period: 6.383ns (frequency: 156.662MHz)

**Data Path:** Unit\_3/Shift\_Reg/data\_tmp\_1 to Unit\_3/Shift\_Reg/data\_tmp\_4

Cell:in->out	Gate	Net	fanout	Delay	Delay	Logical Name (Net Name)
FDRE:C->Q	5	0.720	1.260	Unit_3/Shift_Reg/data_tmp_1		
(Unit_3/Shift_Reg/data_tmp_1)						
LUT3:I0->O	1	0.551	0.869	shift_en2 (shift_en2)		
LUT4:I2->O	28	0.551	1.830	shift_en22 (shift_en)		
FDRE:CE		0.602		Unit_3/Shift_Reg/data_tmp_4		

**Total** 6.383ns (2.424ns logic, 3.959ns route)  
(38.0% logic, 62.0% route)

Лістинг коду програмованого спектрального спецпроцесора наведено у Додатку Б.

## ВИСНОВКИ ПО ЧЕТВЕРТОМУ РОЗДІЛІ

1. Розроблена структура та мікроелектронні компоненти швидкодіючого АЦП у базисі Хаара-Крестенсона на основі застосування парафазних компараторів, який у порівнянні з відомими паралельними АЦП у базисі Радемахера та Хаара-Крестенсона характеризується зменшеною у 2-5 разів апаратною складністю та підвищеною у 2 рази швидкістю.
2. Досліджені системні характеристики базових компонентів спектрального процесора та побудовані порівняльні діаграми його часової та апаратної складності по відношенню до відомих аналогів реалізованих у двійковій системі числення базису Радемахера, що підтвердило переваги його швидкодії.
3. Розроблена структура спектрального спецпроцесора, який характеризується розширеними функціональними можливостями за рахунок того, що в якості базисних функції  $W_j$ , які представлені в залишках по модулю  $P_i$ , можуть бути відповідно обчислені спектри у базисах Фур'є, Радемахера, Крестенсона та інших. Матричні перемножувачі та суматори реалізуються на вентильних модульних матрицях у базисі Хаара, який у порівнянні з відомими пристроями характеризується підвищеною на 2-3 порядки швидкістю.
4. Розроблено програмне забезпечення мовою C# та інтерфейс користувача розрахунку наборів взаємопростих модулів для спектральних спецпроцесорів у базисі Хаара-Крестенсона різної розрядності,
5. На основі програмного комплексу Xilinx Ise 10.3. розроблена програма синтезу компонентів високопродуктивного синтезу компонентів на базі мікроелектронних утилітів ПЛІС фірми Xilinx.
6. Реалізований та впроваджений у промисловість розроблений швидкодіючий спецпроцесор спектрального опрацювання даних на основі мікропроцесорної платформи обладнаної ПЛМ та периферійними комунікаційними модулями вводу/виводу даних, які застосовані в системах моніторингу спектральних

характеристик маніпульованих сигналів спеціалізованих комп'ютерних мереж управління рухомим складом залізничним транспорту.

## ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

1. Виконана систематизація процесорів цифрового спектрального опрацювання сигналів на основі різних архітектур та сфери застосувань у широкому класі задач цифрового опрацювання даних, що дозволило обґрунтувати перспективу розробки та реалізації швидкодіючого спектральних спецпроцесорів у системі числення залишкових класів базису Хаара-Крестенсона. Викладені теоретичні засади спектрального аналізу на основі інтегрального та дискретного перетворення Фур'є, які дозволили оцінити переваги та функціональні обмеження відомих алгоритмів реалізації косинусного перетворень Фур'є при використанні двійкової системи числення базису Радемахера. Досліджені структури цифрових кореляторів, які є базовими компонентами спектральних спецпроцесорів косинусного перетворення Фур'є, проаналізовані характеристики кодових систем ортогональних функцій різних теоретико-числових базисів та встановлена ефективність застосування базису Хаара-Крестенсона при реалізації матрично-модульних операцій у компонентах спектральних спецпроцесорів.
2. Розроблено метод швидкодіючого аналого-цифрового перетворення сигналів у кодовій системі Хаара-Крестенсона, який дає змогу розпаралелити формування кодів залишків у базисі Хаара згідно набору взаємо-простих модулів. Застосування такого методу не вимагає додаткових міжбазисних перетворень Радемахера-Крестенсона і забезпечує швидкодіючу реалізацію модульних операцій множення та додавання цифрових даних у базовому кореляційному модулі спектрального спецпроцесора.
3. Запропоновано метод дискретного косинусного перетворення Фур'є на основі адаптації характеристик ортогональних базисних функцій до експериментальної реалізації коваріаційної функції автокореляції, що дозволило спростити складність обчислювальних операцій та зменшити кількість множень в алгоритмі спектрального опрацювання сигналів.
4. Запропоновано метод генерування ортогональних функцій різних теоретико-числових базисів, адаптованих до характеристик експериментально

отриманих функцій автокореляції, що дозволило реалізувати їх зберігання табличним способом. Це дало змогу вилучити одну з трьох операцій множення в алгоритмі спектрального перетворення Фур'є та здійснити вибірку цифрових значень базисних функцій згідно параметрів дисперсії, математичного сподівання та ступеня затухання енергії функції автоковаріації.

5. Удосконалено метод формування цифрових значень ортогональних базисних функцій перетворення Фур'є шляхом переставлення їх кодів у системі залишкових класів. Це дозволило спростити та підвищити швидкодію спектрального перетворення з можливістю розпаралелення обчислювальних процесів згідно модульної арифметики.
6. Отримав подальший розвиток метод обчислення значення спектру для однієї базисної функції шляхом розпаралелення виконання операцій множення діючих та затриманих у часі залишків цифрових відліків вхідного процесу у системі взаємопростих модулів, множення отриманих залишків добутоків на залишки відповідної ортогональної базисної функції та їх накопичуюче додавання на виході однієї точки спектру. За рахунок цього зменшується об'єм пам'яті для зберігання кодів базисної функції, а також зменшується апаратна складність перемножувача базисної функції на добуток залишків.
7. Розроблена структура та мікроелектронні компоненти швидкодіючого АЦП у базисі Хаара-Крестенсона на основі застосування парафазних компараторів та заміни апаратно складних логічних елементів ВИКЛЮЧАЮЧЕ АБО елементами І-НЕ у кожному  $i$ -тому компоненті АЦП, що дозволило зменшити у 3 рази апаратну складність та підвищити у 2 рази швидкодію АЦП у базисі Радемахера-Крестенсона у порівнянні з АЦП у базисі Радемахера.
8. Розроблена структура спектрального спецпроцесора, який характеризується розширеними функціональними можливостями, які забезпечуються обчисленням спектрів у базисах Фур'є, Радемахера, Хаара та Крестенсона. Підвищення швидкодії такого спецпроцесора на 2 порядки досягнуто

розпаралеленням обчислювальних операцій у матрично-модульних перемножувачах та накопичуючих суматорах, реалізованих на вентильних матрицях у базисі Хаара.

9. Розроблено програмне забезпечення на мові C# для розрахунку наборів взаємопростих модулів спектральних спецпроцесорів у базисі Хаара-Крестенсона різної розрядності, проведено синтез компонентів спецпроцесора на ПЛІС фірми Xilinx. Впроваджена у промисловість схемотехнічна структура швидкодіючого спецпроцесора спектрального опрацювання даних орієнтована на мікроелектронне виконання на кристалі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Трахтман А.М. Основы теорий дискретных сигналов на конечных интервалах /А.М. Трахман, В.А. Трахтман – М.: Сов. радио, - 2008.
2. Залмазон Л.А. Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях/Л.А. Залмазон –М.: Наука, 1989. – 496 с.
3. Бекчанова Ш.Б. Алгоритмы и структуры на основе быстрых преобразований Хаара / Ш.Б. Бекчанова, Х.Н. Зайниддинов // Техника юлдузлари. – Ташкент, –2002. –№4, С.45-54.
4. Задірака В.К. Комп'ютерна арифметика багаторозрядних чисел: Наукове видання / В.К. Задірака, О.С. Олексюк //– Київ. –2003. – 264 с.
5. Николайчук Я.М. Теорія цифрових перетворень мультибазисного супершвидкодіючого процесора /Я.М. Николайчук// Искусственный интеллект. – 2008. – №4. – С. 387-394.
6. Палагин А.В. Реконфигурируемые структуры на ПЛИС / А.В. Палагин, В.Н. Опанасенко, В.Г. Сахарин // УсиМ. – 2000. – № 3. – С.3343.
7. Мельник А О. Програмовані процесори обробки сигналів. – Львів: Вид-тво Національного університету "Львівська політехніка", –2000. –55 с
8. Задірака В.К. Комп'ютерна криптологія: Підручник /В.К. Задірака, О.С. Олексюк // – Київ: – 504 с.
9. Заміховський Л. М. Основи теорії надійності і технічної діагностики систем: Навч. Посіб . / Л. М. Заміховський В. П. Калявін / – Івано-Франківськ, ІФНТУНГ, Факел, 2004. -360с.
10. Заміховський Л. М. Проектування систем діагностування навч. Посіб . /Л. М. Заміховський В. П. Калявін/. – Івано-Франківськ, ІФНТУНГ, Факел, 2004. - 248с.
11. Майоров С.А. Принципы организации цифровых машин / С.А. Майоров, Г.И. Новиков// – Л.: Машиностроение, –1974. – 306 с.
12. Корнейчук В.И. Основы компьютерной арифметики / В.И. Корнейчук, В.П. Тарасенко // – К.: Вища школа. –2003. - С. 34-56.

13. Мельник А.О. Спеціалізовані комп'ютерні системи реального часу /А.О. Мельник// – Львів: Державний університет “Львівська політехніка”, 1996. – 54 с.
14. Самофалов К.Г. Цифровые ЭВМ / К.Г. Самофалов, В.И. Корнейчук, В.П. Тарасенко// – СПб.: Вища школа – К. – 2000. – 528с.
15. Стешенко В.Б. ПЛИС фирмы «Altera»/ В.Б.Стешенко//. – М.: «Додатка» –2002. – 575с.
16. Зотов В.Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx / В.Ю. Зотов/ – М.:Телеком. –2006. – 522с.
17. Палагин А.В. Опыт разработки микропроцессорных распределенных систем реального времени./ А.В. Палагин, Я.М. Николайчук// – Киев: Знание, – 1988. – 19 с.
18. Сергиенко А. Б./ Цифрова обробка сигналів: навч. Учебник для вузов. – Питер, 2002.-608с.:ил
19. Зотов В.Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы XILINX / В.Ю. Зотов / М.:, Горячая Линия. –Телеком, – 2006. –522с.
20. Стешенко В.Б ПЛИС фирмы «Altera» / В.Б. Стешенко // М., «Додека». –2002. –575с.
21. Тарасов И.Е. Разработка цифровых устройств на основе ПЛИС XILINX с применением языка VHDL. /И.Е. Тарасов// М.: Горячая линия. –Телеком, –2005. – 782с.
22. Stallings W. Computer Organization and Architecture / W. Stallings// 5<sup>th</sup> ed., New York, NY: Macmillan Publishing Company Stallings –2000.
23. Tanenbaum A., Structured Computer Organization / A. Tanenbaum// 4<sup>th</sup> ed. Upper Saddle River, NJ: Prentice Hall –1999.
24. Patterson D. Computer Architecture. A Quantitative Approach / D. Patterson, J. Hennessy // Morgan Kaufmann Publishers, Inc –1996.
25. Якименко І.З. Розмежована система числення залишкових класів та спецпроцесори на її основі // І.З. Якименко, О.І. Волинський / Поступ в науку.



Збірник наукових праць Бучацького інституту менеджменту і аудиту. - 2010. - №6, Т1. – С.80-83.

26. Акушский И.Я. Машинная арифметика в остаточных классах / И.Я. Акушский, Д.И. Юдицкий – М: Сов.радио, 1968. – 440 с.

27. Брюхович Е.Н. О принципе построения компьютеров, специализированных на вычислении некоторого набора специальных функций в мультипроцессорной вычислительной системе / Е.Н. Брюхович // Автоматика и вычислительная техника, 1983, № 12. — С. 45-51.

28. Торгашев В.А. Система остаточных классов и надежность ЦВМ. – М.:Советское радио, 1970 – 118 с.

29. Николайчук Я.М. Теорія джерел інформації / Я.М. Николайчук – Тернопіль: ТЗОВ „Терно–граф”, 2010. – 536 с.

30. Червяков Н.И. Нейрокомпьютеры в остаточных классах / Н.И. Червяков, П.А. Сахнюк, А.В. Шапошников, А.Н. Макоха - М.: Радиотехника, 2003. - 272 с.

31. Синьков М.В. Непозиционные представления в многомерных числовых системах / М.В. Синьков, Н.М. Губарени - Киев: Наукова думка, 1979. - 137 с.

32. Мельник А.О. Архітектура комп'ютера: Наукове видання / А.О. Мельник – Луцьк: Волинська обласна друкарня – 2008. – 470с.

33. Петришин Л.Б. Теоретичні основи перетворення форми та цифрової обробки інформації в базисі Галуа / Л.Б. Петришин// Навч. посібник. - Київ.: ІЗіМН МОУ, – 1997. – 237 с.

34. Николайчук Я.М. Проектування спеціалізованих ком'ютерних систем / Я.М. Николайчук, Н.Я. Возна, І.Р. Пітух - навч. посібник для втузів. - Т.: Тенограф, 2010. - 392 с.

35. Николайчук Я.М. Коды поля Галуа: теорія та застосування / Я.М. Николайчук // Монографія - Тернопіль: ТЗОВ «Тернограф» - 2012. – 576с.

36. Лабунец В.Г. Теоретико-числовые преобразования над полями алгебраических чисел. – В кн.: Применение ортогональных методов при обработке

сигналов и анализе систем. – Свердловск: УПИ, 1981, С.44-54.

37. Бекчанов Ш.Б. Синтез быстродействующих спецпроцессоров Хаара на основе матричной диаграммы / Ш.Б. Бекчанов, Х.Н. Зайнидинов – тезис докл. НТК «Молодьож в развитии науки и техники» - Ташкент, 2002 – 78с.

38. Волинський О. Систематизація характеристик теоретико-числових базисів та їх застосування для побудови високопродуктивних спецпроцесорів/ О. Волинський, В. Пуюл // Вісник Тернопільського національного технічного університету. – 2011. – Том 16, №3. – С.183-189

39. Грибанов Ю.И. Автоматические цифровые корреляторы./ Ю.И. Грибанов, Г.П. Веселова, В.Н. Андреев // – М.: Энергия, 1971. – 240с.

40. Албанський І.Б. Спецпроцесори кореляційної обробки сигналів /Албанський І.Б., Заведюк Т.О. // Праці міжнародного симпозіуму ПИТАННЯ ОПТИМІЗАЦІЇ ОБЧИСЛЕНЬ (ПОО-XXXV) Том 1. К.: - 2009.- С.8-13.

41. Ширмовська Н.Г. Оптимальна дискретизація заданих кореляційною функцією сигналів / Методи та прилади контролю якості.- Івано-Франківськ.- 2009.- №22.- С.107-111.

42. А.С. №1282160. Многоканальное устройство для вычисления структурной функции / Я.М. Николайчук – Бюлетень №1.-1987.

43. А.С. №1317455. Многоканальное устройство для вычисления функций эквивалентности / Я.М. Николайчук С.М. Ищеряков – Бюлетень №22.-1987.

44. Кожем'яко В.П. Ефективність функціонування оптико-електронних інтелектуальних структур в системах автоматизованого управління / В.П. Кожем'яко, А.А. Яровий // Вестник Херсонського державного технічного університету. – 2001. - №1(10). - С. 279-281.

45. Календер В. Компьютерная томография / В. Календер // Основы, техника, качество изображений и области клинического использования: фонография. – М.: Техносфера. - 2006. - 344с.

46. Орнатский П.П. Теоретические основы информационно-измерительной техники / 2-е изд., перераб., и доп / П.П. Орнатский – К.: Вища школа. – 1983. – 455с.

47. Explore Zilog's Products. - [Електронний ресурс]. - Режим доступу: <https://www.instrumart.com/products/33416/amprobe-at-5005-underground-cablepipe-locator-system>.
48. Padovani R. Reverse Link Performance of IS-95 Based Cellular Systems. IEEE Personal Communications / Third Quarter // Padovani R., 1994. - P. 28 - 34.
49. Advanced Micro Devices, AMD - Processor Homepage [Електронний ресурс]. - Режим доступу: <https://www.advantest.com/>.
50. RF Solutions and Wireless Communications Technology: RFMD [Електронний ресурс]. - Режим доступу: [http://www.uniTrend.com/en/product/2014\\_0725\\_726.html](http://www.uniTrend.com/en/product/2014_0725_726.html).
51. Шевчук Б. М. Технологія багатофункціональної обробки і передачі інформації в моніторингових мережах. / Б.М. Шевчук, В.К. Задірака, Л.О. Гнатів, С.В. Фраєр // НАН України, Ін-т кібернетики ім. В. М. Глушкова. – К. : Наук. Думка, 2010. – 371 с.
52. Задірака В.К. Методи захисту фінансової інформації: Навчальний посібник / В.К. Задірака. Олексюк О С. - Тернопіль: "Збруч", 2000. - 460с.
53. Аналоговые компараторы. [Електронний ресурс]. - Режим доступу: [http://www.gaw.ru/html.cgi/txt/doc/comparator/comp\\_1.htm](http://www.gaw.ru/html.cgi/txt/doc/comparator/comp_1.htm).
54. Пат. АС 13726221 ССРС, МКИ Н03 М1/38. “Аналого-цифровой преобразователь”/Я.Н. Николайчук. – Опубл. 07.02.88, Бюл. №5.
55. Николайчук Я.М. Дослідження системних характеристик двомірних кодів з особливими кореляційними властивостями/ Я.М. Николайчук, О.М. Заставний //Вісник технологічного університету Поділля –Хмельницький, 2004. – №2. – ч.1, Т2.
56. Николайчук Я.М. Методы цифровой обработки шумоподобных сигналов на основе кодовых ключем / Я.М. Николайчук, Б.М. Шевчук – Киев, Сб. тр. ИКАН УССР, 1988.
57. Заставний О.М Теорія та принципи побудови спецпроцесора на основі базисів Радемахера, Крестенсона, Галуа. / О.М. Заставний, Я.М. Николайчук,

Н.Д. Круцкевич, Р.І. Король // Тези доповідей сьомої міжнародної науково - технічної конференції – Вінниця: «УНШЕРСУМ - Вінниця», 2003 - 114с.

58. Пилипенко І.А. Дослідження методів стиснення інформації. / І.А. Пилипенко // Науковий вісник інституту менеджменту та економіки «Галицька академія» – 2006. – №2(10). – С. 78-82.

59. Яцків Н.Г. Методи стиснення в багатоканальних системах на основі кодів Галуа / Н.Г. Яцків, Я.М. Николайчук // Вісник національного університету «Львівська політехніка». Радіoeлектроніка та телекомунікації. – 2002. - №443. – С. 135-138.

60. Куприянов М., Мартынов О., Панфилов Д. Коммуникационные контроллеры фирмы Motorola. - СПб.: БХВ.-Петербург, 2001.- 560 с.

61. Бекчанова Ш.Б., Зайнидинов Х.Н. Принципы построения высокопроизводительных вычислительных структур// Тезисы докладов НТК Мафкуравий жараёнлар ва Узбекистонда фанлар ривожининг долзарб муаммолари Андижон. - 2002. – С. 441.

62. Садыхов Р.Х., Чеголин П.М., Шмерко В.П. Методы и средства обработки сигналов в дискретных базисах. – Мн.: Наука и техника, 1987. - 296 с.

63. ADSP-21000 Family Application Handbook, Vol. 1, Analog Devices, Free Download at: <http://www.analog.com>.

64. DSP Designer's Reference (DSP Solutions) CDROM // Analog Devices, 1999.

65. General DSP Training and Workshops: [http://www.analog.com/industry/dsp/tech\\_docs.html](http://www.analog.com/industry/dsp/tech_docs.html).

66. Яцків Н.Г., Спецпроцесори обробки даних на основі перетворення Крестенсона – Галуа. / Н.Г. Яцків, Р.І. Король, В.В. Яцків, Т.Г. Федчишин // Вісник Технологічного університету Поділля. – 2003. -ТІ, №3. – С. 105-108

67. Круцкевич Н.Д. Принципи побудови RCG процесора. / Н.Д. Круцкевич, Я.М. Николайчук // Тези міжнар. науково - технічної конф. “Контроль

і управління в складних системах” (КУСС - 2003) – Вінниця: «УНІВЕРСУМ – Вінниця». – 2003. – С. 73.

68. Ширмовська Н. Г. Застосування кореляційного та кластерного аналізу для ідентифікації перед аварійних та аварійних станів процесів буріння. /Н.Г. Ширмовська, І. Б. Албанський, І. Р. Пітух, О.Л. Кулинин // Науковий вісник. - Івано Франківськ. – 2011. - №3 (29).-С.112-116.

69. Волинський О.І. Оптимізація обчислень на основі алгоритмів міжбазисних перетворень Радемахера, Крестенсона та Галуа / О.І. Волинський, О.Д. Круцкевич, П.В. Гуменний // Праці міжнародної молодіжної математичної школи “Питання оптимізації обчислень (ПОО-XXXVII)” Київ: Інститут кібернетики імені В.М. Глушкова НАН України, 2011. С. 32-33.

70. Волинський О.І. Розмежована система числення залишкових класів та спецпроцеси на її основі. / О.І. Волинський, І.З. Якименко // Поступ в науку. Збірник праць Буцацького інституту менеджменту і аудиту – Бучач. – 2010. - №6. Т1. – С. 80-83.

71. L. L. Yang and L. Hanzo, A residue number system based parallel communication scheme using orthogonal signaling: Part II—Multipath fading channels // IEEE Trans. Veh. Technol. – 2002. - Vol. 51. – P. 1541-1553.

72. L.-L. Yang and L. Hanzo, Minimum-distance decoding of redundant residue number system codes // Proc. IEEE ICC '2001. – Helsinki (Finland) – 2001. - P. 2975-2979.

73. L.-L. Yang and L. Hanzo, Performance analysis of coded M-array orthogonal signaling using errors-and-erasures decoding over frequency-selective fading channels // IEEE J. Select. Areas Community. – 2001.- Vol. 19. - P. 211-221.

74. Lakhani G. Some Fast Residual Arithmetic Adders, // International Journal of Electronics, 1994. pp. 225-240.

75. Lenstra H. W., Divisors in residue classes // Math. Comput.- 1984. -Vol. 42. - N 165. - P.331-340.

76. Montgomery P., Modular multiplication without trial division, *Math. of Comput.* – 1985. Vol. 44. - N 170. - P. 519-521.
77. Шауман А. М. Основы машинной арифметики. - Л.: Изд-во Ленинградского ун-та, 1979. -457с.
78. Широчин В.П., Мухин В.Е., Ху Чженбин. Машина состояний в задачах аутентификации пользователей в компьютерных сетях // *Управляющие системы и машины.* - 2003. -№ 5. - С. 59-65.
79. Заставний О.М. Аналіз системних характеристик спецпроцесорів формування вихідних даних аналого-цифрових кодерів // *Вісник Технологічного університету Поділля, Хмельницький*, 2005, №4, ч.1, Т2. С. 223-226.
80. Николайчук Я.М. Теоретичні основи побудови та структура спецпроцесорів в базисі Крестенсона / Я.М. Николайчук, О.І Волинський, С.В.Кулина // *Вісник Хмельницького національного університету.* – 2007. - №3. – Т1. – С. 85-90.
81. Николайчук Я.М., Кусик Я.Б. Коды поля Галуа та їх застосування в перетворювачах форм інформації// *Тезисы докладов 7-го симпозиума Проблемы создания преобразователей формы информации* - Киев: ИКАН Украины. - 1992.
82. Петришин Л.Б. Николайчук Я.Н., Ищеряков С.М., Цифровая обработка сигналов на основе преобразования кодов поля Галуа // *Методы и микроэлектронные средства цифровой обработки и преобразования сигналов.*- Рига: ИЭВТ АН Латвии. - 1989.- С.130 - 132.
83. Столлингс В. Передача данных / В. Столлингс – 4-е изд. – СПб.: Питер, 2004. – 750 с.
84. Бернард С. Цифровая связь. Теоретические основы и практическое применение / С. Бернард – 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильянс», 2003. – 1104с.
85. Таненбаум Э. Компьютерные сети / Э. Таненбаум – 4-е изд. – Питер, 2003. – 992 с.
86. Столлингс В. Современные компьютерные сети / В. Столлингс – СПб.: Питер, 2003. – 783 с.

87. Касянчук М.М. Теорія алгоритмів RSA та Ель-Гамала в розмежованій системі числення Радемахера-Крестенсона / М.М. Касянчук, І.З.Якименко, О.І. Волинський, І.Р. Пітух // Вісник Хмельницького національного університету, Видавництво Хмельницького національного університету. – 2011. – № 3(177). – С.265-272.
88. Николайчук Я.М. Швидкодіючий алгоритм та процесор порівняння чисел у системі залишкових класів / Я.М. Николайчук, О.І. Волинський, С.В.Кулина // Науково-теоретичний журнал "Искусственный интеллект". ППШ МОН і НАН України "Наука і освіта". – 2008. – №3. – С.348-352.
89. Волинський О.І. Методи порівняння та сумування в розмежованій системі числення // Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2009. - №5. Т1. – С.91-94.
90. Албанський І.Б. Дослідження системних характеристик цифрових пристроїв множення реалізованих в різних теоретико-числових базисах / І.Б. Албанський, О.І. Волинський // Вісник Хмельницького національного університету. – 2012. – №2. – С. 179-186.
91. Николайчук Я.М., Якименко І.З.,Воронич А.Р., Волинський О.І. Пристрій визначення залишку багаторозрядного числа // Патент на корисну модель № 68872. МПК G 06 F7/00. Опубл. 10.04.2012. Бюл. № 7
92. Николайчук Я.М., Волинський О.І. Спосіб визначення залишку двійкового числа // Патент на корисну модель № 74576. МПК G 06 F5/00. Опубл. 12.11.2012. Бюл. № 21
93. Наконечний А. Й. Цифрова обробка сигналів: навч. Посібник / А. Й. Наконечний, Р. А. Наконечний, В.А. Павлиш. – Львів: Видавництво Львівської політехніки, 2010.
94. Волинський О.І. Швидкодія міжбазисних перетворювачів Радемахера-Крестенсона./ О.І.Волинський // Збірник матеріалів проблемно-наукової міжгалузевої конференції "Юриспруденція та проблеми інформаційного суспільства"(ЮПС - 2011) )” Івано-Франківськ, 2011. С.71-75.

95. Николайчук Я.М., Волинський О.І. Спосіб визначення залишку двійкового числа. Патент на корисну модель № 74576. МПК G 06 F5/00. Опубл. 12.11.2012. Бюл. № 21
96. Волинський О.І. Теорія, алгоритми та спецпроцесори міжбазисних перетворень Радемахера-Крестенсона./ О.І.Волинський // Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту . – Бучач. - 2012. - №8. С. 50-54.
97. Montgomery P. L.“Modular multiplication without trial division”/ P. L. Montgomery/ Math. Computation, vol. 44, 1985.-pp.519 – 521
98. Волинський О. І., «Методи міжбазисних перетворень на основі розмежованої системи числення залишкових класів», Вісник національного університету “Львівська політехніка” , “Комп’ютерні системи та мережі”– Львів. – 2010. - № 688. - С.53-59.
99. Червяков Н.И. Преобразователь десятичного кода в код системы остаточных классов// А.С. СССР №374595 Бюллетень №15. – 1973
100. Червяков Н.И. Устройство для преобразования чисел из десятичной системы счисления в систему остаточных классов// А.С. СССР №377767 Бюллетень №18. – 1973
101. Болтков А.П., Червяков Н.И., Хлевнов С.Н. Устройство для преобразования чисел из позиционной системы счисления в систему остаточных классов// А.С. СССР №1008729. Бюллетень №12. – 1983
102. Хетагуров Я.А., Руднев Ю.П. Повышение надежности цифровых устройств методами избыточного кодирования. - М.: Энергия, 1974. – 272с.
103. Новиков Л.Г., Шурыгин И.Т. Счётчики импульсов с коэффициентами счёта, управляемыми с помощью двоичного кода. Журнал «Приборы и системы управления» № 6, 1972. –С.30-31
104. Авт. свідоцтво СРСР №206908, кл. G 06 F 15/34.



105. Піх В. Я. Спектральний аналіз інформаційних потоків різних теоретико-числових базисів/ Піх В. Я. // Збірник матеріалів проблемно-наукової міжгалузевої конференції "Юриспруденція та проблеми інформаційного суспільства" – Івано-Франківськ, 2011. – С.65-70.

106. Воронич А. Р. Ентропійні методи формування та цифрового опрацювання сигналів у системах ідентифікації станів квазістаціонарних об'єктів / А. Р. Воронич, І. О. Погонець, В. Я. Піх // Праці міжнародної молодіжної математичної школи «Питання оптимізації обчислень (ПОО-XXXVII) – Крим, Велика Ялта, смт. Кацевелі. – 2011. – с. 34-35

107. Піх В. Я. Методи спектрального косинусного перетворення у нафтогазовій промисловості / В. Я. Піх, Я.М. Николайчук // Збірник тез доповідей Всеукраїнської науково-практичної конференції «Інформаційні технології в освіті, техніці та промисловості» Ів.-Фр., 2013. С.50-51.

108. Николайчук Я. М. Методи спектрального косинусного перетворення Фур'є для розпізнавання сигналів у Хеммінговому просторі на основі різних кореляційних функцій та теоретико-числових базисів/ Я. Николайчук, Я. Піх, Н. Возна, Т.Заведюк // Вісник національного університету «Львівська політехніка» «Комп'ютерні системи та мережі». — 2013. — №773. — С. 89-98

109. Піх В. Я. Методи спектрального діагностування технологічних станів квазістаціонарних об'єктів промисловості/ Николайчук Я.М., Круліковський Б.Б.// Науковий Вісник Івано-Франківського національного технічного університету нафти і газу / Розвідка та розробка нафтових і газових родовищ — 2014. — №2(51). – С.172-178

110. Піх В. Я. Методи побудови спецпроцесорів спектрального аналізу на основі косинусного перетворення Фур'є./ В. Я. Піх, Т. О. Корнійчук// Збірник матеріалів міжнародної координаційної наради ISCM-2014. – Тернопіль, 2014. – С.128-131

111. Піх В. Я. Метод формалізації дискретного косинусного перетворення Фур'є у системі залишкових класів теоретико-числового базису крестенсона / В. Я.

Піх, В.Л. Кімак, Б.Б. Круліковський // Праці VII міжнародної школи-семінару «Теорія прийняття рішень». – Ужгород, УжНУ, 2014.-с.204-205.

112. Воронич А. Р. Ентропійні методи формування та цифрового опрацювання сигналів у системах ідентифікації станів квазістаціонарних об'єктів / А. Р. Воронич, І. О. Погонець, В. Я. Піх // Праці міжнародної молодіжної математичної школи «Питання оптимізації обчислень (ПОО-XXXVII) – Крим, Велика Ялта, см. Кацевелі. – 2011. – С.34-35.

113. Піх В. Я. Методи спектрального косинусного перетворення у нафтогазовій промисловості/ Я.М. Николайчук// матеріали II Всеукраїнської науково-практичної конференції молодих учених і студентів «Інформаційні технології в освіті, техніці та промисловості» Ів.-Франківськ, 2015. – С.16-18.

114. Піх В. Я. Пристрій для обчислення спектрального косинусного перетворення в залишкових класах / Я.М. Николайчук, В.Л. Кімак, Б. Б. Круліковський // Патент України Позитивне рішення по заяві №1865523456

115. Авт. свідоцтво СРСР №337784, кл. G 06 F 15/34. Бюлетень № 15. Опубліковано 05.05.1972.

116. Авт. свідоцтво СРСР №840924, кл. G 06 F 15/36. Бюлетень №23. Опубліковано 23.06.1981.

117. Малашевич Б. М. Неизвестные модулярные суперЭВМ// PC WEEK/RE, М., 2005. № 9. С. 44-45. № 10. – С. 52-54.

118. Волинський О.І. Методи високопродуктивних перетворень великорозрядних чисел з базису Радемахера у базис Крестенсона / О.І. Волинський // Вісник національного університету “Львівська політехніка”, “Комп’ютерні системи та мережі”– Львів. – 2012. - С.39-48.

119. Николайчук Я.М., Волинський О.І. Пристрій для перетворення чисел з позиційної системи в систему залишкових класів. Патент на корисну модель № 76623 МПК G06F5/02 Опублікований 10.01.2013 Бюл.№1.

120. Николайчук Я.М., Албанський І.Б., Волинський О.І. Цифровий автокорелятор. Патент на корисну модель № 76622. МПК G 06F 17/15 Опубл. 10.01.2013 Бюл.№1.

121. Николайчук Я.М. Теорія побудови та компоненти швидкодіючих процесорів на основі досконалої та розмежованої форм системи залишкових класів / Я.М. Николайчук, О.І Волинський, С.В.Кулина // Поступ в науку. Збірник праць Бучацького інституту менеджменту і аудиту.– Бучач.– 2008.– №4. - Т1.– С.31-36.

122. Албанський І.Б. Спецпроцесори кореляційної обробки інформації в різних теоретико-числових базисах / І.Б. Албанський // Вісник національного університету “Львівська політехніка” , “Комп’ютерні системи та мережі ”№ 688 Львів, видавництво львівська політехніка -2010 р., С.10-16.

123. UA №76622 МПК G06F 17/15 (2006.01). Опублікований 10.01.2013, Бюл. №1.

124. Pikh V. Theory and Special Processors of Spectral Cosine Fourier Transformation Based on Various Correlation Functions in Hamming Space/ I. Albanskiy, T.Zavedyuk, G.Korniychuk // Proceedings of the International Conference TCSET’2014. — Lviv, 2014. — PP. 549-551

125. Pikh V. Computer technologies in information security/ Alishov Nadir, Albanskyi Ivan, Bredelev Boris, Davletova Alina, Franko Yurii, Humenniy Petro, Ivasyev Stepan, Kasianchuk Mykhaylo, Kimak Volodymyr, Krulikovskiy Boris, Kudin Anton, Liura Oleg, Nykolaichuk Lubov, Pikh Volodymyr, Pitukh Ihor, Protsiuk Halyna, Segin Andriy, Shvidchenko Inna, Volynskyy Orest, Voronych Artur, Vozna Nataliia, Yakymenko Ihor, Yatskiv Vasyl, Zastavnyi Oleg, Zavediuk Tetiana / edited by Valeriy Zadiraka, Yaroslav Nykolaichuk. -Temopil: "Kart-blansh", 2015- с.287-299.

126. Pikh V. Synthesis of high-performance components of spectral analyzers and special processors for data encryption in Rademacher-Krestenson’s theoretical-numerical basis / V. Kimak, B. Krulikovskiy // Матеріали XIII Міжнародної науково-технічної конференції “Досвід розробки та застосування приладо-технологічних САПР в

мікроелектроніці”: CADSM 2015. – Львів: Видавництво Львівської політехніки, 2015. – 516 с. ISBN 978-617-607-716-9Д 74

127. Kinakh Y.I Reliability of Schoof algorithm and its computational complexity. Proceedings of the Xth International Conference “The Experience of Designing and Application of CAD Systems in Microelectronics”. // Kinakh Y.I, Iakymenko I.Z./ – Lviv-Polyana. – 2009. – С. 107.

128. M.Kasyanchuk Matrix Algorithms of Processing of the Information Flow in Computer Systems Based on Theoretical and Numerical Krestenson’s Basis. Proceedings of the X–th International Conference ”Modern Problems of Radio Engineering, Telecommunications and Computer Science” (TCSET–2010). // M.Kasyanchuk, I.Yakymenko, Ya. Nykolaychuk/–L’viv–Slavske.– 2010. – P.241.

129. Yaroslav Nykolaychuk. Rademacher-Krestenson’s method of between-bases transformations in designing processors / Yaroslav Nykolaychuk, Orest Volynskyy, Andrii Borovyi // Proceedings of the 6<sup>th</sup> International Conference “Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications”. –Prague, Czech Republic. –2011. – pp.310-313.

130. Ivan Albanskiy. Structure and Simulation of Interactive Computer Systems Based on Multibasises Switching Processors / Ivan Albanskiy, Petro Humenniy, Orest Volinskiy, Tanya Zavedyuk // Proceedings of the XI-th International conference “Modern Problems of Radio Engineering, Telecommunications and Computer Science” (TCSET-2012). – Lviv-Slavsk. – 2012. – p.434.

131. Omondi and B. Premkumar,2007, “Residue Number System- theory and implementation,” Imperial College Press.

132. M. Hosseinzadeh, S. J. Jassbi and k. Navi, 2007, “A Novel Multiple Valued Logic OHRNS Modulo rn Adder Circuit,” International Journal of Electronics, Circuits and Systems, Vol. 1, No. 4, pp. 245-249.

133. M. Abdallah and A. Skavantzios, On MultiModuli residue number systems with moduli of forms  $ra$ ,  $rb-1$ ,  $rc+1$  "IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS, VOL. 52, NO. 7 , 2005.

134. F.J. Taylor, "Large Moduli Multipliers for Signal Processing," IEEE Trans. Circuits Syst., vol. CAS-28, pp. 731-736, July 1981.
135. G.A. Jullien, "Implementation of Multiplication, Modulo a Prime Number, with Applications to Number Theoretic Transforms," IEEE Trans. Comput., vol. C-29, pp. 899-905, Oct. 1980.
136. N. Szabo and R. Tanaka, Residue arithmetic and its applications to computer technology, (New York, McGraw-Hill, 1967).
137. M. Hosseinzadeh, K. Navi and S. Timarchi, "Design Residue Number System Circuits in Current mode," 14th Iranian Conference of Electrical Engineering, 2006.
138. M. Hosseinzadeh, K. Navi and S. Timarchi, "New Design of 4-3 Compressor," 11th International CSI Computer Conference of Iran, 2006.
139. K.A. Gbolagade, A Memoryless MRC Technique for RNS-to-Binary Conversion Using The Moduli Set  $(2^n, 2^n-1, 2^{n-1}-1)$ , International Journal of Soft Computing, Vol. 4(3), 2009, pp. 127-130.
140. H. Siewobr, K. A. Gbolagade (2011): "An Efficient RNS Overflow Detection Algorithm" (in press), Far East Journal of Electronics and Communications.
141. K.A. Gbolagade, S. D. Cotofana, Residue Number System Operands to Decimal Conversion for 3-Moduli Sets, Proceedings of 51st IEEE Midwest Symposium on Circuits and Systems (MWSCAS 08), Knoxville, USA, 2008, pp. 791-794.
142. K.A. Gbolagade, S. D. Cotofana, MRC Technique for RNS to Decimal Conversion Using the Moduli Set  $\{2^n + 2, 2^n + 1, 2^n\}$ , Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing., Veldhoven, The Netherlands, 2008, pp. 318-321.
143. K. A. Gbolagade, S. D. Cotofana, Residue-to-Decimal Converters for Moduli Sets with Common Factors, Proceedings of 52nd IEEE International Midwest Symposium on Circuits and Systems, (MWSCAS 2009), Cancun, Mexico, 2009, pp. 624-627.

144. Leonel Sousa, Efficient Method for Magnitude Comparison in RNS Based on Two Pairs of Conjugate Moduli. Electrical and Computer Engineering Department, INESC-ID/IST, TU Lisbon.
145. Theodore L. Residue Addition Overflow Detection Processor Boeing Company, Seattle, Wash. Appl. No.: 414276, Sep. 29, 1989.
146. Theodore L. Houk, Seattle, Wash, Method And Apparatus For Pipelined Detection Of Overflow In Residue Arithmetic Multiplication, Boeing Company, Seattle, Wash. Appl. No.: 472,237, Jan. 30, 1990.
147. Mehrin Rouhifar, Mehdi Hosseinzadeh and Mohammad Teshnehlab, "A new approach to Overflow detection in moduli set  $(2^n-1, 2^n, 2^n+1)$ , International Journal of Computational Intelligence and Information Security, Vol.2, No. 3, March, 2011, pp. 35-43.
148. Tadeusz Tomczak. Hierarchical residue number systems with small moduli and simple converters. International Journal of Applied Mathematics and Computer Science. Volume 21, Issue 1, ,2011, pp. 173–192
149. Askari Yadollahpour , Mehdi Hosseinzadeh. New Squaring Schema for Residue Number System. International Journal of Computer Applications in Technology, Vol 2 (4), pp. 1127-1130.
150. Ittee Teli. Residue number systems in optical computing. International Journal of Electronics and Communication Engineering, vol 2, No 1, 2012, pp. 27-29.
151. Shahram Moharrami and Davar Kheirandish Taleshmekaeil, 2012. The Application of the Residue Number System in Digital Image Processing: Propose a Scheme of Filtering in Spatial Domain. Research Journal of Applied Sciences, 7: 286-292.
152. M. Roshanzadeh, S. Saqaeyan. Error Detection & Correction in Wireless Sensor Networks By Using Residue Number Systems. International Journal of Computer Network and Information Security, 2012, 2, 29-35.
153. Hamidreza Mahyar. Reliable and High-Speed KASUMI Block Cipher by Residue Number System Code. World Applied Sciences Journal 17 (9): 2012, pp. 1149-1158.

154. A. Omondi, B. Premkumar. Residue Number Systems: Theory and Implementation. World Scientific Publishing Co. Pte. Ltd. 2007.p.296.

155. N. Szabo and R. Tanaka, Residue arithmetic and its applications to computer technology, New York: McGraw Hill, (1967), p. 236.

156. H. Siewobr, K. A. Gbolagade, “Application of Residue Number System to Advance Encryption StandardAlgorithm”, International Journal of Computational Intelligence and Information Security, Vol. 2, No. 7, July, 2011, pp.66-72.

157. режим доступа: <http://www.learnabout-electronics.org/Digital/dig43.php>

## ДОДАТОК А

Лістинг програми розрахунку модулів та їх параметрів для спецпроцесорів  
різної розрядності

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Numerics;
namespace KrestensonFindPrimes
{
    public class PrimesFinder
    {
        private readonly Dictionary<string, uint> _primesList;
        private string _primesCvsPath;

        public bool ShowFullInfo
        {
            get;
            set;
        }
        public bool ShowOnlyCoprime
        {
            get;
            set;
        }
        public PrimesFinder(string primesCvsPath)
        {
            _primesCvsPath = primesCvsPath;

            ShowFullInfo = false;
            ShowOnlyCoprime = false;

            _primesList = _parsePrimesCvsFile(_primesCvsPath);
        }
        private Dictionary<string, uint> _parsePrimesCvsFile(string primesCvsPath)
        {
            var strR = new StreamReader(primesCvsPath);
            var res = new Dictionary<string, uint>();
            do
            {
                var line = strR.ReadLine();
                if (line == null) break;
                var lineData = line.Split(new[] { ' ' });

                if (Utils.IsBinaryValue(lineData[1]))
            {
                res.Add(lineData[1], uint.Parse(lineData[0]));
            }
            }
            while (true);
            return res;
        }
        public string FindPrimes(int processorCapacity, int blockCapacity, int
additionalValues)
        {
            if (processorCapacity < blockCapacity)
            {

```



## продовження додатку А

```

throw new ArgumentException("processorCapacity повинен бути більшим чи рівним за
blockCapacity");
    }
    if (processorCapacity == 0 || blockCapacity == 0)
    {
        throw new ArgumentException("processorCapacity i blockCapacity
повинні бути більші за 0");
    }
    var primesCount = (int)Math.Round((double)(processorCapacity /
blockCapacity), MidpointRounding.AwayFromZero) + additionalValues;
    if (primesCount % 2 == 1)
    {
        primesCount++;
    }
    var res = new StringBuilder();
    var ress = _findPrimes(_primesList, blockCapacity, primesCount);
    if (ShowOnlyCoprime)
    {
        ress = _findCoprimes(res);
    }
    foreach (var value in ress)
    {
        res.AppendLine(ShowFullInfo ? value.ToString() :
value.GetRealNumber().ToString());
    }
    return res.ToString();
}
private List<ExtendedNumber> _findCoprimes(IEnumerable<ExtendedNumber>
allItems)
{
    var existMultipliers = new List<uint>();
    var result = new List<ExtendedNumber>();
    foreach (var number in allItems)
    {
        var multipliers = (from entry in number.GetMultipliers() where
entry > 1 select entry).ToArray();
        var multiplierExist = multipliers.Any(existMultipliers.Contains);
        if (multiplierExist)
        {
            var res2 = new List<ExtendedNumber>();
            var res = new List<uint>();
            var searchList = new Dictionary<string, uint>();
            var maxValue1 = (uint)Math.Pow(2, itemMaxCapacity);
            res.Add(maxValue1);
            res2.Add(new ExtendedNumberPowerer(2, (uint)itemMaxCapacity));
            var maxValue = (uint)Math.Pow(2, itemMaxCapacity)-1;
            res.Add(maxValue);
            res2.Add(new
ExtendedNumberMultiplier(Utils.GetMultipliers(maxValue)));
            foreach (var pair in primesList)
            {
                if (pair.Key.Length < itemMaxCapacity)
                {
                    searchList.Add(pair.Key, pair.Value);
                }
                else if (pair.Key.Length == itemMaxCapacity)
                {
                    if (res.Contains(pair.Value))
                        continue;
                    res.Add(pair.Value);
                    res2.Add(new ExtendedNumberPrime(pair.Value));
                }
            }
        }
    }
}

```

## продовження додатку А

```

else
    {
        break;
    }
}
var needContinue = res.Count < returnPrimesCount;

if (needContinue)
{
    var tempRes = new List<uint>();

    foreach (var pairOuter in searchList)
        {
            uint temp;
            foreach (var pairInner in searchList)
                {
                    if (pairOuter.Key.Length +
pairInner.Key.Length - 1 > itemMaxCapacity)
                    {
                        continue;
                    }
                    temp = pairInner.Value * pairOuter.Value;

                    if (temp >= maxValue || tempRes.Contains(temp) ||
res.Contains(temp))
                        continue;
                    tempRes.Add(temp);
                    res2.Add(new ExtendedNumberMultiplier(new[] {
pairInner.Value, pairOuter.Value }));
                    if (!tempRes.Contains(pairOuter.Value))
                    {
                        tempRes.Add(pairOuter.Value);
                        res2.Add(new ExtendedNumberPrime(pairOuter.Value));
                    }
                    for (var i = 2; i <= itemMaxCapacity; i++)
                    {
                        temp = (uint)Math.Pow(pairOuter.Value, i);
                        if (temp > maxValue)
                        {
                            break;
                        }
                    }
                    if (tempRes.Contains(temp))
                        continue;
                    tempRes.Add(temp);
                    res2.Add(new ExtendedNumberPowerer(pairOuter.Value,
public class ExtendedNumberPowerer : ExtendedNumberMultiplier
{
    private readonly uint _powerCoefficient;
    public ExtendedNumberPowerer(uint multiplier, uint powerCoefficient)
        : base(Enumerable.Repeat(multiplier, (int)powerCoefficient).ToArray())
    {
        _powerCoefficient = powerCoefficient;
    }
    public override string ToString()
    {
        return RealNumber + " (Складене число: " + Multipliers[0] +
            " ^ " + _powerCoefficient + ")";
    }
}

```

## Додаток Б

## Лістинг коду програмованого спектрального спецпроцесора

```

/// Digital_Correlator///
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
=====
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
=====
-----
entity Digital_Correlator is
  Port ( clk      : in STD_LOGIC;
        rst      : in STD_LOGIC;
        serial_in : in STD_LOGIC;
        data_out  : out STD_LOGIC_VECTOR (15
downto 0));
end Digital_Correlator;
--
=====
-----
architecture Behavioral of Digital_Correlator is

-- Signal Declaration
  signal unit_2_serial_in : std_logic;
  signal unit_3_serial_in : std_logic;
  signal reg_1_data_out  : std_logic_vector (3
downto 0);
  signal reg_2_data_out  : std_logic_vector (3
downto 0);
  signal reg_3_data_out  : std_logic_vector (3
downto 0);
  signal shift_en        : std_logic;
  signal tmp
  : std_logic_vector (3 downto 0);
  signal control_bus
  : std_logic_vector (6 downto 0);
  signal data_out_int
  : std_logic_vector (15 downto 0);
-----
-----
-- Component Declaration
component Processing_Unit is
  Port ( clk      : in
        rst      : in
        shift_en  : in
        serial_in : in STD_LOGIC;
        control_bus : in
        STD_LOGIC_VECTOR (6 downto 0);
        reg_data_feedback : in STD_LOGIC_VECTOR
(3 downto 0);
        serial_out : out STD_LOGIC;
end component;

reg_data_out      : out
STD_LOGIC_VECTOR (3 downto 0);
function_out      : out
STD_LOGIC_VECTOR (7 downto 0);
end component;
-----
-----
begin

Unit_1 : Processing_Unit
  port map ( clk => clk,
            rst => rst,
            shift_en => shift_en,
            serial_in =>
            serial_in =>
            control_bus =>
            reg_data_feedback
            serial_out =>
            reg_data_out =>
            reg_1_data_out,
            function_out =>
            data_out_int(7 downto 0));

Unit_2 : Processing_Unit
  port map ( clk => clk,
            rst => rst,
            shift_en => shift_en,
            serial_in =>
            control_bus =>
            reg_data_feedback
            serial_out =>
            reg_data_out =>
            reg_2_data_out,
            function_out =>
            data_out_int(15 downto 8));
Unit_3 : Processing_Unit
  port map ( clk => clk,
            rst => rst,
            shift_en => shift_en,
            serial_in =>
            control_bus =>
            reg_data_feedback
            serial_out => open,
            reg_data_out =>
            function_out =>
            open);

```

## продовження додатку Б

```

-----
control_reg : process (clk)
begin
    if clk'event and clk = '1' then
        if rst = '1' then

            control_bus <= "1111000";
            else

                control_bus <= control_bus(5 downto 0) &
control_bus(6);
                    end if;
                end if;
            end process;
-----
            tmp(0) <= not(control_bus(0) and
control_bus(3) and reg_1_data_out(0));
            tmp(1) <= not(control_bus(1) and
control_bus(4) and reg_1_data_out(2));
            tmp(2) <= not(control_bus(2) and
control_bus(5) and reg_2_data_out(0));
            tmp(3) <= not(control_bus(3) and
control_bus(6) and reg_2_data_out(3));
-----
            shift_en <= not(tmp(0) and tmp(1) and tmp(2)
and tmp(3));
            data_out <= data_out_int;
-
=====
end Behavioral;
/// Processing_Unit ///
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
=====
entity Processing_Unit is
    Port ( clk : in
STD_LOGIC;
        rst : in
STD_LOGIC;
        shift_en : in
STD_LOGIC;
        serial_in : in STD_LOGIC;
        control_bus : in
STD_LOGIC_VECTOR (6 downto 0);
        reg_data_feedback : in STD_LOGIC_VECTOR
(3 downto 0);
        serial_out : out STD_LOGIC;
        reg_data_out : out
STD_LOGIC_VECTOR (3 downto 0);
        function_out : out STD_LOGIC_VECTOR (7
downto 0));
end Processing_Unit;
-----
--
=====
architecture Behavioral of Processing_Unit is
-----
-- Signals declarartion

        signal reg_data_int : std_logic_vector (7 downto
0);
        signal data_int : std_logic_vector (7 downto
0);
        signal data_tmp : std_logic_vector (7 downto
0);
        signal data_out_int : std_logic_vector (15
downto 0);
-----
-- Components declarartion
        component Serial_In_Shift_Reg is
            generic ( width : natural);
            port ( clk : in STD_LOGIC;
                rst : in
STD_LOGIC;
                shift_en : in
STD_LOGIC;
                serial_in : in
STD_LOGIC;
                data_out : out
STD_LOGIC_VECTOR ((width-1) downto 0));
            end component;

        component Logic_Unit is
            port ( data_in_bus : in
STD_LOGIC_VECTOR (7 downto 0);
                control_bus : in
STD_LOGIC_VECTOR (6 downto 0);
                data_out_bus : out STD_LOGIC_VECTOR (7
downto 0));
            end component;

        component Serial_Adder is
            port ( clk : in STD_LOGIC;
                rst : in
STD_LOGIC;
                wr_en : in
STD_LOGIC;
                add_1 : in
STD_LOGIC_VECTOR (7 downto 0);
                sum_out: out
STD_LOGIC_VECTOR (15 downto 0);
                over_flow : out
STD_LOGIC );
            end component;
-----
begin

```

```

Shift_Reg : Serial_In_Shift_Reg
  generic map ( width => 8)
  port map      ( clk => clk,
                                     rst => rst,
                                     shift_en => shift_en,
                                     serial_in => serial_in,
                                     data_out => reg_data_int );

Logic_Circuit : Logic_Unit
  port map ( data_in_bus => data_tmp,
            control_bus => control_bus,
            data_out_bus => data_out_int);

Math_Circuit :Serial_Adder
  port map ( clk  => clk,
            rst => rst,
            wr_en => shift_en,
            add_1 => data_int,
            sum_out => data_out_int,
            over_flow => open);
-----
      data_tmp <= reg_data_feedback &
reg_data_int(3 downto 0);
-----
      serial_out <= reg_data_int(7);
      reg_data_out <= reg_data_int(3 downto 0);
      function_out <= data_out_int (7 downto 0);
--
=====
=====
end Behavioral;

/// Logic_Unit///

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---- Uncomment the following library declaration if
instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
-----
entity Logic_Unit is
  Port ( data_in_bus : in STD_LOGIC_VECTOR (7
downto 0);
        control_bus : in STD_LOGIC_VECTOR (6
downto 0);

```

продовження додатку Б

```

data_out_bus : out STD_LOGIC_VECTOR (7 downto
0));
end Logic_Unit;
-----
-----
architecture Behavioral of Logic_Unit is
-----
      signal data_out : std_logic_vector (7 downto 0);
-----
begin
      data_out <= data_in_bus(7) & (data_in_bus(6
downto 0) and control_bus);
      data_out_bus <= data_out;
-----
end Behavioral;

/// Serial_In_Shift_Reg///

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
=====
=====
entity Serial_In_Shift_Reg is
  Generic ( width : natural);

  Port ( clk      : in STD_LOGIC;
        rst      : in
STD_LOGIC;
        shift_en : in
STD_LOGIC;
        serial_in : in
STD_LOGIC;
        data_out  : out
STD_LOGIC_VECTOR ((width-1) downto 0));
end Serial_In_Shift_Reg;
--
=====
=====
architecture Behavioral of Serial_In_Shift_Reg is
-----
      signal data_tmp : std_logic_vector ((width-1)
downto 0);
-----
begin
      process ( clk, shift_en )
        begin
          if clk'event
            and clk ='1' then
              if
                rst = '1' then
                  data_tmp <= (others => '0');

```

## продовження додатку Б

```

        elsif shift_en = '1' then
            data_tmp(width-1) <=
serial_in;
            data_tmp((width-2) downto 0)
<= data_tmp((width-1) downto 1);
        end
    if;
        end if;
    end process;
-----
-----
data_out <=
data_tmp;
-----
-----
end Behavioral;
/// Serial_Adder ///
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
-----
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
-----
-----
entity Serial_Adder is
    Port ( clk   : in  STD_LOGIC;
          rst   : in  STD_LOGIC;
          add_1 : in  STD_LOGIC_vector (7 downto 0);
          wr_en : in  STD_LOGIC;
          over_flow : out STD_LOGIC;
          sum_out : out STD_LOGIC_VECTOR (15
downto 0));
end Serial_Adder;
--
-----
-----
architecture Behavioral of Serial_Adder is
-----
-----
-- Signals declarartion

```

```

signal result_tmp : std_logic_vector (16 downto 0);
    signal add_tmp  : std_logic_vector (15 downto
0);
    signal sum_tmp  : std_logic_vector (15 downto
0);
    signal sum_int   : std_logic_vector (15
downto 0);
    signal over_flow_int : std_logic;
-----
-----
begin
    add_tmp <= "00000000" & add_1;
-- Unsigned 16-bit Adder with Carry-Out
    result_tmp <= conv_std_logic_vector
((conv_integer (add_tmp) + conv_integer(sum_int)),17);
    sum_tmp <= result_tmp(15 downto 0);
    over_flow_int <= result_tmp(16);
-- Parallel Register
    process ( clk, rst )
        begin
            if clk'event and clk = '1' then
                if rst = '1' then
                    sum_int <= (others
=> '0');
                elsif wr_en = '1' then
                    sum_int <=
sum_tmp;
                else
                    sum_int <=
sum_int;
                end if;
            end if;
        end process;
-----
-----
        over_flow <= over_flow_int;
        sum_out <= sum_int;--
-----
-----
end Behavioral;

```

## ДОДАТОК В

акти про впровадження дисертаційної роботи

**ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ  
«ТЕРНОПІЛЬСЬКЕ КОНСТРУКТОРСЬКЕ БЮРО РАДІОЗВ'ЯЗКУ «СТРІЛА»**

46020, м. Тернопіль, вул 15 Квітня, 6, тел/факс – 28-75-00, 28-72-00, tkbr\_strila@ukr.net, tkbr\_strila@mail.ru  
р/р 26009308101054 в ТВБВ 10019/08 Філія Тернопільського обласного управління АТ "Ошадбанк",  
МФО 338545, код 14042350  
№ 156 від 30.11.2015 р.

**АКТ'**

**про впровадження результатів дисертаційної роботи аспіранта кафедри  
Комп'ютерних систем та мереж Івано-Франківського національного  
технічного університету нафти і газу Піха Володимира Ярославовича  
"Процесори спектрального косинусного перетворення Фур'є на основі різних  
кореляційних функцій та теоретико-числових базисів"**

Даний акт складений про те, що результати дисертаційної роботи Піха В.Я, "Процесори спектрального косинусного перетворення Фур'є на основі різних кореляційних функцій та теоретико-числових базисів " передані для впровадження ТОВ ТКБР "Стріла" що включають:

1. Високопродуктивні алгоритми дискретного спектрального косинусного перетворення шляхом кодування даних та адаптованих до асимптотики автоковаріації ортогональних функцій у теоретико числовому базисі Хаара-Крестенсона.

2. Схемотехнічні рішення спецпроцесора у системі залишкових класів базису Крестенсона, який дозволяє підвищити на 1-2 порядки швидкодію спектрального аналізу сигналів у реальному часі

Ці результати використані у процесі впровадження програмно-апаратних засобів моніторингу спектральних характеристик маніпульованих сигналів у спеціалізованих низових комп'ютерних мережах моніторингу автоматики рухомого залізничного складу.

Директор ТОВ „ТКБР „Стріла”




О.О.Рафалюк



«ЗАТВЕРДЖУЮ»

Проректор з наукової роботи  
Тернопільського національного  
економічного університету

 проф. З.-М.В.Задорожний  
« 27 »  2016р.

АКТ

про впровадження результатів дисертаційної роботи молодшого наукового співробітника  
Науково-дослідного інституту інноваційного розвитку та державотворення  
Піха Володимира Ярославовича “Процесори спектрального косинусного перетворення  
Фур’є на основі різних кореляційних функцій та теоретико-числових базисів” у науково-  
дослідній роботі на тему:

“Розробка теоретичних засад та алгоритмів ідентифікації запусків потужних електроприводів у  
високовольтних ЛЕП на основі теорії розпізнавання образів”  
(державний реєстраційний номер 0115U002340)

Ми, комісія у складі завідувача кафедри спеціалізованих комп’ютерних систем Тернопільського національного економічного університету, наукового керівника науково-дослідної роботи, д.т.н., професора Николайчука Я.М. та завідувача відділу організації науково-дослідних робіт та маркетингу Науково-дослідного інституту інноваційного розвитку та державотворення ТНЕУ Письменного В.І., створена для приймання роботи, виконаної в рамках науково-дослідної роботи на тему “Розробка теоретичних засад та алгоритмів ідентифікації запусків потужних електроприводів у високовольтних ЛЕП на основі теорії розпізнавання образів” (державний реєстраційний номер 0115U002340), встановила:

1. Розроблено метод мультибазисного аналого-цифрового перетворення сигналів у ТЧБ Хаара-Крестенсона, який характеризується високою швидкістю і може бути використаний у пристроях розпізнавання накидів та замикань на землю у високовольтних ЛЕП.
2. Розроблено метод швидкодіючого виконання дискретного косинусного перетворення Фур’є на основі адаптації характеристик ортогональних базисних функцій до експериментальної реалізації коваріаційної функції автокореляції, який може бути використаний для аналізу якості та спектральних гармонічних складових електроенергії.

Завідувач кафедри спеціалізованих  
комп’ютерних систем ТНЕУ  
д.т.н., професор



Николайчук Я.М.

Науковий керівник  
науково-дослідної роботи,  
д.т.н., професор



Николайчук Я.М.

Завідувач відділу організації  
науково-дослідних робіт  
та маркетингу НДІ ІРД



Письменний В.І.

**ЗАТВЕРДЖУЮ:**

Проректор з науково-педагогічної роботи Івано-Франківського національного технічного університету нафти і газу,  
д-р техн. наук, проф.

  
О. М. Мандрик


« 12 » грудня 2015 р.

**АКТ**

**про впровадження у навчальний процес наукових результатів дисертаційної роботи асистента кафедри програмного забезпечення автоматизованих систем Піха В. Я. «Процесори спектрального косинусного перетворення Фур'є на основі різних кореляційних функцій та теоретико-числових базисів»**

Ми, що нижче підписалися, директор інституту інформаційних технологій, канд. техн. наук, доцент Чигур І.І., завідувач кафедри програмного забезпечення автоматизованих систем д-р. техн. наук, професор Юрчишин В.М., асистент кафедри Піх В. Я., склали цей акт у тому, що матеріали дисертації «Процесори спектрального косинусного перетворення Фур'є на основі різних кореляційних функцій та теоретико-числових базисів» Піха В.Я. впроваджені у навчальний процес при вивченні дисципліни «Алгоритми та структура даних», змістовний модуль ЗМ2, навчальний елемент НЕ2.4 «Побудова алгоритмів на основі спектрального косинусного перетворення Фур'є, з використанням кореляційних функцій та теоретико-числових базисів» для студентів спеціальності для студентів спеціальності 6.05010301 – Програма інженерія.


Директор інституту інформаційних технологій,  
канд. техн. наук, доцент

  
І.І. Чигур

Завідувач кафедри програмного  
забезпечення автоматизованих систем,  
д-р техн. наук, професор

  
В.М. Юрчишин

асистент кафедри програмного забезпечення  
автоматизованих систем

  
В.Я. Піх